Ecole Nationale Supérieure des Sciences de
l'Information et des Bibliothèques

Université Claude Bernard Lyon 1

DESS en INFORMATIQUE DOCUMENTAIRE

Rapport de Stage

Système d'information sous Linux, et logiciels libres

Christophe LIENARD

Effectué sous la direction de

M. Claude AVISSE

INSTITUT NATIONAL DE LA RECHERCHE AGRONOMIQUE
17 rue Sully
BV 1540
21034 DIJON CEDEX

1999

SYSTÈME D'INFORMATION SOUS LINUX, ET LOGICIELS LIBRES - Christophe LIENARD

RÉSUMÉ

L'évolution actuelle en matière de logiciels libres a motivé la mise en place d'un Intranet expérimental au sein de l'Institut National de la Recherche Agronomique de Dijon. Un serveur Apache a été installé sur une plate-forme Linux, afin d'accueillir une copie du site déjà existant sur le centre et tournant sous Unix. Les principales préoccupations ont été ensuite de reproduire l'ensemble des services proposés par ce site au moyen d'applications compatibles avec Linux, et de développer de nouvelles applications répondant aux besoins du centre.

MOTS-CLÉS

INTERNET – INTRANET – LOGICIEL LIBRE – LINUX - SERVEUR – SITE WEB – Z39.50 – WAIS - BASE DE DONNÉES – SYSTÈMES D'INFORMATION – INDEXATION

INFORMATION SYSTÈM UNDER LINUX, AND OPEN SOURCES APPLICATIONS - Christophe LIENARD

ABSTRACT

Subsequent development of open sources movment has been the origin of an experimental Intranet creation within the Institut National de la Recherche Agronomique de Dijon center. An Apache server has been put together with a Linux operating system, in order to install within it a copy from the existing site, which runs under Unix. Main purpeoses were then to reproduce the whole services which were offered by this site thanks to applications being able to run under Linux, and next to develop new applications in order to satisfy center's wants.

KEYWORDS

INTERNET - INTRANET - OPEN SOURCE - LINUX - SERVER – WEB SITE - Z39.50 - WAIS – DATABASE - INFORMATION SYSTEMS – INDEXING

Remerciements

Je tiens à remercier

Claude Avisse, responsable de l'Unité Présidence Equipe Documentation de Dijon, pour m'avoir donné la possibilité de travailler sur un sujet novateur et particulièrement intéressant, ainsi que pour son soutien durant l'ensemble du stage ;

Christophe Caron, Ingénieur d'Etudes en Informatique au centre INRA de Jouy-en-Josas, pour son concours précieux lors de l'installation du module SFgate, et Régine Szymanski, Ingénieur d'Etudes en Informatique au centre de Dijon, pour l'aide technique apportée ;

Enfin, je remercie Dominique Aouchiche, Florence Contour et Jacques Prévost, de l'équipe de documentation de Dijon, pour l'accueil chaleureux qu'ils m'ont réservé.

Abréviations

ANSI : American National Standards Institute
BBS : Bulletin Board System
CGI : Common Gateway Interface
CPAN : Comprehensive Perl Archive Network
DAEMON : Disk And Extension MONitor
DNS : Domain Name Server
FAQ : Frequently Asked Questions
FSF : Free Software Foundation
FTP : File Transfert Protocol
GID : Group IDentification
GNU : GNU's Not Unix
GPL : General Public Licence
HTML : HyperText Markup Language
NCSA : National Center for Supercomputing Applications
NISO : National Information Standards Organization
OCR : Optical Characters Recognition
ODBC : Open DataBase Connectivity
PERL : Practical Extraction and Report Language
PID : Process IDentification
SSI : Server-Side Includes
UID : User IDentification
URL : Uniform Resource Locator

Table des matières

Introduction

Le renouveau du logiciel libre se matérialise actuellement de plusieurs façons. On peut constater un engagement massif de l'industrie informatique dans ce créneau, un des derniers exemples en date étant la décision de Sun Microsystems de publier le code source d'une application de référence au sein de la communauté Linux, la suite bureautique StarOffice. Sun venait en effet de racheter l'éditeur de la suite (entreprise Star Division). Une autre manifestation de ces changements est le développement d'applications qui concurrencent ou même dominent leurs homologues commerciaux. Deux logiciels appartenant à cette catégorie, Apache, qui règne sur le monde des serveurs, et le système d'exploitation Linux, bénéficiant d'une croissance exceptionnelle, ont été à l'origine d'un projet Intranet expérimental au sein du centre de Dijon. Ce projet devait servir d'extension au site INRA de Dijon géré par l'Unité Présidence Equipe Documentation, mais hébergé sur le serveur d'un autre centre INRA (Jouy-en-Josas, près de Paris), qui tournait sous Unix.

Un budget a été alloué à l'aspect matériel du projet (achat d'un PC pentium II cadencé à 350 MHz et doté de 64 Mo de mémoire vive). L'aspect logiciel, quant à lui, devait reposer en totalité sur le principe du logiciel libre. La seule composante établie au départ de ce stage était le choix du système d'exploitation : Linux. Le reste devait être déterminé par la suite, le problème de la compatibilité d'applications tournant sous Unix avec un système d'exploitation encore jeune devant être étudié au cas par cas. Il s'est avéré très rapidement que le deuxième point crutial du projet, le choix du logiciel serveur, ne posait aucun problème. L'application sur laquelle s'était portée notre attention dès le départ, Apache, était entièrement compatible avec Linux, et présentait toutes les garanties de fiabilité et de performance désirées. Le début de l'"expérimentation" pouvait commencer. La première entrée en matière fut donc de se familiariser avec Linux, puisque M. Avisse s'était chargé de l'installation préalable du système d'exploitation.

I. L'Institut National de la Recherche Agronomique

A. Présentation générale

L'INRA est un établissement public de recherche placé sous la tutelle du ministère de l'Education Nationale, de la Recherche et de la Technologie, et du ministère de l'Agriculture et de la Pêche. Il est composé de 21 centres de recherche régionaux, de 256 unités de recherche, et de 79 unités expérimentales, soit 8 570 agents titulaires comprenant le personnel scientifique, les ingénieurs, les techniciens, le personnel administratif et les stagiaires. Chaque structure de recherche ou d'expérimentation est rattachée géographiquement à l'un des 21 centres, et fonctionnellement à l'un des 17 départements de recherche ou à l'une des 16 directions (6 directions scientifiques, 5 directions administratives, 5 directions relationnelles).

L'INRA définit ainsi ses orientations de recherche

- Mieux nourrir les hommes et préserver leur santé.
- Aménager et gérer avec sagesse leurs espaces de vie.
- Innover sur le front des sciences et des technologies, notamment celles du vivant, en restant vigilant et responsable.
- Comprendre et piloter la complexité de nos systèmes biologiques, économiques et sociaux.

B. Le centre de recherche de Dijon

1. Présentation

Le centre INRA de Dijon regroupe l'ensemble des laboratoires et domaines expérimentaux implantés en Bourgogne et Franche-Comté et un laboratoire de la région Rhône-Alpes. Il est constitué de 20 unités de recherche et recense 420 agents (dont 190 chercheurs et ingénieurs, 195 techniciens et 30 agents administratifs), soit 5% de l'effectif total de l'INRA. Son domaine représente 138 hectares pour 8 000 m² de serres et enceintes climatisées.

Quatre axes de recherche sont couverts

- Pôle qualité des aliments (qualité organoleptique et fonctionnelle des produits alimentaires, et valeur santé des aliments).
- Pôle végétal (création de matériel végétal, maladies épidémiques des végétaux - en particulier de la vigne - causées par des phytoplasmes, maîtrise de la qualité du pois, et dynamique des peuplements végétaux, systèmes de culture et lutte contre les mauvaises herbes).
- Pôle sciences sociales (agriculture familiale, dynamique des espaces ruraux, évaluation des politiques socio-culturelles, gestion technique et économique d'exploitations agricoles et de systèmes agraires soumis à la protection des ressources naturelles, innovations socio-techniques et changements des métiers en agriculture).
- Centre de microbiologie du sol et de l'environnement (CMSE) : microbiologie du sol à l'interface entre la plante, le sol, l'air et l'eau, préservation de la fertilité des sols.

Ces pôles de recherche sont répartis sur cinq implantations :

- Dijon-Ville, rue Sully : pôles du Centre de Microbiologie du Sol et de l'Environnement, du Centre d'Etude et de Recherche sur la Qualité des Aliments et leur VALeur Santé (CERQUAVALS), et des Production et Protection Végétales (PPV).
- Dijon-Epoisses, Bretenières : pôles PPV
- ENESAD, boulevard Petitjean à Dijon : pôles Sciences Sociales et Développement.
- Région Franche-Comté, Poligny : Station de Recherche en Technologies et Analyses Laitières.
- Région Rhône-Alpes, Thonon : Station d'Hydrobiologie Lacustre.

2. L'Unité Présidence Equipe Documentation (UPE-Doc)

L'UPE-Doc appartient aux services relationnels de l'INRA. Elle relève de la Direction de l'Information et de la Communication, qui regroupe aussi les services des éditions et des publications.

Les principaux objectifs de l'UPE-Doc sont la gestion du fonds documentaire, la collecte de l'information scientifique et technique et sa diffusion au sein des chercheurs, le recensement de l'ensemble des productions scientifiques du centre, et enfin le conseil et la formation des usagers aux techniques documentaires.

a. *Effectifs*

Mme AOUCHICHE, bibliothécaire, M. AVISSE, responsable de la documentation, Mme CONTOUR, secrétaire, et M. PREVOST, documentaliste.

b. *Ressources matérielles*

➢ Informatique

- Un ordinateur Dell Pentium II350 sous Windows 95 / Linux Red Hat 5.2.
- Un ordinateur Dell Pentium II350 sous Windows 95 relié à une imprimante laser en réseau.
- Un ordinateur Pentium 233MMX sous Windows NT Workstation relié à un scanner HP, à un imageur HR 6000, et à l'imprimante réseau.
- Un ordinateur 486/33 sous Windows 3.11 relié à une imprimante jet d'encre couleur.
- Un ordinateur 486/33 sous Windows 3.11
- Un ordinateur 486 sous Windows 3.11 relié à un lecteur de Cédéroms (chargeur de 6 unités) et à l'imprimante réseau.
- Une plate-forme de sauvegarde.

➢ Divers

- Un photocopieur couleur
- Un lecteur de microfiches

c. *Ressources logicielles*

- Documentation : Texto-Ligne, Texto-Windows, Texto-Web, EndNote, Psylog
- Editeur HTML : HoTMetaL Pro 5.0
- Statistiques (bibliométrie) : Sampler
- Traitement des images : Corel Paint

d. *Le fonds documentaire*

Les domaines traités sont la recherche agronomique (agro-alimentaire, agriculture, environnement), et les activités spécifiques aux quatre stations du centre de Dijon.

Les ouvrages

- Les ouvrages de référence
  - Les collections encyclopédiques universelles : l'Encyclopaedia Universalis, l'Encyclopédia Britannica, etc.
  - Les collections encyclopédiques spécialisées : le Traité de chimie organique, les Techniques de l'ingénieur, les Techniques agricoles, etc.
  - Les dictionnaires : le Robert de la Langue Française, le Nouveau Larousse Agricole, etc.
  - Les manuels et atlas spécialisés : la série des "Handbook", l'Atlas de la France Verte, etc.
  - Les annuaires et répertoires : le Kompass, le Bottin Administratif, le Bottin des Professions, Agrirep (répertoire des organismes agricoles), etc.

- Les monographies

  Plus de 2 500 ouvrages généraux, répartis en deux catégories :
  - Les ouvrages édités par l'INRA et appartenant à l'UPE-Doc.
  - Les ouvrages de la bibliothèque commune, et issus des différentes stations du centre de Dijon.

Les publications en série

- Les collections de périodiques : 450 collections appartenant aux différentes stations de Dijon. Seuls les derniers numéros sont conservés au sein des laboratoires, les autres sont regroupés à la bibliothèque commune de l'UPE-Doc.
- Abonnements : Biofutur, Info PC, Pour la science.

Les productions scientifiques du centre de Dijon

L'UPE-Doc recense l'ensemble des publications des chercheurs appartenant au centre de Dijon et des thèses soutenues en son sein. Elle dispose aussi de plus de 2 000 diapositives.

Les cassettes vidéo

L'UPE-Doc détient une soixantaine de cassettes vidéo issues des éditions INRA.

Les bases de données

- Les Cédéroms
  - AGICOLA : agriculture et économie agricole de 1979 à 1998.
  - FSTA (Food Science and Technology Abstracts) : technologie alimentaire, nutrition et toxicologie de 1969 à 1999.

- Les bases de données internes accessibles par le serveur INRA national
  - PUBINRA : ensemble des publications des chercheurs de l'INRA au niveau national. Cette base est alimentée par les différentes unités régionales de documentation.
  - OUVINRA : catalogue collectif des ouvrages disponibles dans les différents centres de recherches de l'INRA en France.
  - MEDLINE.
  - Commonwealth Agricultural Bureaux (CAB).

- Les bases de données externes
  - Chemical Abstracts (CA).
  - CAB.
  - FPAT (French PATent), EPAT (European PATent), PCTPAT (Patent Cooperation Treaty PATents).
  - World Patent Index (WPI) de Derwent.
  - PASCAL.

Ces bases sont accessibles sur les serveurs :

- QUESTEL ORBIT.
- STN (Scientific and Technical Network).
- DIDIM (Deutches Institut Dokumentation und Information für Medizinische).

Enfin, l'UPE-Doc est abonnée aux CURRENT CONTENTS sur disquettes (Life Science et Agricultural Biological and Environmental Sciences - ABES) diffusés en réseau local.

e. ***Les services proposés par l'UPE-Doc***

A l'ensemble des utilisateurs

- Prêt des monographies.
- Consultation sur place.
- Interrogation des Cédéroms.
- Service quotidien de questions/réponses par téléphone, télécopie, et accueil des usagers sur place.

Au personnel de l'INRA

- Numérisation de documents.
- Réalisation de diapositives.
- Recherches documentaires.

II. Le site INRA de Dijon

A. Origine du site et définition du projet de stage

Les origines du site de Dijon remontent à 1994, alors qu'Internet n'en était qu'à ses débuts, son hébergement étant assuré par un serveur localisé à Dijon et qui tournait sous Unix. L'installation d'un système de gestion de bases WAIS permettait alors d'interroger une ou plusieurs bases à partir du site. Cette application fut perdue suite à des problèmes informatiques, et le site fut transféré sur le serveur de Jouy-en-Josas. Texto est le logiciel documentaire sélectionné par l'INRA, et son utilisation a été généralisée dans ses services de documentation ; l'apparition du logiciel Texto-Web n'a plus justifié la réinstallation de l'application WAIS, puisqu'il étendait les possibilités d'utilisation de Texto (accessible en local ou par telnet) à une utilisation sur le Web.
Les bases Texto actuellement disponibles via Texto-Web sur le site de Dijon sont les notes de services, une partie des ouvrages du laboratoire de microbiologie des sols et les périodiques du centre.
Le site n'a pas connu depuis 1994 d'évolutions majeures en terme de gestion des flux d'informations.

A ces considérations se superpose une évolution marquante des mentalités, favorisée par la croissance d'Internet : la renaissance du mouvement des logiciels libres, ou "open source" (logiciels dont les exécutables et le code source sont accessibles gratuitement à tous).
Le développement de plus en plus important des logiciels libres, et tout particulièrement de logiciels serveurs tels qu'Apache, ou de systèmes d'exploitation appartenant au monde Unix tels que Linux, a été à l'origine de ce projet : installer à l'UPE-Doc une plate-forme d'hébergement du site INRA de Dijon. Si d'un point de vue logique, il pouvait paraître critiquable de devoir passer par un serveur situé vers Paris pour consulter des ressources locales, mener à bien ce projet offrait plusieurs avantages, dont une indépendance vis-à-vis du serveur de Jouy-en-Josas. L'administration d'un serveur est en effet considérablement facilitée lorsque l'on dispose des droits du super-utilisateur … Et le site demeure bien entendu accessible sur le réseau local ethernet couvrant le centre de Dijon, même en cas de problème de liaison avec Internet, ce qui s'est produit à plusieurs reprises durant la période de stage, parfois pendant plusieurs heures.

Mais l'indépendance a son prix. L'inconvénient majeur de ce projet résidait dans les problèmes de sécurité que l'on rencontre lorsque l'on se connecte au réseau des réseaux. Il a donc été décidé de scinder le projet en deux parties distinctes : d'une part, le site INRA de Dijon serait dupliqué en intégralité sur un serveur

Apache tournant sous Linux et installé à l'UPE-Doc, et constituerait un site expérimental. Il y recevrait tous les développements jugés utiles, et permettrait diverses expérimentations, son accès étant réservé exclusivement aux machines de l'INRA. D'autre part, le site hébergé par le serveur national serait conservé, et en accès public (mis à part la section Intranet, réservée au centre de Dijon). Il profiterait néanmoins des réorganisations effectuées initialement sur le site expérimental, ainsi que de certains des services nouvellement apportés.
Une évolution prévue est de passer définitivement sur le serveur de l'UPE-Doc lorsqu'un ingénieur en sécurité informatique sera à même d'assurer la protection du site contre les intrusions extérieures.

B. Présentation du Travail réalisé

Le site INRA de Dijon répond à deux objectifs majeurs :

- Présenter le centre, ses chercheurs et leurs travaux.
- Permettre une meilleure diffusion de l'information au sein du centre, via la section Intranet.

Dans un premier temps, une plate-forme d'hébergement a été installée pour accueillir le site expérimental, qui au départ n'était qu'une copie du site INRA de Dijon, transférée à l'UPE-Doc.

Puis une réorganisation des deux sites a été entreprise dans le but de mieux répondre aux objectifs fixés ; elle a porté sur deux aspects principaux :

- Fonctionnalité du site

Le site de Dijon présentait initialement sur sa page d'accueil l'intitulé détaillé de chacun de ses services les uns à la suite des autres, obligeant l'utilisateur à recourir aux ascenseurs pour rechercher la section désirée. Un regroupement en quatre rubriques a été effectué, afin de présenter le contenu du site dans un seul écran de façon concise et claire.
Par ailleurs, des outils de communication (forums de discussions, services d'inscriptions, de réservations) et de recherche (moteur) ont été installés sur le site expérimental ; des sources d'informations supplémentaires ont été mises à la disposition des utilisateurs : bases des notices bibliographiques et des monographies de l'UPE-Doc, possibilité de consulter chaque sommaire, mise en lignes des diaporamas de chaque congrès.

- Ergonomie du site

A l'origine constituées de textes agrémentés des seuls logos de l'INRA et du centre de Dijon, les quatre rubriques nouvellement créées ont été agencées sur la page d'accueil, au sein d'un montage comprenant les deux logos et des photos illustrant les différents pôles d'activités du centre (voir la page d'accueil en Annexe).
Les index des différentes rubriques ont été placés dans des structures tabulaires, et le fond de chaque page HTML choisi selon sa place dans l'arborescence du site. Ceci pour permettre une meilleure lisibilité, et faciliter le repérage au sein de cette arborescence.

Outre un remodelage du site, il a été procédé à la création de nouvelles pages HTML selon les besoins du centre : par exemple à l'occasion de l'annonce du congrès DEFI'99 (rencontre docteurs-chercheurs-entreprises à l'ENESAD de Dijon) prévu en décembre prochain.

1. Plan du site

Ce plan a été établi après les travaux effectués sur les deux sites, et fait la distinction entre la partie commune à l'ensemble, et celle spécifique à un site donné.

Présentation du centre	Situation géographique et implantations		
	Présentation du centre de Dijon		
	Les unités logistiques		
Recherche scientifique	Liste des unités de recherche	Centre de Microbiologie du Sol et de l'Environnement (CMSE)	•Flore Pathogène du sol •Microbiologie des sols •Phytoparasitologie
		Projet CST830"Microbial inoculant in agriculture and environment"	
		Centre d'Etudes et de Recherches sur la Qualité des Aliments et leur Valeur Santé	•Arômes •Plate-forme de Prédéveloppement en Biotechnologies •Qualité des aliments de l'homme •Technologie et Analyse Laitières
		Pôle Végétal	•Génétique et amélioration des plantes •Agronomie •Malherbologie •Phytopharmacie et biochimie des interactions cellulaires
		Sciences Sociales	•Economie et Sociologie Rurales •Systèmes Agraires et Développement
		Station d'Hydrobiologie Lacustre de Thonon-les-Bains	
	INRA - CompAct[1]		
Information Scientifique et Technique	L'information brevets à l'usage des chercheurs de l'INRA		
	Index synonymique de la flore de France		
	Publications des chercheurs		
	Recherches documentaires	•Notices bibliographiques •Sommaires •Documentation Catalogue(s) en ligne ouvrages, périodiques, notes de service	

Intranet	Les notes de service INRA depuis 1971	
	Informations du Centre	•Conclusion des débats des conseils scientifiques et de gestion du Centre de Dijon sur la consultation nationale concernant la réorganisation de l'INRA 17/10/97. •Restructurations et mouvements d'unités sur le centre 23/10/97
	Comptes rendus	•Compte rendu de la réunion du conseil de gestion 29/11/96 •Compte rendu de la réunion du conseil scientifique 22/11/96
	Divers	•Logiciel documentaire EndNote •Une information sur l'Euro
	Accès réservé à l'administrateur du site	•Gestion du forum •Mise à jour des index •Analyse du site

Cette partie est commune au site hébergé par le serveur de Jouy-en-Josas et à celui hébergé par le serveur de l'UPE-Doc, mis à part la dernière partie ("Accès réservé à l'administrateur du site") qui ne concerne que le serveur de Dijon. Le premier niveau de l'arborescence (caractères en rouge foncé gras) correspond à ce que l'on voit sur la page d'accueil du site. Le second niveau, en grisé, est constitué d'une page par

[1] CompAct : Compétences et activités des chercheurs de l'INRA

rubrique, chacune de ces pages utilisant un fond commun (<BODY BACKGROUND="images/bg524.jpg">). En bas de la page d'accueil se trouve un bandeau, différent selon le serveur, qui offre les services suivants :

Pour le serveur de l'UPE-Doc :

Forums de discussions	Accès réservé (administration)	Recherches sur le site	Statistiques du site	Nouveautés •Agenda •Congrès • L'information brevets à l'usage des chercheurs de l'INRA

Pour le serveur de Jouy-en-Josas :

Retour au serveur national	Congrès •Defi'99 •Congrès de Dijon	Nouveautés •L'information brevets à l'usage des chercheurs de l'INRA	Envoi de courrier électronique à l'administrateur du site

2. Développement du site

Les règles d'édition donnent une certaine unité aux pages du serveur, facilitent la navigation et améliorent la qualité d'accès à l'information. Si nécessaire, l'administrateur du serveur se réserve le droit de faire modifier la présentation des pages réalisées par les rédacteurs des unités pour une meilleure lisibilité des informations. Le document "Politique et règles éditoriales du serveur du centre de Tours-Nouzilly" (http://www.tours.inra.fr/inranet/general/regles.htm) a servi de base à l'élaboration de ces règles, qui ont été placées en annexe.

Conventions d'écriture adoptées à partir de ce chapitre

Les lignes de texte sur fond grisé correspondent au report dans le mémoire d'un résultat affiché sur l'écran d'ordinateur suite à l'exécution d'un programme ou d'une requête. Celles dans un cadre correspondent à du code HTML ou à des scripts Perl issus d'un travail personnel. Enfin, le paramétrage de la presque totalité des applications installées sur ce serveur (ainsi que le serveur lui-même) passe par l'édition et la modification de fichiers de configuration : pour chaque application, les lignes modifiées sont explicitées et reportées sous forme de tableau.

III. Linux : la base fondatrice du projet

Introduction

Linux est une implémentation libre et gratuite des spécifications POSIX, avec des extensions System V et Berkeley, et protégée par le copyright GNU (GNU's Not Unix). Depuis 1994 (version 1.0), Linux n'est plus considéré comme un système en béta-test. Son développement est ouvert et réparti, ses nouvelles versions, stables ou non, sont accessibles au grand public. Une convention de numérotation basée sur trois nombres (x.y.z) permet de caractériser la stabilité d'une version x.y.z : une version stable aura son "y" pair, et l'incrémentation de "z" correspond à une correction de bogues. Lorsque "y" est

impair, la version est en béta-test, contenant de nouvelles fonctionnalités, mais destinée aux seuls développeurs.

Remarque - Logiciels libres et GNU.

L'un des premiers projets de logiciels libres vit le jour dans le monde Unix, et fut le projet GNU. Derrière GNU, la FSF[2] (Free Software Foundation) a créé la Licence Publique Générale (GPL) GNU. Elle permet de partager et de modifier les logiciels librement accessibles afin de garantir qu'ils restent disponibles pour l'ensemble des utilisateurs, et sert de référence aux droits des logiciels libres. Toute modification d'un logiciel GPL doit être diffusée sous GPL.
Un logiciel libre n'est pas forcément gratuit : de nombreux logiciels libres se trouvent gratuitement sur le web, ce qui n'interdit pas aux entreprises de commercialiser des versions payantes (mais souvent très bon marché) sous forme de cédéroms avec notice complète, et contrat d'assistance à l'installation ou de maintenance. Les sociétés RedHat, Caldera et SuSE distribuent ainsi différentes versions de Linux.

A. Les raisons de ce choix

Les contraintes étaient les suivantes : le système d'exploitation devait être multi-tâches et multi-utilisateurs afin de pouvoir supporter la gestion et l'utilisation d'un serveur ; en outre, il devait provenir de sources libres, indépendantes d'une société quelconque. Linux a été privilégié par rapport à des versions libres d'Unix (comme FreeBSD) en raison de sa croissance remarquable dans tous les domaines (développement, utilisation, implication des grandes firmes …) ; 13% des entreprises sont maintenant équipées avec un système d'exploitation Linux, ce qui constitue une avancée notable si l'on considère que cette proportion avoisinait 0% en 1997 [2].

Créé en 1991 par l'étudiant finlandais Linus Torvalds, Linux compte aujourd'hui des millions d'utilisateurs avec un taux d'augmentation annuel de plus de 100%, sans publicité ni marketing. Gratuité, aptitudes en réseaux, réputation de grande fiabilité, implication croissante des acteurs économiques majeurs, l'ascension de Linux semble suivre celle d'Internet.

Linux bénéficie déjà du soutien de géants de l'industrie (Intel, Oracle), et d'une reconnaissance officielle, dont voici par exemple deux illustrations [5] :

- En France, le ministère de l'Education Nationale et l'Association Francophone des Utilisateurs de Linux ont signé le 29 octobre 1998 un accord pour accompagner le développement de Linux dans le système éducatif de l'hexagone.
- Au Mexique, le projet pour l'informatisation des établissements scolaires, Scholar Net, prévoit d'installer Linux sur près d'un million d'ordinateurs en 5 ans, sur les postes de travail comme sur les serveurs de réseau.

Enfin, seulement huit ans après sa naissance, Linux dispose de plusieurs milliers de logiciels pour la plupart gratuits, reprenant les fonctionnalités des principaux best-sellers de chaque domaines. Quelques exemples de logiciels libres :

- Apache : voir le chapitre suivant.
- Star Office offre un traitement de textes, un tableur, un système de gestion de bases de données, un logiciel graphique et de nombreux utilitaires. Il est de plus compatible avec les formats de fichiers des logiciels de bureautique de Microsoft.
- Corel WordPerfect : version Linux d'un des traitements de textes les plus utilisés au monde.
- GIMP : logiciel de retouche d'images reprenant l'essentiel des fonctionnalités de Photoshop, avec possibilité de créer des applications graphiques ou des modules optionnels.
- Etc.

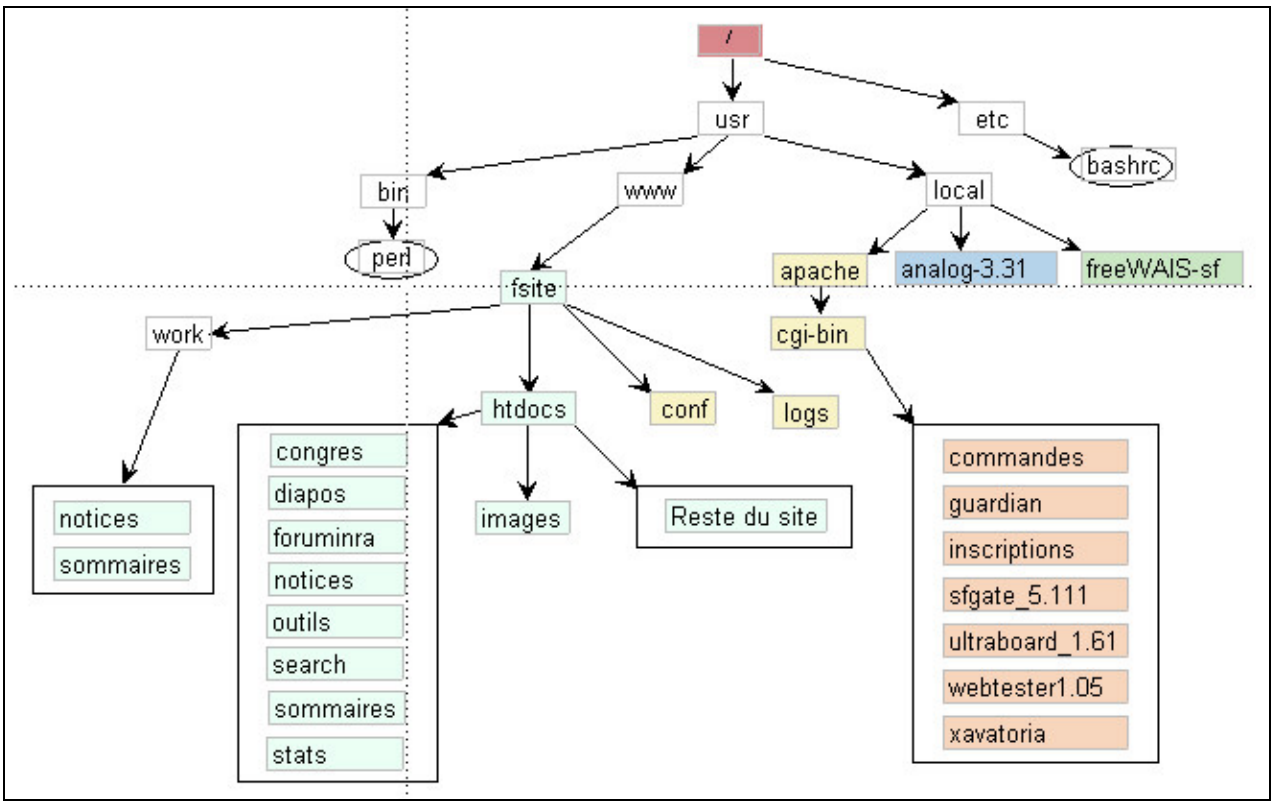
[2] Le site de la FSF est à l'adresse : http://www.fsf.org/

B. Exploitation de Linux

Ce chapitre traite les notions de base qu'il a fallu acquérir afin de mener à bien le stage.

1. Systèmes de fichiers

Linux utilise une arborescence de répertoires unique, mais composée de différents systèmes de fichiers répartis sur un ou plusieurs disques durs. Un système de fichiers est formé lorsqu'une partition est rattachée à l'arborescence de fichiers par un répertoire appelé *point de montage*. Cette arborescence de répertoires est transparente aux yeux des utilisateurs, qui n'ont, sauf exception, pas à se soucier des points de montage ni des différentes partitions. Dans notre cas, quatre systèmes de fichiers ont été montés : `/`, `/home`, `/usr` et `/var`.
Ci-dessous quelques répertoires importants de l'arborescence de Linux, et la majeure partie des répertoires créés au cours du stage.



Remarque - liens durs et symboliques entre fichiers

Cette remarque est introduite afin de clarifier l'explication d'un paramétrage d'Apache relatif aux liens symboliques, dans le chapitre consacré au serveur.
Un inode (ou nœud d'index) est une structure qui contient les informations fondamentales d'un fichier. Chaque inode est conservé dans une table, où il est identifié par un numéro, le numéro d'index ou inumber. Un répertoire ne contient en fait que les noms et inumbers des fichiers. Quand Linux a besoin d'accéder à un fichier, il cherche son nom dans le répertoire, lit le inumber, et utilise les renseignements fournis par l'inode correspondant. Cette connexion entre un nom de fichier et son inode s'appelle un lien dur, qui en fait relie un nom de fichier avec le fichier proprement dit. Il est possible de définir plusieurs liens sur le même fichier, qui sera donc connu sous plusieurs noms. Ce sera l'inumber qui identifiera un fichier donné, et non son nom. Pour créer ce type de lien, la commande est :

```
ln fichier_source fichier_destination
```

Le deuxième type de lien est le lien symbolique. Ce dernier ne contient plus l'inumber du fichier original, mais son chemin d'accès, et permet un lien entre fichiers n'appartenant pas aux mêmes systèmes de fichiers (contrairement au lien dur). Il est créé par la commande :

`ln -s fichier_source fichier_destination`

Le lien dur permet une économie de place sur le disque. Le lien symbolique visualise les origines du lien.

Présentation de quelques répertoires importants

- **`/usr/bin`** : conserve une bonne partie des programmes exécutables du système.
- **`/usr/etc`** : contient des fichiers très divers de configuration du système.
- **`/usr/local`** : comprend l'essentiel des logiciels qui ne font pas partie de la distribution d'origine, et qui sont rajoutés par la suite.
- **`/usr/man`** : contient l'aide en ligne pour les programmes du système.
- **`/usr/src`** : conserve les codes source des différents programmes du système.

Montage et démontage des systèmes de fichiers

L'utilisation des lecteurs de cédéroms et de disquettes passe par leur montage (à chaque utilisation) puis leur démontage (à chaque fin d'utilisation).

- Le lecteur de Cédéroms
  `% mount -t iso9660 -r /dev/cdrom /cdrom` (montage)
  `% umount /cdrom` (démontage)

- Le lecteur de disquettes
  `% mount /dev/fd0 /floppy` (montage)
  `% umount /floppy` (démontage)

2. Fichiers de configuration des interpréteurs

Le fichier `/root/.bashrc` (qui concerne donc le shell Bash, rc signifiant "running command", et le super-utilisateur *root*) est susceptible de contenir des commandes ou des éléments de programmation. A la différence du fichier `/etc/.bash_profile` qui n'est exécuté qu'une fois, à la connexion, `/root/.bashrc` est exécuté à chaque nouveau lancement d'un shell (à chaque ouverture d'une fenêtre xterm). Son contenu est reporté ci-dessous. Les alias permettent de créer des raccourcis clavier, pour accéder par exemple rapidement aux répertoires les plus souvent utilisés.

```
# .bashrc

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
alias print='lpr -Php'
alias Xc='cd /usr/X11R6/lib/X11'
alias so='cd /usr/local/Office40/bin'
alias cgi='cd /usr/local/apache/cgi-bin/'
alias htd='cd /usr/www/fsite/htdocs/'
alias not='cd /usr/www/fsite/work/notices/'
alias som='cd /usr/www/fsite/work/sommaires/'

if [ -f /etc/bashrc ]; then
  . /etc/bashrc
fi
```

Le bloc if vérifie si le `fichier /etc/bashrc` existe, auquel cas il l'exécute. Voici son contenu :

```
# /etc/bashrc
```

```
PS1="\$PWD > "
PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/usr/bin/mh:/root/bin:
/usr/local:/usr/local/bin:/usr/local/apache/cgi-bin/SFgate-5.111"

alias which="type -path"
```

- La variable PS1 a été paramétrée pour que le prompt affiche en permanence la position courante de l'utilisateur.
- Lorsque l'on veut pouvoir lancer un programme, des lignes de commandes, etc., sans avoir à spécifier le chemin d'accès complet, on rajoute ce chemin d'accès à la variable PATH. Par exemple, le fichier `/usr/local/go` contient la ligne de commnande :

```
/usr/local/apache/bin/httpd -f /usr/www/fsite/conf/httpd.conf
```

L'exécution de ce fichier permet de lancer le daemon[3] httpd du serveur Apache :

`% /usr/local/go.`

Mais puisque l'on a rajouté `/usr/local` dans la variable PATH, il suffit de taper (où que l'on se situe dans l'arborescence :

`% go`

3. Gestion des comptes utilisateur

Création de comptes utilisateur

Chaque utilisateur du système doit disposer de son propre compte et mot de passe, exceptions faites du compte *guest* (compte en accès libre) ou de comptes particuliers permettant par exemple de consulter une base de données. Le fichier `/etc/passwd` contient toutes les informations relatives aux comptes des utilisateurs. Seul root doit avoir des accès en écriture, les autres utilisateurs n'ayant que des droits en lecture. Les lignes du fichier `/etc/passwd` comprennent (dans cet ordre) les sept champs suivants :

- Le nom de compte de l'utilisateur.
- Le mot de passe du compte (crypté).
- L'UID (User ID : identifiant d'utilisateur), qui est un entier identifiant l'utilisateur pour le système d'exploitation.
- Le GID (Group ID, identifiant de groupe).
- Le commentaire : contient souvent le nom réel de l'utilisateur, ou ses coordonnées.
- Le répertoire de connexion : répertoire dans lequel l'utilisateur est placé lorsqu'il se connecte au système.
- La commande : elle est exécutée après la phase de connexion ; il s'agit souvent d'un interpréteur de commandes.

Chaque champ est séparé par deux points, même lorsque le champ est vide. Un extrait du fichier `/etc/passwd` est reporté ci-dessous :

```
root:8Gted0Oxc67hc:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
…
ftp:*:14:50:FTP User:/home/ftp:
nobody:*:99:99:Nobody:/:
```

[3] Le serveur est démarré en lançant httpd : le d signifie daemon (Daemon : Disk And Extension MONitor), et désigne un processus non invoqué manuellement mais s'exécutant en tâche de fond dans l'attente d'un signal précis ou d'une condition qui se vérifie. De nombreux deamons sont lancés au démarrage du système (par exemple `/etc/inittab`, qui est chargé d'attendre sur un terminal qu'un utilisateur se connecte, et qui est responsable de l'affichage de l'invite login).

```
avisse:9YwfeZSesUa2Y:200:510::/home/avisse:/bin/bash
lienard:ydpE6k6VTEWrg:201:510::/home/lienard:/bin/bash
```

Ajout d'utilisateurs

Il est possible d'effectuer cette opération de deux manières :

Manuellement

➲ Editer le fichier `/etc/passwd`, et rajouter les lignes adéquates. Le champ du mot de passe reste vide ; lorsque les modifications sont sauvegardées, lancer la commande :

```
% passwd nom_utilisateur
```

➲ Le mot de passe est alors demandé, *root* peut en rentrer un générique, qu'il donnera ensuite à l'utilisateur concerné. Celui-ci devra alors le modifier dès qu'il se connectera au système.

➲ L'étape suivante est la création du répertoire de connexion de l'utilisateur, et la modification de son propriétaire, qui doit être l'utilisateur en question :

```
% mkdir /home/nom_utilisateur
% chown nom_utilisateur /home/nom_utilisateur
```

Il faut ensuite placer le nouvel utilisateur au sein d'un groupe : éditer le fichier `/etc/group` et ajouter le nom de l'utilisateur au(x) groupe(s) désiré(s) (voir plus bas pour le fichier `/etc/group` et les groupes d'utilisateurs).

➲ La dernière étape consiste à copier les fichiers de configuration des interpréteurs désirés (ici le bash) dans le répertoire de connexion nouvellement créé, et d'y donner accès à l'utilisateur :

```
% cp /home/ancien_utilisateur/.bashrc /home/nom_utilisateur/
% chown nom_utilisateur /home/nom_utilisateur/.bashrc
```

En utilisant un programme d'ajout d'utilisateur

Linux Red Hat fournit le programme `adduser`. Une fois lancé, toutes les informations nécessaires sont demandées.

Retrait d'utilisateurs

De même, il est possible d'opérer manuellement ou à l'aide d'un programme. La première méthode est privilégiée, car elle permet de mieux connaître son environnement de travail.

➲ La destruction manuelle d'un compte commence par le retrait de la ligne associée dans le fichier `/etc/passwd`, puis l'effacement des répertoires de l'utilisateur :

```
% rm -rf /home/répertoire_de_connexion_utilisateur
```

➲ Puis supprimer la boîte aux lettres de l'utilisateur. L'ensemble des boîtes sont réunies par lien symbolique dans le répertoire `/var/mars_nwe/sys/mail/user/` (chaque lien porte le nom d'un utilisateur), les boîtes aux lettres réelles étant dans `/var/mars_nwe/sys/mail/` (chaque boîte est nommée par un numéro).

➲ Enfin, effacer toutes les tâches périodiques ou programmées pouvant lui appartenir. Elles se situent dans le répertoire `/var/spool/cron/crontabs/nom_utilisateur/` (voir dans ce chapitre : *5. L'automatisation des tâches*).

4. Gestion des groupes d'utilisateurs

Chaque utilisateur d'Unix ou de Linux appartient à un ou plusieurs groupes, choisis pour des raisons très diverses mais faisant toujours appel à une communauté d'intérêts. Les groupes peuvent être configurés de façon à ce que leurs membres disposent d'accès spécifiques à des périphériques, des systèmes de fichiers, des machines, etc.

Le fichier `/etc/group`, dont l'organisation est similaire à celle du fichier `/etc/group`, contient toutes les informations relatives aux groupes d'utilisateurs. Il est constitué des quatre champs suivants, de gauche à droite, chaque champ étant séparé par deux points :

- Le nom du groupe (nom unique).

- Le mot de passe. Champ souvent vide ou contenant une étoile, un "x". Lorsque le mot de passe est définit, un utilisateur désirant changer de groupe devra l'indiquer.
- L'identifiant du groupe (GID). Entier unique pour chaque groupe, utilisé par le système d'exploitation.
- Le champ *membres*. Contient la liste de tous les utilisateurs du groupe, séparés par des virgules.

Ci-dessous, un extrait du fichier `/etc/group` :

```
root::0:root
bin::1:root,bin,daemon,caron
daemon::2:root,bin,daemon,caron
…
ftp::50:
nobody::99:
users::100:
floppy:x:19:
doc:x:510:avisse,lienard
pppusers:x:230:
popusers:x:231:
slipusers:x:232:
```

Pour ajouter un utilisateur à un groupe, il suffit d'ajouter son nom à la ligne du groupe souhaité. Pour le détruire, on procède inversement et on contrôle le fichier `/etc/passwd` afin de modifier la définition de tous les utilisateurs dont c'était le groupe par défaut (leur GID doit être modifié, sans quoi, ils ne pourraient plus se connecter au système, leur groupe n'étant plus valide).

Remarque - changement de groupe

Sur Linux, un utilisateur connecté ne peut appartenir qu'à un seul groupe à un moment donné. Si l'on est membre de plusieurs groupes, la commande **`newgrp`** permet d'en changer. Le groupe initial d'un utilisateur est défini par le champ GID du fichier `/etc/passwd`.

5. Automatisation des tâches : la "crontable"

La "crontable" est une table dans laquelle on stocke les commandes à exécuter automatiquement à intervalle régulier. Pour l'utilisateur *root*, elle se situe dans le répertoire `/var/spool/cron/crontabs/root`.

Trois tâches étaient à automatiser pour le bon fonctionnement du site :

- Une sauvegarde journalière des fichiers modifiés (voir le dernier chapitre).
- Le lancement hebdomadaire de Webtester, qui effectue une cartographie et une vérification des liens du site.
- Le lancement journalier d'Analog, qui réalise une analyse statistique du site.

Pour ce faire la "crontable" `crontab.fsite` a été créée. Le fichier contient les lignes suivantes :

```
# lancement de Webtester tous les dimanches à 2h
0 2 * * 0 /usr/local/apache/cgi-bin/webtester_files/config.pl
# lancement d'Analog tous les jours à 2h.
0 2 * * * /usr/local/analog3.31/analog
```

Chaque ligne contient six colonnes :

- Minute (0 à 59)
- Heure (0 à 23)
- Date du mois (1 à 31)
- Mois (1 à12)
- Jour de la semaine (0 à 6 : dimanche = 0, lundi = 1, etc.)
- La commande à exécuter (binaire, exécutable, script, etc.)

Il est possible d'utiliser des jokers (*), qui signifient "tous", ou des choix multiples, séparés par des virgules.

6. Commandes diverses utilisées fréquemment

- **`df`**

Affiche l'espace libre sur chacun des systèmes de fichiers montés ou sur celui précisé, si un nom est ajouté à la suite de la commande.

- **`find / -name nom_fichier -print`**

Permet de retrouver l'emplacement d'un fichier ou d'un répertoire dans l'arborescence complète. La troncature (*) est possible.

- **`gzip <options><fichiers>`**

Les options utilisées au cours de ce stage ont été :

c : affiche la sortie sur la sortie standard et ne modifie pas les fichiers d'entrée.
d : décompression des fichiers (identique à la commande gunzip).

- **`kill -9 nom_processus`**

Arrête un processus, par exemple `httpd` ou `waisserver`. L'option "-9" renforce la commande kill. Par exemple, kill seul n'était pas suffisant pour arrêter `waisserver`.

- **`ls -la | more`**

Donne la liste détaillée de l'ensemble des fichiers présents dans le répertoire courant, fichiers cachés inclus. La commande **`more`** permet un affichage page par page.

- **`ps aux`**

Donne la liste détaillée de tous les processus démarrés sur la machine.

- **`su nom_utilisateur`**

Pour exécuter une commande en étant identifié comme un autre utilisateur. C'est une alternative commode à la déconnexion puis la reconnexion. Il est demandé un mot de passe pour un utilisateur faisant **`su root`**, au contraire de *root*, bien sûr, lorsqu'il exécute **`su nom_utilisateur`**.

- **`tar <fonction><options><fichiers>`**

Les options de `<fonction>` utilisées ont été :

c : créer une nouvelle archive.
x : extraire des fichiers d'une archive, sans modifier l'archive.
t : afficher le contenu d'une archive.

Les paramètres `<options>` utilisés au cours de ce stage ont été :

v : passer en mode "verbeux" afin d'obtenir davantage d'informations.
f : spécifier le nom du fichier d'archive à lire où à écrire.

7. Mise à jour de Linux

Les sources du noyau officiel sont diffusées sous la forme d'une archive tar compactée par gzip, à laquelle il convient d'ajouter les différentes révisions. Chaque révision fait l'objet d'un fichier séparé contenant ce que l'on appelle un patch (produit par la commande `diff`), qui peut être appliqué à l'aide d'une commande portant le même nom, `patch`.
Ex. Pour passer de la version 2.0.15 à celle 2.0.36, télécharger tous les fichiers patch numérotés 16 à 36. Leur nom débutent par patch-, le numéro de la version du noyau suit, puis celui du patch, et ils sont souvent compactés grâce à gzip. Par exemple, patch-2.0.25.gz.

8. Ressources sur Internet

Les liens sur Linux sont innombrables. Ne seront reportés ici que ceux ayant été les plus utilisés durant le stage.

- Le site suivant se définit lui-même comme un " index thématique de pages Web consacrées au système d'exploitation Linux, à ses applications et plus généralement au logiciel libre".

http://www.linux-center.org/fr/

- La liste (langue française) et le forum (langue anglaise) répertoriés ici sont particulièrement actifs.
  ☞ linux-debutland@onelist.com
  comp.os.linux.misc

IV. Apache : un complément parfaitement adapté

Introduction

Apache est né d'un projet réunissant via Internet un groupe de volontaires (connus sous le nom d'Apache Group) disséminés de part le monde. Ce projet visait à implémenter un serveur HTTP puissant, pouvant supporter une utilisation commerciale, et dont le code source et la documentation devaient être librement accessibles par l'intermédiaire du Web. Des centaines d'utilisateurs ont par la suite contribué au projet [11].
En février 1995, Apache n'était encore constitué que du daemon HTTP (appartenant au domaine public) développé par Rob McCool, au National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign. Le 1er décembre 1995, Apache 1.0 était mis à la disposition du public.

A. Les raisons de ce choix

D'après une étude du cabinet Netcraft sur le marché des serveurs Web publiée en avril dernier, Apache équipe 56% des serveurs Web dans le monde, avec un gain de 10% sur un an. Son seul concurrent sérieux, avec 24% des parts du marché, est IIS de Microsoft. Netscape vient en troisième position mais reste néanmoins la plate-forme la plus utilisée pour les transactions chiffrées. D'autres solutions telles que NCSA sont en voie de disparition. Le cabinet d'analyses Giga Informations Group recommande d'ailleurs Apache pour les utilisateurs soucieux de qualité sur plates-formes Intel [1].
Il est à noter qu'Apache est adopté par les grands noms du monde Internet comme IBM, qui l'a intégré à son offre de serveur d'applications Websphere, ou Apple qui l'utilise dans son Mac OS X. Par ailleurs, les développeurs de NCSA HTTPd[4] annoncent que ce serveur ne sera bientôt plus développé, et conseillent de se tourner vers Apache.

B. Principe de fonctionnement d'Apache

➢ Le mode inetd

Le serveur est configuré pour utiliser le daemon Unix (ou Linux) inetd qui écoute sur l'ensemble des ports avec lesquels il doit travailler. Lorsqu'une connexion s'établit, il détermine à partir de son fichier de configuration `/etc/inetd.conf`, à quel service le port de connexion correspond, puis lance le programme requis. En mode inetd, le serveur sera exécuté à partir du processus système inetd. Pour chaque connexion http demandée, une nouvelle instance du serveur est créée, le programme tournant une fois la connexion établie. La commande nécessaire au démarrage du serveur devra être ajoutée au fichier `/etc/inetd.conf`.

[4] page d'accueil : NCSA HTTPd Home Page - http://hoohoo.ncsa.uiuc.edu/index.html.

➢ Le mode standalone

Dans ce mode, qui a été choisi dans notre cas pour des raisons de performances, le serveur est lancé en tant que daemon : il n'est démarré qu'une fois, et dessert toutes les connexions ultérieures. La commande de démarrage du serveur sera soit ajoutée aux scripts de démarrage du système d'exploitation, soit lancée manuellement.

Lorsqu'on démarre Apache sans le paramètre -X (ce qui est le cas), plusieurs exemplaires fils inutilisés du serveur sont également lancés, afin que toute requête entrante puisse être immédiatement satisfaite. Ceci est géré à l'aide de cinq directives : MaxClients, MaxRequestsPerChild, MaxSpareServers, MinSpareServers et StartServers (voir page 25). Il n'est alors pas nécessaire de deviner combien de serveurs fils il faut lancer, Apache s'adaptant dynamiquement à la demande, mais en fonction de ces directives.

C. Mise en place du serveur Apache

1. Installation

➲ Dans `/usr/local`, copier `apache_1.3.6.tar.gz`, le décompresser et désarchiver. Renommer le répertoire créé (`apache_1.3.6`) sous le nom "`apache`".

```
% gunzip apache_1.3.6.tar.gz | tar xvf -
% mv apache_1.3.6 apache
```

➲ Aller dans `/usr/local/apache/src`, lire le fichier INSTALL, qui contient les consignes de configuration. Editer le fichier `Configuration`, et sélectionner les modules désirés, soit presque l'ensemble des modules d'Apache. Le rajout d'un module s'effectue par invalidation d'une ligne de commentaire, en supprimant '#').

Remarque - créer un exécutable avec la commande make

L'objectif de la commande `make` est de permettre la construction d'un fichier étape par étape. Lorsque le programme exécutable est issu d'un grand nombre de fichiers, ce système permet de modifier les fichiers adéquats et de reconstruire le binaire sans avoir à tout recompiler. "`make`" est un programme généraliste et "intelligent" : il construit des cibles (programme exécutable, document PostScript, etc.) à partir d'éléments (code source C, texte TEX, etc.), et ne recompilera un fichier objet que si son fichier source a été modifié. Le fichier utilisé par la commande "`make`" pour créer l'exécutable est appelé Makefile.

```
% ./Configure
```

Le script Configure utilise les fichiers `Configuration` et `Makefile.tmpl` ("`tmpl`" pour template) afin de construire un Makefile opérationnel.

```
% make
```

La commande make compile un fichier exécutable `httpd`.

Si l'on effectue un test par `% ./httpd`, on obtient le message : "`could not open document config file .../httpd.conf`". Ce qui est normal puisque pour l'instant le fichier de configuration `httpd.conf` n'existe pas encore (à ne pas confondre avec le fichier `Configuration`).

➲ Création de l'arborescence du site

Créer le répertoire `/usr/www/fsite` (répertoire d'accueil du site), se placer dans `../fsite` et créer les répertoires `conf` (contient les fichiers de configuration), `htdocs` (contient l'arborescence des documents du site) et `logs` (contient les fichiers d'erreurs).

Se placer dans le répertoire `apache/conf` et effectuer la copie des fichiers de configuration du serveur, dans l'arborescence nouvellement créée.

```
% cp httpd.conf-dist srm.conf-dist access.conf-dist
    /usr/www/fsite/conf
```

Puis éditer ces fichiers pour les paramétrer.

2. Configuration

La configuration d'Apache se fera en deux étapes. Dans un premier temps, l'objectif sera d'obtenir rapidement un serveur en bon état de marche et susceptible d'accueillir les applications voulues. Dans un deuxième temps, un paramétrage plus fin est envisagé, probablement au cours du CDD de deux mois prévu à la suite du stage.

Apache est entièrement paramétrable à l'aide de fichiers textes de configuration. Dans la configuration standard, le serveur utilise d'abord `httpd.conf`, puis `srm.conf`, et enfin `access.conf`. Le premier définit les attributs généraux du serveur (tels que le numéro de port ou l'utilisateur sous lequel il fonctionne). Le second définit la racine de l'arborescence de documents, et des fonctions spéciales telles que l'analyse du HTML effectuée par le serveur, l'analyse d'implantations d'images internes, etc. Enfin, `access.conf` concerne les différents cas d'accès. Cependant, les deux derniers fichiers sont maintenant distribués vides, car il est recommandé que toutes les directives soient conservées dans un même fichier, pour des raisons de simplicité.

Tous les éléments de paramétrage d'Apache décrits par la suite sont donc issus du seul fichier de configuration `httpd.conf`.

Les directives de configuration d'Apache sont regroupées en trois sections :

- Les directives qui contrôlent l'environnement global du serveur apache.
- Les directives qui définissent les paramètres du serveur principal (ou du serveur par défaut), répondant aux requêtes non issues d'hôtes virtuels.
- Les directives qui définissent les paramètres relatifs aux hôtes virtuels. Grâce aux hôtes virtuels, les requêtes Web peuvent être envoyées à différentes URL, mais seront traitées par le même processus serveur. Il ne sera pas fait ici usage d'hôtes virtuels.

L'ensemble de ces directives est disponible en français à l'adresse :

 http://www.eisti.fr/eistiweb/docs/apache/manual/mod/core.html

Remarque - directives explicitées dans le rapport

De nombreuses directives ont été laissées à leur valeur par défaut et ne seront pas reportées ici. Il ne sera fait état dans ce mémoire que de celles ayant été modifiées et/ou éclairant un aspect particulier de la configuration d'Apache (sur la sécurité par exemple).

- Environnement global

`ServerType standalone`	Cette directive définit comment le serveur est exécuté par le système d'exploitation. Elle accepte deux options : standalone[5] ou inetd[6].
`ServerRoot "/usr/local/apache"`	Répertoire principal sous lequel les fichiers de configuration, d'erreur et log sont stockés.
`PidFile logs/httpd.pid`	Fichier dans lequel le serveur enregistre son PID (Proccess Identification) lorsqu'il est lancé.

[5] Le mode standalone est le plus utilisé car il offre de meilleures performances. Dans le cas d'un site très sollicité, le mode standalone sera certainement la seule solution possible.

[6] Ce mode est parfois préféré pour des raisons de sécurité malgré le coût important en ressources pour chaque connexion : ni le mode standalone ni le mode inetd ne peuvent assurer une sécurité totale, mais ce dernier étant nettement moins utilisé, a par conséquent moins de chance de subir des attaques. Cependant, certains paramètres avancés d'Apache ne sont pas compatibles avec ce mode, qui risque d'ailleurs de ne plus être supporté dans les versions ultérieures.

`ResourceConfig conf/srm.conf /dev/null` `AccessConfig conf/access.conf /dev/null`	Ces deux directives indiquent au serveur de ne pas prendre en compte `srm.conf` et `access.conf` (voir ci-dessus).
`Timeout 300`	Temps d'attente maximum (en secondes) du serveur pour recevoir une requête, qui s'applique pour chaque bloc de données transféré plutôt que pour l'ensemble du transfert. Ce qui autorise le téléchargement de fichiers importants avec une connexion lente, sans interrompre le transfert.
`KeepAlive On`	Après que l'utilisateur se soit connecté sur le site, il y accèdera probablement de nouveau par la suite. Cette directive, paramétrée sur "on", permet de maintenir la connexion persistante, et ainsi de gagner du temps.
`MaxKeepAliveRequests 100ı`	Nombre maximum de requêtes autorisées durant cette connexion persistante. Sur 0, le nombre de requêtes est illimité.
`KeepAliveTimeout 15`	Temps maximum (en secondes) entre deux requêtes, lors d'une connexion persistante. La directive s'applique dès que la requête a été reçue.
`MinSpareServers 5`	Apache vérifie périodiquement le nombre de serveurs fils en attente de connexion. Si celui-ci est inférieur à `MinSpareServers`, de nouveaux serveurs fils sont lancés, au rythme d'un par seconde.
`MaxSpareServers 10ı`	Représente le nombre maximum de serveurs fils en attente de connexion. Sauf nécessité, il est préférable ne pas utiliser des valeurs élevées pour ces deux dernières directives, afin de ne pas épuiser inutilement les ressources.
`StartServers 5`	Nombre de serveurs fils lancés au démarrage. Selon les auteurs de "Apache - The definitive Guide" (Cf. bibliographie), des serveurs traitant un million de contacts par jours fonctionnent bien avec `MinSpareServers` et `MaxSpareServers` paramétrés respectivement à 32 et 64. La performance au démarrage peut être optimisée en situant `StartServers` entre `MinSpareServers` et `MaxSpareServers`.
`MaxClients 150`	Nombre de serveurs pouvant s'exécuter en même temps.
`MaxRequestsPerChild 30ı`	Chaque exemplaire fils d'Apache traite ce nombre de requêtes et meurt. Lorsque `MaxRequestsPerChild` est paramétré à 0, le processus dure jusqu'au redémarrage de la machine ; ce qui est à éviter pour des raisons de sécurité.

- Configuration du serveur principal

Certaines directives appartenant à cet ensemble ont été regroupées dans un chapitre suivant relatif à la gestion de la sécurité.

`Port 80`	Le port écouté par le serveur en mode standalone.
`ServerAdmin lienard@dijon.inra.fr`	Lorsqu'une erreur survient sur le serveur suite à une requête, il est possible d'envoyer un courrier électronique à l'adresse spécifiée par cette directive.
`ServerName http://yyy.yyy.yyy.yyy/`	URL du serveur. L'INRA marche avec un système d'alias[7]. Le numéro IP yyy.yyy.yyy.yyy a pour alias le numéro xxx.xxx.xxx.xxx. Ce dernier permet de joindre le serveur depuis un poste situé au sein du centre de Dijon, alors que le premier est utilisé par tout poste à l'extérieur du centre.
`DocumentRoot`	Répertoire principal contenant l'arborescence de documents.

[7] Le système d'alias est explicité à la fin de ce mémoire dans le chapitre relatif aux considérations générales sur la sécurité.

`"/usr/www/fsite/htdocs"`	Chacun des répertoires auquel Apache a accès peut être configuré selon son rôle au sein du site.

A la demande du responsable de la sécurité au sein du centre de Dijon, les deux numéros IP attribués à la plate-forme Linux demeureront masqués : le numéro routable sera représenté par `yyy.yyy.yyy.yyy`, et le numéro non routable par `xxx.xxx.xxx.xxx`.

3. Exploitation

➲ Lancement du serveur Apache par pointage de l'exécutable `httpd` vers le site `fsite`

```
% cat > /usr/local/go
/usr/local/apache/bin/httpd -f /usr/www/fsite/conf/httpd.conf
^D
% chmod +x /usr/local/go
% go
```

L'exécutable go ne pourra être lancé de la sorte que si son chemin d'accès (`/usr/local`) a été mentionné dans la variable d'environnement PATH (voir le paragraphe *Fichiers de configuration des interpréteurs* dans ce chapitre).

Pour vérifier qu'Apache est bien lancé en arrière plan :

```
% ps aux | grep httpd
```

```
USER  PID   %CPU %MEM SIZE RSS TTY STAT START TIME COMMAND
Root  1365  0.0  1.4  1744 936 ?   S    17:11 0:00 /usr/local/httpd -f...
```

Apache est lancé en fond. Pour l'arrêter, utiliser :

`% kill 1365` (1365 correspond dans cet exemple à l'identifiant du processus, ou PID).

➲ Redémarrage du serveur

Arrêter puis redémarrer Apache peut s'avérer utile, par exemple pour modifier le fichier de configuration principal[8]. On obtient un élégant redémarrage avec le paramètre -16. Il laisse les processus fils fonctionner jusqu'à la fin, terminant toutes les transactions en cours, puis il relit les fichiers de configuration et redémarre le processus principal :

```
% kill -16 PID
```

Si l'on désire modifier les fichiers de configuration, une alternative au redémarrage est d'utiliser le mécanisme `.htaccess`. Les lignes suivantes sont à rajouter au fichier `httpd.conf` :

```
<Files .htaccess>
order allow, deny
deny from all
</Files>
```

- Les directives de configuration peuvent être stockées un fichier secondaire sauvegardé sous `../htdocs`. Ce fichier, contrairement au fichier principal qui est lu par Apache au démarrage, est lu à chaque accès. Ce système offre une grande flexibilité, puisque l'administrateur peut l'éditer lorsqu'il le désire sans arrêter le serveur. En revanche, cela nuit sérieusement aux performances, car le fichier `.htaccess` doit être analysé à chaque requête. Ce bloc de directives limite l'accès à ce fichier au seul administrateur.

```
AccessFileName .htaccess
```

- Le fichier .`htaccess` peut être renommé en changeant cette directive.

Il est également possible, si plusieurs personnes administrent leurs propres pages d'accueil mais n'ont pas les droits pour modifier le fichier de configuration principal, de créer un fichier `.htaccess` pour

[8] C'est-à-dire le fichier `httpd.conf`. Les fichiers de configuration secondaires sont évoqués ci-après.

chacune. La directive `AllowOverride` permet à l'administrateur de limiter les directives autorisées dans ces fichers `.htaccess`.

➲ Paramètres acceptés par httpd.

`-d` : indique un autre ServerRoot initial.
`-f` : indique un autre fichier ServerConfig.
`-v` : montre le numéro de version.
`-h` : liste les directives.
`-X` : lance un seul exemplaire d'Apache. A n'utiliser que pour le débogage.

D. Gestion de la sécurité du serveur

1. Attribution des permissions

- Une mesure de sécurité élémentaire consiste à bien gérer les permissions des différents répertoires et fichiers :

```
cd /usr/local/apache/
chown 0 . bin logs
chgrp 0 . bin logs
chmod 755 . bin logs
```

Procéder de même avec `/usr/www/fsite/conf` et `/usr/www/fsite/cgi-bin` :

```
cd /usr/local/apache/bin/httpd
chown 0 httpd
chgrp 0 httpd
chmod 511 httpd
```

- L'accès aux scripts CGI doit être interdit en lecture pour ne pas permettre aux clients du Web de déceler les failles de sécurité.

2. Utilisation des directives du fichier de configuration `httpd.conf`

➢ Lancement du serveur

```
User lienard
Group doc
```

- Lorsque Apache est lancé (en tant que *root*), il se connecte au réseau et crée des copies de lui-même. Ces copies sont alors toutes lancées sous l'utilisateur spécifié par les directives *User* et *Group*, lequel ne doit pas avoir de privilèges particuliers. Par la suite, seul le processus originel demeure sous l'identité de *root*. Il a pour rôle de gérer les processus fils, démarrant de nouveaux processus ou arrêtant les anciens selon les besoins. Ce sont ces processus fils qui traitent les requêtes adressées au serveur. J'ai choisi mon compte, qui ne possède aucun privilège, et ne contient pas de documents importants, puisque j'ai toujours travaillé en tant que *root*.

➢ Restrictions appliquées à tout ou partie de l'arborescence de documents

```
<Directory /usr/www/fsite/htdocs>
    AllowOverride None
    allow from xxx.xxx
    …
    allow from zzz.zzz
    deny from all
</Directory>
```

- Apache dispose de directives de blocs qui se présentent sous la forme : `<directive de blocs> ensemble de directives </directive de blocs>`. La directive `Directory` permet de n'appliquer l'ensemble de directives qu'au répertoire ou groupe de répertoires

spécifiés. L'emploi des expressions régulières[9] pour décrire le nom du répertoire est possible, il suffit de faire précéder l'expression par le tilde.

- Ce bloc, concernant l'ensemble de l'arborescence du site, est très restrictif par défaut : la directive `AllowOverride none` indique qu'aucune directive de `.htaccess` ne peut surcharger les directives précédentes (Cf. mécanisme `htaccess` page 27).
- Enfin, les directives `allow` et `deny from` sont utilisées ici pour restreindre l'accès du serveur aux numéros IP correspondant aux machines de l'INRA (ce qui en soit, n'empêchera pas le "hacker" de passer outre, mais arrêtera l'utilisateur moyen). L'ordre dans lequel les commandes `allow` et `deny` apparaissent dans le fichier ne détermine pas leur ordre d'application, l'ordre par défaut est `deny` puis `allow` : la commande `allow` s'applique dans ce cas en dernier, et l'emporte sur `deny`. Ici, l'ordre a été inversé, et c'est la commande `deny` qui l'emporte.

Il peut être pratique de rendre accessible depuis le serveur des fichiers appartenant au répertoire personnel des utilisateurs de la machine. Cette possibilité est susceptible d'être employée prochainement, aussi est-elle décrite ici. L'exemple suivant concerne un utilisateur hypothétique "userX" dont le répertoire de connexion est `/home/userX/` :

```
UserDir /home/userX
<Directory /home/userX>
    Options Indexes
    AllowOverride None
</Directory>
```

- La directive `Option` de ce bloc indique que l'on peut lire la liste des répertoires et fichiers contenus dans le répertoire `/userX/`.

➢ Directives susceptibles d'entraîner des failles dans la sécurité du serveur

La directive `Option` peut être accompagnée de multiples paramètres et est utilisée au sein d'une directive de blocs. Elle n'est donc appliquée qu'à l'arborescence définie par cette directive.

```
Option FollowSymLinks
```

- Dans certains cas, on peut être amené à établir des liens entre fichiers. Un lien dur n'est pas nécessairement la bonne solution : si par exemple on supprime puis on recrée le fichier "`data`", les fichiers liés à "`data`" le seront à l'ancienne version. L'emploi d'un lien symbolique ne pose pas ce problème. Cependant, cela est susceptible d'engendrer des failles dans la sécurité. On peut empêcher de "suivre" un lien symbolique en supprimant l'option `FollowSymLinks` de la directive `Option`.
- Une alternative est d'appliquer l'option `SymLinksIfOwnerMatch` à la directive `Option` : l'accès aux liens symboliques est autorisé seulement si le propriétaire est le même aux deux extrémités du lien.

```
AddHandler server-parsed .shtml
…
Option Includes
```

- Cette directive `AddHandler` (qui ne fait pas partie d'un bloc) active les SSI[10] dans Apache. Dans ce cas, tous les fichiers dont l'extension est "`.shtml`" seront associés aux SSI.
- La directive `Option Includes` indique au bloc dans lequel elle est contenue, que l'on peut avoir des fichiers avec des directives SSI dans l'arborescence.

➢ Server-Side Includes (SSI)

Les SSI sont un bon compromis entre le langage Javascript et les CGI, d'autant plus que la technologie SSI est à présent normalisée et se trouve implantée sur les serveurs HTTP de Netscape et Microsoft. En

[9] Les expressions régulières sont évoquées dans le cadre du langage Perl, page 51.
[10] SSI : Server-Side Includes.

outre, une fois supportée par le serveur HTTP, elle est compatible avec tous les navigateurs puisque le code des SSI est transformé en HTML.

Les SSI permettent par des portions de code incluses n'importe où dans une page HTML, d'introduire des directives qui seront traitées par le logiciel serveur lors d'une requête à la page, avant son transfert au client. Des primitives peuvent être exécutées en temps réel sur le serveur, comme l'affichage de la date ou du nombre de visiteurs, la réalisation de tests conditionnels avant d'envoyer un courrier électronique, ou encore la connexion à une base de données pour y référencer la personne qui vient de nous lire.

Les SSI sont des programmes exécutés par le serveur à la demande d'un client, ces programmes étant directement apportés par le client, puisque le code est inclu dans la page HTML. Autoriser les SSI demande donc soit une surveillance étroite du contenu des fichiers déposés, soit une confiance absolue dans les personnes autorisées à mettre des pages sur le serveur. Une préférence sera donnée aux programmes CGI ; les SSI ne seront donc pas activés.

- Programmes CGI

Un programme CGI est également exécuté sur le serveur suite à une requête. Il détient les droits de l'utilisateur auquel appartient les processus fils du serveur et peut être écrit dans n'importe quel langage supporté par la plate-forme d'hébergement. Mais contrairement aux SSI, un script CGI ne peut être normalement placé sur le serveur que par l'administrateur, pour peu que cet administrateur ait paramétré son serveur un minimum. Le principal danger vient donc des faiblesses du script, qu'un esprit mal intentionné utilisera pour se frayer un chemin dans le système.
Le module suEXEC décrit dans le paragraphe suivant est un produit destiné à sécuriser l'emploi des scripts CGI (et SSI).

`ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"`	Transforme les requêtes d'URL commençant par `/cgi-bin/` en exécution du programme situé dans le répertoire `../cgi-bin`.
`ScriptLog /usr/local/apache/cgi-bin/cgi_err`	Fichier d'historique dans lequel est consigné ce qui se passe suite à l'exécution d'un script cgi
`AddHandler cgi-script .cgi`	Active l'emploi de scripts CGI, et associe l'extension de fichier `.cgi` à un script CGI exécutable[11].

- L'emploi de scripts CGI s'est avéré indispensable à la réalisation des objectifs fixés. Les risques encourus sont cependant minimes, même en l'absence de suEXEC, puisque le champ d'application du serveur s'étendait au départ aux machines de l'INRA mais a été restreint par la suite au centre de Dijon (Cf. page 74).

3. Modules optionnels

Apache Guardian est une application dont le but est de déceler les tentatives d'accès douteuses, et est décrite page 31.

Les modules qui suivent sont cités pour mémoire étant donné leur importance, mais n'ont pas été installés par manque de temps et de connaissances dans le domaine de la sécurité.

a. *Sécurisation des scripts CGI avec suEXEC*

Le module suEXEC est introduit depuis la version 1.2 d'Apache, et a pour objectif de minimiser les risques lorsque l'autorisation est données aux utilisateurs de créer et exécuter leurs propres scripts. Quand un programme CGI ou SSI est lancé, il tourne sous le même utilisateur que le serveur Web. SuEXEC permet à un programme de ce type d'être lancés sous un UID différent de celui de l'utilisateur associé au

[11] Utilisation d'une routine de prise en charge, ou "handler", qui associe un nom d'extension à un nom de "handler".

serveur Web. Un programme "lieur" setuid est appelé par le serveur principal toutes les fois qu'une requête HTTP vise un programme CGI ou SSI tournant sous un autre UID que celui du serveur. Apache donne alors à suEXEC le nom du programme, les UID et GID sous lesquels il doit s'exécuter. Vingt conditions doivent être remplies (sans exception) pour rendre possible l'exécution du programme. Voici par exemple quelques questions pour lesquelles une réponse négative entraîne un échec :

1. Le "lieur" a-t-il été appelé avec un nombre correct d'arguments ?
2. L'utilisateur exécutant le "lieur" est-il un utilisateur valide de ce système ?
3. Cet utilisateur est-il l'utilisateur habilité à exécuter ce programme ?
4. Le programme cible est-il spécifié par un chemin non sécurisé ?
5. Le nom d'utilisateur cible est-il valide ?
6. Le groupe cible est-il un groupe valide ?

Etc.

Cependant, une configuration maladroite de ce module est susceptible d'engendrer de nouvelles failles dans la sécurité. Les concepteurs de suEXEC recommandent donc d'éviter son installation si l'on n'est pas familier avec les problèmes de sécurité liés à la gestion des programmes sous *setuid root*.

Un support est proposé en français à l'adresse :


http://www.eisti.fr/eistiweb/docs/apache/manual/suexec.html

b. *Transactions sécurisées avec Apache-SSL (Secure Socket Layer)*

La fonction SSL (Secure Socket Layer) de transactions sécurisées permet d'avoir des communications sécurisées et cryptées sur 128 bits entre un utilisateur et le site web.
Le protocole SSL est apporté par plusieurs modules (Apache-SSL et mod_ssl), dont l'absence au sein d'Apache s'explique par des complications légales, notamment les restrictions par les Etats-Unis à l'exportation de logiciels cryptographiques. Apache-SSL est basé sur SSLeay/OpenSSL. SSLeay est une implémentation libre du protocole d'encryptage utilisé par Netscape pour son serveur sécurisé et son navigateur (Netscape's Secure Socket Layer). OpenSSL est également un projet "Open Source" visant à faire d'Apache un serveur de haute fiabilité.

Projet Apache-SSL :


Http://www.apache-ssl.org/

Projet OpenSSL :


Http://www.openssl.org/

4. Un script CGI conçu pour Apache : Apache Guardian


http://www.xav.com/scripts/guardian

a. *Présentation*

Apache Guardian est un script réalisé pour les versions 1.1 et supérieures d'Apache. Il prévient le webmaster du site par un e-mail prioritaire lorsqu'un visiteur tente de visualiser un document qui n'existe pas, ou échoue en essayant d'accéder à une zone réservée, ou encore initie une erreur dans un script CGI. Car ces tentatives peuvent faire partie de la stratégie d'un individu qui cherche à déceler une faille dans la sécurité du serveur. Le visiteur à l'origine d'une alerte reçoit un message contenant le code de l'erreur en cause.

Mais la sécurité n'est pas le seul intérêt d'Apache Guardian. Si d'autres sites ont créé des liens vers certaines pages de notre site, et que l'on renomme ces pages par la suite, il s'ensuivra des "error 404 Not found". Ce script est un bon moyen de se rendre compte de l'importance quantitative de ces liens

défectueux (un nombre important de visiteurs "insatisfaits" mérite peut-être que l'on redonne à ces pages leur nom d'origine …).

b. *Installation et configuration*

L'application est installée dans le répertoire : `/usr/local/apache/cgi-bin/guardian/`

➲ Seuls quatre paramètres doivent être adaptés :

`#!/usr/bin/perl`	Emplacement du compilateur Perl
`$email = 'lienard@dijon.inra.fr';`	Adresse de l'administrateur du site
`$main_page = 'http://xxx.xxx.xxx.xxx';`	URL de la page principale
`$mailprog = '/usr/sbin/sendmail';`	Emplacement du programme sendmail

➲ Après un "**`chmod 755 guardian.cgi`**", plusieurs tests sont réalisés en donnant manuellement comme URL au navigateur :

`http://xxx.xxx.xxx.xxx/cgi-bin/guardian.cgi`
`http://xxx.xxx.xxx.xxx/cgi-bin/guardian.cgi?401`
`http://xxx.xxx.xxx.xxx/cgi-bin/guardian.cgi?403`
`http://xxx.xxx.xxx.xxx/cgi-bin/guardian.cgi?404`
`http://xxx.xxx.xxx.xxx/cgi-bin/guardian.cgi?500`

Les caractères situés après le point d'interrogation constituent un paramètre, qui est passé au script CGI : le paramètre "401". Le message découlant du premier test (deuxième ligne) est le suivant :

```
To: lienard@dijon.inra.fr
From: guardian@xav.com (Apache Guardian)
Subject: Guardian Report [Mysterious Reason]

Flags tripped for attempted access to
by .

Visitor linked in from http://xxx.xxx.xxx.xxx/. You may wish to contact the
administrator of http://xxx.xxx.xxx.xxx.

Details follow:

DOCUMENT_ROOT: /usr/www/fsite/htdocs
GATEWAY_INTERFACE: CGI/1.1
HTTP_ACCEPT: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
HTTP_ACCEPT_CHARSET: iso-8859-1,*,utf-8
HTTP_ACCEPT_LANGUAGE: fr
HTTP_CONNECTION: Keep-Alive
HTTP_HOST: xxx.xxx.xxx.xxx
HTTP_REFERER: http://xxx.xxx.xxx.xxx/
HTTP_USER_AGENT: Mozilla/4.05 [fr] (WinNT; I)
PATH: /sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/usr/bin/mh:/usr/local:/usr/local/bin:
/root/bin:/usr/local:/usr/local/bin:/usr/local/bin:
QUERY_STRING:
REMOTE_ADDR: 10.21.210.5
REMOTE_PORT: 1245
REQUEST_METHOD: GET
REQUEST_URI: /cgi-bin/guardian/guardian.cgi
SCRIPT_FILENAME: /usr/local/apache/cgi-bin/guardian/guardian.cgi
SCRIPT_NAME: /cgi-bin/guardian/guardian.cgi
SERVER_ADMIN: lienard@dijon.inra.fr
```

```
SERVER_NAME: http://yyy.yyy.yyy.yyy/
SERVER_PORT: 80
SERVER_PROTOCOL: HTTP/1.0
SERVER_SIGNATURE: <ADDRESS>Apache/1.3.6 Server at http://yyy.yyy.yyy.yyy/ Port
80</ADDRESS>

SERVER_SOFTWARE: Apache/1.3.6 (Unix)
UNIQUE_ID: N5YAFwoV0gYAAAXsUvQ
```

➲ Enfin, le fichier `.htaccess` est modifié en fonction de la localisation du script `guardian.cgi`, et placé dans le répertoire principal du site, `htdocs`.

```
ErrorDocument 401 /cgi-bin/guardian/guardian.cgi?401
ErrorDocument 403 /cgi-bin/guardian/guardian.cgi?403
ErrorDocument 404 /cgi-bin/guardian/guardian.cgi?404
ErrorDocument 500 /cgi-bin/guardian/guardian/guardian.cgi?500
```

Ces lignes associent chaque type d'erreur à la partie du script adéquate.

5. Considérations sur les mesures de sécurité prises

L'objectif de ce stage ne portant pas sur la sécurité proprement dite, cette gestion de la sécurité reste sommaire. Des considérations générales sur le problème de la sécurité seront abordées à la fin de ce mémoire. Par ailleurs, une documentation abondante sur le sujet est disponible sur Internet. Sont reportés ici quelques documents jugés incontournables :

- le CNRS a publié sur le Web un document qui essaie de faire le tour de la question sur les aspects de la configuration d'Apache entrant en jeu dans les problèmes de sécurité. Le document, dont une version PostScript est proposée, est accessible à l'adresse :
  ☞ Http://www.urec.cnrs.fr/cours/securite/Apache/plan.html
- Les FAQ (Frequently Asked Questions) sur la sécurité au sein du World Wide Web sont disponibles à l'adresse :
  ☞ http://www.w3.org/Security/Faq/
- Consignes de sécurité pour la configuration du serveur Apache, issues directement de la documentation du serveur.
  ☞ http://www.apache.org/docs/misc/security_tips.html

V. Un projet centré sur les besoins des utilisateurs

A. Comprendre ces besoins : le point de vue utilisateur

Les fondements de ce site ont été définis selon des objectifs bien précis (mise à la disposition du fonds documentaire, présentation des unités de recherche, du centre et de ses chercheurs, …) qu'il est nécessaire de toujours garder à l'esprit. L'administrateur est amené, sur la base d'une réflexion personnelle, à mettre en place des applications axées sur la politique générale du site.

L'un de ces objectifs est la mise de l'information à la disposition de l'utilisateur : de toute l'information jugée utile (dans la mesure des possibilités), et cela par les moyens les plus pertinents. Le point de vue utilisateur devient alors un paramètre indissociable d'une évolution du site adaptée aux besoins. Des remarques ponctuelles peuvent être envoyées à cet effet à l'administrateur via un hyperlien présent sur la page d'accueil. Cependant, dans le cas d'évolutions de plus grande envergure, un nombre important d'acteurs est susceptible de participer à ce qui deviendrait alors un débat, débat parfaitement gérable à

l'aide d'un forum de discussions. Ce forum peut d'ailleurs être à l'origine de développements spontanés, une idée lancée pouvant en entraîner une autre.

1. Meep!Board 1.0

Http://www.meep.com/product/meepboard

Ce forum[12], le premier installé sur le serveur, avait été choisi pour sa simplicité d'installation et d'utilisation. Par la suite, la découverte d'UltraBoard et de toutes les possibilités supplémentaires offertes a justifié le remplacement de Meep!Board. Ce dernier présente cependant des caractéristiques intéressantes ; il a donc été "rangé" à un emplacement du serveur accessible à l'administrateur du site.

Meep!Board a la possibilité de se scinder en plusieurs forums indépendants, pour peu que l'on dédouble certains fichiers et qu'on assigne à chacun les bons paramètres (avec UltraBoard, tous les forums sont accessibles à partir d'un même écran, et la création d'un nouveau forum se fait par les outils d'administration prévus à cet effet, sans nécessiter la réinstallation de fichiers de configuration). Au sein d'un forum, la réponse à un message ne se fait que sur un niveau (la réponse, et la réponse à cette réponse seront au même niveau). De plus, la modification d'une réponse par son auteur n'est pas possible.

Bien que l'installation et la configuration de Meep!Board aient été réalisées avec le même soin que pour UltraBoard, elles ne seront pas reportées ici. Ceci afin de ne pas surcharger le rapport, puisque cette application n'a pas été retenue par la suite.

2. UltraBoard 1.61

http://www.ultrascripts.com/

a. *Présentation*

Ce BBS a reçu de la part de ses utilisateurs des appréciations parmi les meilleures et les plus nombreuses de sa catégorie. Il s'est avéré, dans notre cas, répondre largement à nos attentes. Son format de téléchargement unique, Zip, nécessite l'emploi d'un extracteur sous Linux, tel que UnZip. Il est disponible à l'adresse :

ftp://ftp.cdrom.com/pub/infozip/UNIX/LINUX
http://www.cdrom.com/pub/infozip/

InfoZip autorise l'extraction d'archives .zip (UnZip 5.4 - 28 Novembre 1998) et la compression ou l'archivage au format zip (Zip 2.2 - 3 Novembre 1997) sous Unix/Linux.

b. *Installation et configuration*

Après l'installation de l'arborescence d'UltraBoard dans le répertoire `/cgi-bin`, l'exécution du script `Setup.pl` crée une page HTML destinée à faciliter le paramétrage du BBS (pas d'édition de fichier de configuration, il suffit de remplir un formulaire).

- *Paramètres de configuration*

CGI path	`/usr/local/apache/cgi-bin/UltraBoard`
Database path	`/usr/local/apache/cgi-bin/UltraBoard/UBData`

[12] Il est utile de connaître l'appellation anglaise du forum de discussion, en particulier lorsqu'on recherche des forums dans des banques de scripts: ils sont répertoriés sous la dénomination BBS (Bulletin Board System).

Members path	`/usr/local/apache/cgi-bin/UltraBoard/UBData/Members`
Members session path	`/usr/local/apache/cgi-bin/UltraBoard/UBData/Sessions`
Stats path	`/usr/local/apache/cgi-bin/UltraBoard/UBData/Stats`
Site URL	`http://xxx.xxx.xxx.xxx`
CGI URL	`http://xxx.xxx.xxx.xxx/cgi-bin/UltraBoard`
Images URL	`http://xxx.xxx.xxx.xxx/images/Images`
Email address	`Lienard@dijon.inra.fr`
Use SendMail	`SendMail location : /usr/lib/sendmail`
Date format	`European format (DD-MM-YYYY)`
Time format	`24 hour time format`

➢ *Options*

Sont décrites ici les options paramétrées lors de l'installation du forum.

- Possibilité de fermer temporairement le forum (mise à jour …) et de prévenir les utilisateurs.
- Utilisation de la fonction `lock()` qui permet d'éviter les perturbations lorsque plusieurs personnes postent en même temps.
- Statistiques (sur l'accès horaire, journalier, mensuel).
- Fichier `log` relatif aux personnes accédant au forum (le nombre maximum de lignes, ou celui de visiteurs, est paramétrable).
- Restriction d'accès au forum par le numéro IP ou le nom de domaine.

Beaucoup d'autres options de paramétrage sont offertes dans la section Administration, une fois le forum mis en place.

➢ *Profil administrateur*

Un compte administrateur (username, password, nick name, email) est créé lors de l'installation.

c. ***Exploitation***

L'accès au forum se fait par login et mot de passe.

➢ *Section Administrateur*

Cette section donne de nombreuses informations sur le forum :

- Informations générales : nombre de membres, discussions, période de trafic la plus élevée (dans la journée, dans la semaine), navigateur le plus utilisé, etc.
- Liste des 20 derniers visiteurs du forum.
- Informations sur les 20 dernières actions effectuées sur le forum.
- Même chose concernant les 20 dernières actions de l'administrateur.

L'administrateur dispose en outre de six axes de gestion :

1. Gestion générale
   - Options de configuration du système (paramètres entrés lors de l'installation du forum)
   - Options générales de configuration (une cinquantaine de paramètres régissant l'aspect et le fonctionnement général d'UltraBoard)
   - Options de configuration de style (68 paramètres pour configurer en détail polices et couleurs de caractères, aspect des différents fonds, des boîtes de dialogue, etc.)
   - Définition des icônes employées
   - Modification du profil administrateur
2. Gestion des groupes
   - Création, Modification, Suppression. Un nouvel utilisateur sera classé au choix dans le groupe modérateur ou dans l'un des groupes d'utilisateurs.
   - Remarque : lors de la création d'un groupe d'utilisateurs, un `Group ID` est demandé. Il s'agit en fait d'une chaîne de caractères par laquelle le script identifiera le groupe.
3. Gestion des comptes utilisateur
   - Création, Modification, Suppression, Déplacement, Activation/Désactivation, Restriction d'accès à un ou plusieurs forums, à une ou plusieurs catégories.
4. Gestion des forums

Création, Modification, Suppression, Réorganisation de l'ordre des forums : une scission d'UltraBoard en autant de forums que l'administrateur le désire est possible. Les différents forums apparaissent les uns à la suite des autres.

Lors de la création d'un forum, l'administrateur peut autoriser ou interdire l'accès à chacun des groupes.

5. Gestion des catégories de discussions

   Création, Modification, Suppression, Réorganisation de l'ordre des catégories (chaque forum peut être subdivisé en grandes catégories de discussions). Chaque catégorie peut être autorisée ou interdite en lecture ou en postage pour chaque groupe. La catégorie peut dépendre d'un groupe de modération (groupe administrateur ou groupe modérateur) ou non.

6. Gestion des messages

   Nouveau message, Suppression, Déplacement, Modification, Fermeture/Ouverture

   Enfin, une recherche complexe dans la section désirée est possible pour retrouver un groupe, un utilisateur, un forum ou une catégorie.

   Chaque message est un fichier texte portant l'extension "`.post`", et se trouve stocké dans le répertoire `/cgi-bin/UltraBoard/UBData/test/`.

➢ *Utilisateurs*

- L'utilisateur poste ses messages dans la catégorie souhaitée (et autorisée). La réponse à un message peut être modifiée (par exemple rajout à la fin de la réponse) par son expéditeur, et le nombre de modifications apportées à un message est indiqué.
- La recherche de mots au sein d'une ou plusieurs discussions à la fois est possible. De nombreuses options sont disponibles, comme limiter la recherche au corps du message, à son titre, au nom de l'expéditeur, à la date d'expédition etc.
- L'utilisateur peut à tout moment modifier son profil (password, fiche descriptive etc.).

➢ *Messages*

Les messages sont affichés par ordre ascendant ou descendant basé sur le nom de l'expéditeur, ou le nombre de réponses, la date de dernière modification, ou enfin le titre du message. L'affichage pourra se limiter aux messages datant d'un nombre de jours donné.

d. *Conclusion*

Les multiples options fournies par UltraBoard ne peuvent être toutes détaillées, et font de ce logiciel libre un système de gestion de forums très performant et entièrement paramétrable (par formulaire HTML, ce qui est très pratique). Sur une dizaine de forums (libres) testés en démonstration, UltraBoard semble être de loin le meilleur. Petit reproche, l'absence de documentation, qui oblige à découvrir le fonctionnement du forum par soi-même. Il est également dommage que la gestion de l'archivage des messages soit réduite à sa plus simple expression (stockage d'un fichier par message dans un répertoire). Cependant, il est prévu de pallier cet inconvénient en indexant ces messages au moyen de freeWAIS-sf.

B. Evaluation de l'utilité des services proposés grâce à l'outil statistique : Analog

http://lowrider.hili.com/analog/
http://www.gekko.de/analog/ (site miroir en Allemagne, celui français ne répondant pas)

L'outil statistique est un moyen de rendre compte indirectement de la qualité des services proposés ou de l'information apportée. Les données issues d'outils statistiques tels qu'Analog, à interpréter bien sûr avec prudence, sont susceptibles d'orienter l'évolution du site vers une amélioration et une croissance rationnelle. Une section très demandée pourra faire l'objet de développements ultérieurs ; au contraire, des pages demeurant dans l'oubli peuvent révéler soit un manque de pertinence de l'information présentée, soit de mauvaises voies d'accès à cet information.

1. Présentation d'Analog 3.31

Analog est un programme libre écrit en C, qui analyse les fichiers `log` d'un serveur, afin de déterminer les pages les plus visitées, depuis quels pays elles sont visitées, etc. La dernière étude du GVU (Graphics Visualization and Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA), menée du 10 Octobre au 15 Décembre 1998, montre qu'Analog est l'analyseur de fichiers `log` le plus utilisé au monde. Sur 434 webmasters interrogés, 24.9% se servaient d'Analog, contre 20.3% pour son concurrent le plus proche.

Très rapide (traitement de 1 Go en 5 minutes avec un processeur 266 MHz), Analog peut analyser des fichiers `log` de plus de 25 Go sans problème, dans pratiquement n'importe quel format (formats NCSA et dérivés, Microsoft IIS, Netscape, WebSTAR, Netpresentz etc.).

2. Installation et exploitation

➲ Télécharger `analog3.31.tar.gz`, le décompresser puis le désarchiver :

```
$ gunzip analog3.31.tar.gz | tar xvf -
```

et placer `/analog3.31/` dans le répertoire `/usr/local/`.

Les instructions se trouvent dans `/analog3.31/docs/Readme.html`.

➲ Editer `/analog3.31/analog.cfg` et effectuer les modifications pour avoir les lignes suivantes :

```
LOGFILE /usr/local/apache/logs/access_log
OUTFILE /usr/www/fsite/htdocs/stats/access_log.html
```

Pour produire un fichier log dont le nom comporte par exemple le mois et l'année, utiliser la ligne :

```
OUTFILE /usr/www/fsite/htdocs/stats/fichier%M%y.html
```

Ce qui aura pour effet de créer le fichier log : `fichier0599.html` (pour un fichier créé en mai 1999).

➲ Créer `/usr/local/etc/httpd/` puis `/usr/local/etc/httpd/analog/`, et y placer le fichier `analog.cfg`.

➲ Copier le répertoire `/usr/local/analog3.31/lang` dans le répertoire `/usr/local/etc/httpd/analog/`.

➲ Il suffit maintenant pour lancer `Analog` de taper les commandes suivantes :

```
$ cd /usr/local/analog3.31
$ ./analog
```

3. Paramétrage

Les paramètres de configuration ne peuvent être décrits dans leur intégralité.
De nombreuses lignes peuvent être rajoutées au fichier de configuration `analog.cfg` pour adapter Analog aux besoins du site, par exemple :

`HOSTEXCLUDE mycomputer.myisp.com` `FILEEXCLUDE répertoire/*`	Permet d'exclure les utilisateurs locaux, ou de ne pas analyser certains fichiers ou répertoires.
`CONFIGFILE other.cfg`	Permet d'inclure un autre fichier de configuration.

`LOGFORMAT format`
 	
`APACHELOGFORMAT`
`(%h %l %u %t \"%r\" %s %b)`
	Dans la très grande majorité des cas, Analog détecte automatiquement le format du fichier log à analyser. Dans le cas contraire, il est possible de spécifier un format non reconnu par cette commande.
	
	Uniquement dans le cas d'un serveur Apache. Le format est décrit par une syntaxe spécifique, par exemple :
	%h : heure du jour,
	%u : user,
	%b : nombre d'octets transférés, etc.

`CACHEOUTFILE fichiercache`	Analog peut archiver certaines données dans un fichier cache. La perte du fichier log n'entraîne donc plus celle des données les plus importantes.

- *27 rapports sont proposés par Analog.*

Le paramètre ON active la création du rapport, OFF l'inactive. L'activation ou l'inactivation de l'ensemble des rapports se fait par la ligne ALL ON ou ALL OFF. Ci-dessous, un échantillon de ces rapports :

Lignes rajoutées au fichier analog.cfg	**Ecrit dans la page de rapport**
`MONTHLY ON`	Un rapport mensuel
`WEEKLY ON`	Un rapport hebdomadaire
`FULLDAILY ON`	Un rapport journalier
`DAILY ON`	Un résumé journalier
`FULLHOURLY ON`	Un rapport horaire
`HOURLY ON`	Un résumé horaire
`QUARTER ON`	Un rapport tous les quarts d'heure
`FIVE ON`	Un rapport toutes les cinq minutes
`REQUEST ON`	Le nom des fichiers demandés
`FAILURE ON`	Le nom des fichiers qui n'ont pu être envoyés par le serveur
`HOST ON`	Le nom des ordinateurs demandant les fichiers
`DOMAIN ON`	De quel pays proviennent les requêtes
`BROWSER ON`	Quels navigateurs sont utilisés pour accéder au site
`FILETYPE ON`	Le type des fichiers demandés
`SIZE ON`	La taille des fichiers demandés

- *Personnalisation du fichier de sortie - généralités*

`HOSTNAME "l'INRA Dijon"`	Donne le titre : "Stats du serveur Web l'INRA Dijon"
`IMAGEDIR images/stats/`	Le répertoire des `.gif` est placé sous `htdocs/images/stats/`.
`LANGUAGE FRENCH`	Traduit automatiquement l'interface dans la langue spécifiée. 30 langues sont disponibles.

`FILEALIAS /nom1.html /nom2.html`
 	
`TYPEOUTPUTALIAS`
`txt ".txt (Fichier texte)"`
	Utile par exemple pour transcrire des noms d'hôtes locaux en leur équivalent Internet.
	Cet exemple permet d'expliciter le rapport « type de fichier demandés ».

`LOGO images/stats/logo.gif`	Remplace le logo du début de page (aucun logo avec le paramètre `none`).

`SEPCHAR " "`
`REPSEPCHAR none`
`DECPOINT ,`
	Définit le format d'affichage des nombres au sein du rapport : respectivement le séparateur des milliers dans un texte, dans un tableau, et le caractère marquant la décimale. Dans cet exemple, 1250,45 sera écrit "1 250,45" dans un texte et "1250,45" dans un tableau.

➢ *Personnalisation du fichier de sortie - les différents rapports*

Rapports relatifs au temps (MONTHLY, WEEKLY, FULLDAILY, DAILY, FULLHOURLY, HOURLY, QUARTER et FIVE) :

`MONTHCOLS PR`
 	
`(WEEKCOLS, FULLDAYCOLS, etc.)`
	Chacun de ces rapports est contrôlé par des commandes COLS variées :
	R : nombre de requêtes
	r : pourcentage de requêtes
	P : nombre de pages demandées
	p : pourcentage de pages demandées
	B : nombre d'octets transférés
	b : pourcentage d'octets transférés
	Dans cet exemple, le nombre de pages demandées suivi du nombre de requêtes effectuées mensuellement sont affichés.
	Remarque - La commande COLS marche également pour les autres rapports.

`MONTHBACK ON` `WEEKBACK OFF`	Affiche les rapports mensuels en commençant par la date la plus récente, et les rapports hebdomadaires en commençant par la date la plus ancienne.

`MONTHROWS 0`
`QUARTERROWS 60`
`Etc.`
	La commande ROWS permet de n'afficher que les dernières lignes d'une période de temps. Dans ce cas, toutes les lignes du rapport mensuel et seules les 60 dernières lignes du rapport établi tous les quarts d'heure seront affichées.

Concernant les autres rapports :

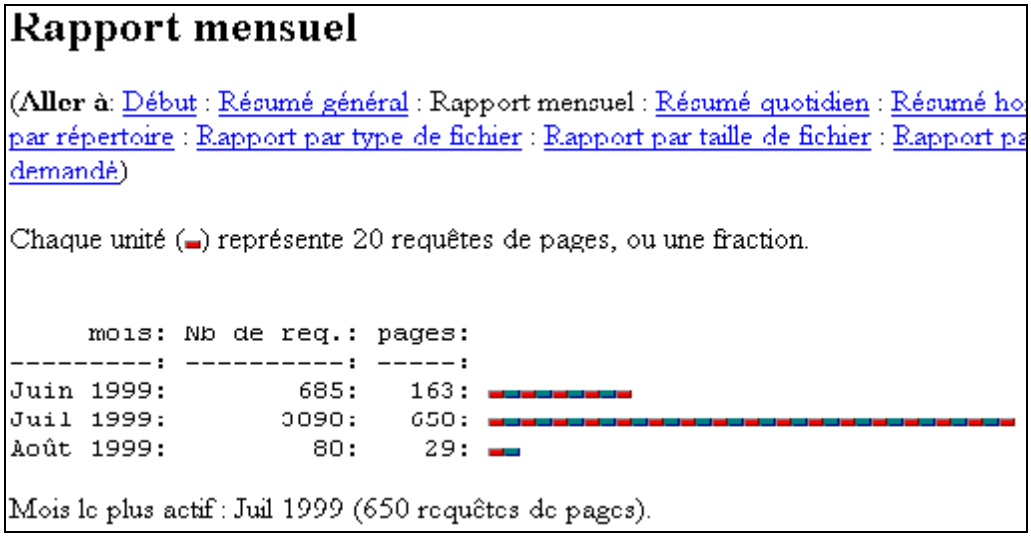
`REQINCLUDE pages`	La liste des pages demandées est fournie, plutôt que le nom des fichiers. Option associée au rapport REQUEST ON
`BARSTYLE h`	Aspect des histogrammes (huit styles possibles, de "a" à "h").

`HOSTSORTBY ALPHABETICAL`
 	
`(REQSORTBY, DIRSORTBY, TYPESORTBY, FAILSORTBY, FULLBROWSORTBY etc.)`
	La commande SORTBY permet de sortir les lignes de rapports selon six critères de tri : ALPHABETICAL, REQUESTS, PAGES (pages demandées), BYTES, DATE et RANDOM (pas de tri - utile en terme de vitesse pour les longs rapports).

`DOMFLOOR 1000r`
`DOMFLOOR 1000p`
`DOMFLOOR 1Mb`
`DOMFLOOR 0.5%r`
`DOMFLOOR 970701d`
`DOMFLOOR -00-01-00d`
`DOMFLOOR -100r`
	La commande FLOOR détermine un seuil minimal en-dessous duquel un item n'est pas pris en compte.
	
	Tous les domaines avec au moins 1000 requêtes
	Au moins 1000 requêtes par page
	Au moins 1 Mo transféré
	Au moins 0.5% des requêtes
	Les derniers accès depuis le 1er juillet 1997
	Les derniers accès depuis le dernier mois
	Les domaines avec les 100 nombres de requêtes les plus élevés.
	Remarque - La commande FLOOR ne marche pas pour la totalité des rapports ; un mauvais emploi est signalé par un message d'avertissement.

`SUBDIR /répertoire/*`
 	
`SUBTYPE *.gz`
	Garantit que l'ensemble des sous-répertoires de `/répertoire` sera prit en compte dans le rapport DIRECTORY.
	Même principe pour le rapport FILETYPE.

➢ *Exemple de rapport : le rapport mensuel*



Rapport mensuel

(**Aller à**: Début : Résumé général : Rapport mensuel : Résumé quotidien : Résumé ho
par répertoire : Rapport par type de fichier : Rapport par taille de fichier : Rapport pa
demandé)

Chaque unité (▬) représente 20 requêtes de pages, ou une fraction.

mois:	Nb de req.:	pages:	
Juin 1999:	685:	163:	▬▬▬▬▬▬▬
Juil 1999:	3090:	650:	▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬
Août 1999:	80:	29:	▬

Mois le plus actif : Juil 1999 (650 requêtes de pages).

➢ *Remarques concernant l'installation d'Analog*

- En lançant Analog par **`./analog -settings>settings.txt`**, on obtient un fichier `settings.txt` contenant une liste détaillée des options activées ou non, des variables utilisées avec leur valeur, et des commandes de configuration.

- On peut noter qu'il existe un jeu de commandes pour lancer Analog avec une quantité faible de mémoire vive, et un autre jeu pour déboguer Analog si besoin est.

4. Protocole d'utilisation

Les mises à jour sont effectuées localement par le daemon cron. Celui-ci lance chaque nuit le programme `/usr/local/analog3.31/analog`, lequel écrit le fichier `access_log.html` dans le répertoire `/usr/www/fsite/htdocs/stats/`.

5. Conclusion

Si l'installation sommaire d'Analog est aisée, un paramétrage fin fait intervenir un nombre considérable d'options : il n'a pas été réalisé par manque de temps. L'application n'en demeure pas moins parfaitement exploitable.
Au-delà de toutes les possibilités offertes par Analog, existe en graticiel sur Internet un interfaçage graphique par l'intermédiaire de Report Magic, qui élabore des courbes, graphiques ou images à partir des données analysées. Report Magic est disponible à l'adresse :

☞ http://www.wadsack-allen.com/digitalgroup/reportmagic/docs/index.html

Ce module fera l'objet d'une installation éventuelle après le stage.

C. Communication et circulation de l'information

L'intérêt d'un forum de discussions au sein d'un Intranet est aisément compréhensible en matière de circulation de l'information. L'idée était d'installer un forum que l'on puisse scinder en différents sous-forums, chacun ciblant un sujet déterminé. Quatre types de sujets ont été envisagés au départ :

- Utilisation de divers logiciels (EndNote, Word, Excel, PowerPoint, …).

- Evolution du site de Dijon (remarques, suggestions, questions, …). Cet aspect a été traité dans le paragraphe A de ce chapitre.
- Vie courante (informations générales, besoins de renseignements, …).
- Sujets ponctuels. Contrairement aux trois forums précédents, ce type de forum n'est destiné à exister qu'à l'occasion de ce qui a justifié sa création.

La réalisation de ce projet a été rendue particulièrement aisée avec UltraBoard, grâce à ses grandes facilités d'administration, et cela entièrement depuis le site.

VI. Accès à l'information

A. Création de bases données WAIS : FreeWAIS-sf et SFgate

1. Remarques préliminaires : norme Z39.50 et applications WAIS

La norme Z39.50

Z39.50 est un protocole d'interrogation de bases de données bibliographiques, documentaires, ou autres, selon un mode client-serveur : un logiciel client Z39.50, détenu par l'utilisateur, s'adresse à un logiciel serveur Z39.50, où se trouve la base. Z39.50 est parfaitement adapté à Internet puisqu'il s'appuie sur le protocole TCP/IP. Un logiciel client Z39.50 connecté à Internet permet ainsi l'interrogation de toute base reliée à Internet mettant en œuvre un logiciel serveur Z39.50. Le cadre de la consultation pure peut être élargi, pour rejoindre celui de l'échange ou de la communication d'informations. Cette communication est rendue possible entre des systèmes qui tournent sur des plates-formes matérielles différentes et utilisent des logiciels différents, puisque la norme est ouverte (norme d'application de réseau) [13].

La norme Z39.50 est une norme nationale américaine connue officiellement sous la dénomination ANSI/NISO Z39.50 — Information Retrieval (Z39.50) : Application Service Definition and Protocol Specification. La plus récente version de la norme Z39.50 a été approuvée en 1995 par la National Information Standards Organization (NISO), seule organisation accréditée par l'American National Standards Institute (ANSI) pour approuver et tenir à jour les normes applicables aux services d'informations, aux bibliothèques et aux maisons d'édition. Elle contient les versions 2 (publiée en 1992, sous le titre ANSI/NISO Z39.50-1992) et 3 de la norme. La version 2 autorise les recherches bibliographiques (principalement les notices bibliographiques au format MARC) et la recherche d'informations. La version 3, dotée de fonctions de recherches beaucoup plus nombreuses, permet entre autres une recherche sur des notices non bibliographiques.
La norme Z39.50 bénéficie d'une reconnaissance mondiale et est devenue une norme internationale en juillet 1998, remplaçant la norme Recherche documentaire approuvée en 1991 par l'Organisation internationale de normalisation (ISO). Sa nouvelle dénomination est ISO 23950. Elle est cependant dite équivalente à la norme ANSI Z39.50 (source Périnorm).

Les possibilités offertes par Z39.50 sont multiples ; de nombreuses fonctions de bibliothèque sont couvertes telles que la consultation de bases de données (bibliographiques, en texte intégral, ou encore d'images), le catalogage, le prêt entre bibliothèques, etc.
Z39.50 offre un ensemble d'une demi-douzaine de grammaires de requêtes, d'une centaine de critères d'accès supportés, de multiples formats de restitution des données : la douzaine de formats Marc, les jeux de caractères habituels (ISO 5426, ALA, ISO 8859, ANSI,), etc. Revers de la médaille, cette richesse fonctionnelle peut être à l'origine d'incompréhension entre un client et un serveur répondant pourtant tous deux à la norme Z39.50. Un logiciel client ou un serveur ne peut en effet pas implémenter toutes ces fonctionnalités. L'emploi de "profils" (définissant les options réellement implémentées dans un logiciel

client ou serveur) pourrait remédier à ce problème. Il existe d'ailleurs des listes de "profils" éditées par les principaux implémenteurs de Z39.50 La solution idéale étant bien sûr de trouver un client Z39.50 universel.

FreeWAIS et FreeWAIS-sf

FreeWAIS est une implémentation domaine public de la version 1988 du protocole Z39.50. Il supporte un sous-ensemble des caractéristiques de la version commerciale de WAIS éditée par WAIS Inc. Son développement est arrêté, seules des corrections de bogues seront encore effectuées. Son remplaçant se nomme Zdist.

Contrairement à FreeWAIS, qui n'est capable que d'indexation en texte intégral, FreeWAIS-sf autorise l'indexation de documents structurés, comme des notices bibliographiques. Il est capable de différencier les champs textuels, numériques et de date, il supporte les requêtes booléennes complexes, les caractères 8 bits pour les langues européennes. Il offre la possibilité de faire des recherches phonétiques ou sur la racine des mots, ainsi qu'une définition simple du format et de la présentation des résultats. Il est cependant à regretter que freeWAIS et freeWAIS-sf n'évoluent pas ensemble.

En complément, le module SFgate, écrit en Perl, permet l'interrogation des bases Wais via une interface Web relativement conviviale, qui offre beaucoup de facilités de paramétrage.

2. Adaptation de la plate-forme Linux aux ressources du site déjà existant

L'orientation générale du projet était de dupliquer le site du serveur de Jouy-en-Josas sur le serveur de l'UPE-Doc, pour ensuite le développer à loisir. Une préoccupation essentielle était alors de pouvoir reproduire les services initialement proposés. Le premier problème résidait dans l'exploitation des données issues de Texto : l'application Texto-Web servant d'interface Web était opérationnelle sur une machine Unix, mais n'était pas encore développée pour Linux. Les recherches entreprises pour solutionner ce problème ont abouti à la conclusion que dans le domaine des logiciels libres, FreeWais-sf se trouvait être le meilleur substitut de Texto-Web. D'autant plus que le module SFgate pouvant lui être adjoint, constituait une interface Web relativement conviviale et entièrement paramétrable.

Malheureusement, la compilation de SFgate a posé de sérieux problèmes. Une aide a pu être obtenue fin août, cependant, il a été envisagé la possibilité que le module SFgate ne puisse être installé. Une solution de remplacement a été cherchée afin d'offrir un accès aux données via le site sans FreeWais-sf. Cette solution passait par un moteur de recherche ; elle est décrite dans le chapitre consacré au moteur Xavatoria.

3. Gestion de nouvelles sources d'informations

Les sources de données initialement destinées à FreeWAIS-sf ont donc été exploitées pour des raisons techniques par le moteur de recherche Xavatoria. L'utilité de poursuivre le travail sur FreeWAIS-sf - outre le fait de disposer d'un logiciel de gestion de données performant et répondant à un standard mondialement reconnu - réside dans la mise en place d'autres bases WAIS durant mon CDD consécutif au stage.

- Il s'agit d'une part d'un système d'indexation des courriers électroniques issus de certaines listes de diffusions. Le logiciel de messagerie installé sous Windows est Eudora Light. Dans son répertoire principal, Eudora place pour chaque boîte aux lettres créée, un fichier texte portant l'extension "`.mbx`". Il est alors extrêmement aisé de récupérer ce fichier contenant l'ensemble des messages de la boîte aux lettres, et de le soumettre à une indexation WAIS.
  Il est évident que le "niveau de qualité" de la base ainsi constituée dépendra de la façon dont le flux d'informations aura été traité. Selon le temps accordé à cette application, les messages pourront être ajoutés tels quels dans la base (par exemple en relevant le courrier toutes les semaines), ou après une

lecture pour ne retenir que les plus intéressants. Ou bien encore après retraitement, pour concaténer par exemple toutes les réponses à un message jugé particulièrement pertinent.
Quoiqu'il en soit, possibilité est offerte, et pour des coûts très réduits, de créer autant de bases WAIS que de listes de diffusion souhaitées, et permettre un accès relativement efficace à l'information, par interrogation mono ou multibases.

- Une application voisine concerne une indexation des messages du forum nouvellement installé. Le système d'archivage des messages d'UltraBoard est en effet très réduit puisque seul un simple stockage des messages est effectué, sans possibilité de recherches. Un petit script Perl peut très facilement concaténer l'ensemble des messages (constituant autant de fichiers) en un unique fichier texte et fournir ainsi à freeWAIS-sf la source de données requise.

Il a été procédé a une application test de FreeWAIS-sf et SFgate avec la création de la base WAIS `bdmail`. Une base de données WAIS est formée par l'association des documents source et de leur jeu d'index. La base `bdmail` a été réalisée à partir du dépouillement des courriers des listes de diffusion sur Linux et Perl auxquelles je suis abonné, et aussi à partir du forum Perl français.

➢ Source des données

La source des données est constituée par fichier texte, `bdmail.txt`, qui réunit tous les courriers électroniques et dont voici un échantillon :

```
NU: 15
CA: linux
TH: message d'erreur
DA: 07/1999
QU: après avoir installé la Mandrake 6, et l'avoir laissée tester (avec
succès) la carte video, KDE semble fonctionner correctement, sauf que
lorsque je quitte KDE et retourne en mode console, je lis le message:
"waiting for X Server to shutdown mach64 Program CPKMach64CT : warning Q
10.66666667"
RE: même problème avec certains postes équipés d'une ATI Expert@Play98,
et ce malgré toutes les vérifications nécessaires (no clock settings, et
video ram ok). D'autres dont la configuration matérielle est strictement
identique ne le font pas. Heureusement, cela n'empêche pas X de
fonctionner. Même problème avec une carte ATI Expert@work98.
```

➢ Création du fichier de description des documents

Le fichier `bdmail.fmt` est construit à partir de `bdmail.txt`. Il décrit la structure des champs au sein des enregistrements. Ce fichier est explicité et reporté ci-dessous :

```
record-sep: /^\n$/
```

Définit le séparateur de documents. Cette expression régulière[13] signifie : ligne vide

```
layout:
 headline: /^DA: / /^[A-Z][A-Z]:/ 7 /DA: */
 headline: /^CA: / /^[A-Z][A-Z]:/ 6 /CA: */
 headline: /^TH: / /^[A-Z][A-Z]:/ 70 /TH: */
end:
```

Définit les lignes de titres, qui décrivent la liste des documents correspondant à une équation de recherche. Chaque ligne est formée de la date du document (sur 7 caractères, puisque le format est MM/AAAA), de sa catégorie (Linux ou Perl), et des 70 premiers caractères du champ TH (thème).

```
region: /^NU: /
 TEXT GLOBAL
end: /^\n$/
```

[13] Voir le paragraphe consacré à Perl et les expressions régulières

```
region: /^NU: /
 nu TEXT LOCAL
end: /^[A-Z][A-Z]:/

region: /^CA: /
 ca TEXT LOCAL
end: /^[A-Z][A-Z]:/

region: /^TH: /
 th TEXT LOCAL
end: /^[A-Z][A-Z]: /

region: /^DA: /
 da TEXT LOCAL
end: /^[A-Z][A-Z]:/

region: /^QU: /
 qu TEXT LOCAL
end: /^[A-Z][A-Z]:/

region: /^RE: /
 re TEXT LOCAL
end: /^[A-Z][A-Z]:/
```

Chaque région définit une portion du document à indexer. La description d'une région suit le format :

```
/expression régulière 1/
portée de l'indexation
/expression régulière 2/
```

Où "`expression régulière 1`" décrit le début de la région à indexer et "`expression régulière 2`" sa fin. La portée de l'indexation peut être globale, auquel cas il n'est pas nécessaire de nommer un champ pour interroger l'index, ou locale. Dans ce cas, les caractères devant la mention "`TEXT LOCAL`" correspondent au nom du champ que l'on utilisera dans la requête. Ce nom peut être différent du nom réel du champ de la base employée.

Par exemple, le champ "th" commence par "TH: " en début de ligne et se termine à la prochaine occurrence de deux lettres majuscules suivies de deux points et d'un espace en début de ligne.

L'emploi des expressions régulières peut sembler rebutant pour une personne non familière du monde Unix ou de Perl, mais permet un paramétrage très souple du format des documents à traiter. Nous les utiliserons beaucoup par la suite, soit au sein de scripts Perl, soit pour de futures applications freeWAIS-sf.

- Indexation par `waisindex` :

  `% waisindex -d bdmail -t fields -export bdmail.txt`

  Signification des options :

  `-d bdmail` : nom du fichier de la base pour les fichiers d'index.

  `-t` : format des fichiers manipulés par `waisindex`. "`-t fields`" est utilisé pour l'indexation de champs.

La construction d'une base de données FreeWAIS-sf par `waisindex` aboutit à la création de plusieurs fichiers :

- **`bdmail.src`** : description de la base de données. Cette description peut être utilisée à deux fins. Premièrement pour fournir les informations de connexion au logiciel client (nom d'hôte, adresse IP, port ou emplacement des fichiers d'index pour une recherche locale). Deuxièmement pour construire une base de données de descriptions. Cette base constitue ainsi un annuaire de bases WAIS (annuaire mondial : "directory-of-servers").

- **bdmail.fn** : pour une base WAIS donnée, contient la liste de tous les fichiers source (de données). Le nom, les dates de modifications et le type de document sont spécifiés.
- **bdmail.hl** : contient l'en-tête de chaque document (cet en-tête est défini dans le fichier bdmail.fmt).
- **bdmail.doc** : tableau des documents de la base de données, avec une entrée par document. Chaque entrée contient un pointeur vers les fichiers bmail.fn et bdmail.hl, le début et la fin du document, le nombre de mots et de lignes, ainsi que la date de création du document.
- **bdmail.cat** : fichier catalogue. Version lisible de bdmail.doc comprenant les noms de fichiers de bdmail.fn et les en-têtes de bdmail.hl. La taille de ce fichier étant souvent très importante, l'option "-nocat" permet de ne pas le générer.
- **bdmail.dct** : dictionnaire global. Contient une entrée par terme valide de la base de données. Le fichier est scindé en deux blocs : le premier contient la première entrée de chaque groupe de 1 000 entrées, et sa position dans le dictionnaire. Le second contient l'entrée de chaque mot de la base, et sa position dans le fichier inversé (voir bdmail.inv). Ce système potentialise la recherche d'un mot dans l'index.
- **bdmail.inv** : chaque entrée est représentée dans ce fichier, avec son poids et ses positions au sein de chaque document.
- **bdmail.stop** : mots rajoutés par waisindex, lorsque leur nombre d'occurrences est trop élevé et pourrait interférer avec le bon fonctionnement de l'index.

➢ Recherche dans la base avec waissearch

Test de la base en local :

% waissearch -d /usr/local/sfbases/bdmail "th=doc*"

Signification de l'option :

-d : spécifie le chemin d'accès à la base WAIS.

Il est bien entendu possible d'interroger la base à distance par une connexion de type telnet. Une fois connecté, l'utilisateur lance le script shell sfmail reporté ci-dessous, qui présente les modalités d'interrogation de la base et attend une requête :

% sh sfmail

```
#!bin/sh
echo
echo "--------------------"
echo
echo "          Vous pouvez interroger les champs suivants : "
echo
echo "    : rentrer directement un mot clé fait porter l'interrogation"
echo "      sur l'ensemble des champs"
echo " ca : catégorie (linux, perl)"
echo " th : thème"
echo " da : date d'émission du message (ex : 06/1999)"
echo " qu : question à l'origine du message"
echo " ex : exemple d'application"
echo " re : réponse à la question"
echo
echo "          La troncature illimitée (*) est disponible"
echo
echo "          Exemple d'interrogation :"
echo
echo " th=doc* and qu=(howtos or FAQ)"
echo
echo "--------------------"
echo "Entrez votre requête"; read req
echo "--------------------"
waissearch -d /usr/local/sfbases/listdiff/bdmail $req
echo
echo
```

Un exemple de requête est fourni par le script : `th=doc* and qu=(howtos or FAQ)`.
L'ensemble des documents correspondant à une requête donnée est d'abord présenté à l'utilisateur, sous la forme d'une ligne numérotée par document, selon un ordre décroissant de pertinence. Une ligne comprend entre autres le score (nombre représentant la pertinence) et l'en-tête décrit dans le fichier `bdmail.fmt`. L'utilisateur a alors le choix de visionner le document en tapant le numéro de ligne correspondant, ou de quitter la consultation pour poser une nouvelle requête.

Une deuxième modalité d'interrogation permet l'interrogation de la base sur le site de Dijon, via SFgate.

- Recherche dans la base à l'aide du module SFgate

SFgate ne fait aucun appel à `waisq` ou `waissearch`, mais se connecte de lui-même au serveur `waisserver`. Cette connexion n'est d'ailleurs même pas nécessaire dans le cas d'une interrogation locale. Dans les autres cas, l'utilisation du module SFgate nécessite au préalable le lancement du serveur `waisserver` en mode standalone :

```
% waisserver -p 210 -d /usr/local/sfbases
```

La construction d'un formulaire, simple mais opérationnel, a été réalisée à partir des lignes de code suivantes (seul le code "spécifique" à SFgate est repris ici ; les bases nécessaires à l'élaboration d'un formulaire HTML sont supposées acquises) :

Exécution du script CGI SFgate

```
<FORM METHOD=POST ACTION="/cgi-bin/SFgate-5.111/SFgate">
```

- Cet événement survient suite à la validation du formulaire par un bouton de type "`submit`". La méthode utilisée est "`POST`", elle sera préférée à la méthode "`GET`" afin de ne pas avoir d'ennui avec la longueur de la chaîne des arguments qu'on lui passe.

Emplacement de la ou des bases à interroger

```
     Option 1
<INPUT NAME="database" TYPE="hidden" VALUE="xxx.xxx.xxx.xxx/sfgate/bdmail">
      Option 2
<INPUT NAME="database" TYPE="checkbox" VALUE=" xxx.xxx.xxx.xxx /sfgate/bdmail1">
<INPUT NAME="database" TYPE="checkbox" VALUE=" xxx.xxx.xxx.xxx /sfgate/bdmail2">
etc.
```

- L'option 1 utilise le paramètre "`hidden`" pour l'attribut `TYPE`, qui comme son nom l'indique, pointe SFgate sur la base désirée sans qu'il y ait d'affichage sur la page HTML.
- L'option 2 propose un choix de bases à interroger (plusieurs bases peuvent être interrogées simultanément), à l'aide de cases à cocher.

Interrogation par les opérateurs booléens

```
     Option 1
<INPUT TYPE="hidden" NAME="tie" VALUE="and">
     Option 2
<INPUT TYPE="hidden" NAME="tieinternal" VALUE="and">
     Option 3
<INPUT TYPE="radio" NAME="tie" CHECKED VALUE="and">
<INPUT TYPE="radio" NAME="tie" CHECKED VALUE="or">
```

- Les opérateurs "`and`" et "`or`" sont supportés. D'une façon générale, le paramètre "`tie`" pour l'attribut `NAME` signifie que les opérateurs booléens seront appliqués entre les champs. Avec le paramètre "`tieinternal`", ils opéreront entre les mots d'un champ.
- Comme pour l'option 1 du cas précédent, le choix n'est pas demandé à l'utilisateur dans le cas des options 1 et 2.

- L'option 3, par exemple, propose un choix entre "and" et "or" pour lier les champs, le "and" étant le paramètre par défaut.

Mise en forme des résultats suite à une requête

```
    Option 1
<INPUT TYPE="hidden" NAME="language" VALUE="french">
    Option 2
<INPUT TYPE="hidden" NAME="application" VALUE="nom1">
    Option 3
<INPUT TYPE="hidden" NAME="multiple" VALUE="1">
    Option 4
<INPUT TYPE=TEXT NAME="maxhits" VALUE="200" SIZE=4>
    Option 5
<INPUT TYPE="hidden" NAME="range" VALUE="1">
```

- L'option 1 précise la langue dans laquelle les résultats sont présentés. Si l'on utilise le français, les opérateurs booléens seront "et", "ou" à la place de "and", "or".
- L'option 2 permet d'ajouter à la page de résultats un en-tête et un pied de page. Ils seront respectivement définis dans les fichiers nom1_header et nom1_footer. Ces fichiers sont conservés dans le répertoire spécifié lors de l'installation de SFgate (après avoir lancé le premier "make") : /usr/local/apache/cgi-bin/Sfgate-5.111/applifiles/.
- Comme pour FreeWAIS-sf, l'ensemble des documents correspondant aux critères de recherche sont présentés sur une ou plusieurs pages HTML, à raison d'une ligne par document. Dans l'option 3, le paramètre "multiple" de l'attribut NAME place une case à cocher en face de chaque ligne pour permettre de rapatrier plusieurs documents primaires à la fois.
- L'option 4 permet à l'utilisateur de spécifier le nombre maximum de documents pouvant être retournés suite à une interrogation. La valeur par défaut est fixée à 200.
- L'option 5 place en fin de page un lien vers la suite des résultats, lorsque ceux-ci ne tiennent pas sur une seule page.

De nombreuses autres possibilités de SFgate restent à explorer, notamment :

- Les routines de conversion : ce sont des sous-routines Perl qui permettent de modifier la présentation des documents primaires rapatriés suite à une requête. En format standard, SFgate écrit le titre du document entre les balises <H2> et </H2>, le reste étant placé entre <PRE> et </PRE>. Une routine doit être nommée nom_de_routine.pm et placée dans le répertoire des bibliothèques Perl spécifié à l'installation : /usr/lib/perl5/site_perl/.

- Interrogation multi-bases hétérogènes. Il a été vu plus haut comment proposer une interrogation sur plusieurs bases à la fois, par un système de cases à cocher. Ceci ne concernait cependant que des bases présentant les mêmes champs. Notre application d'archivage de messages issus des listes de diffusions et du forum s'en satisfait d'ailleurs très bien.
Dans le cas où certains champs différeraient, il faut recourir au fichier lattice du répertoire d'applications /usr/local/apache/cgi-bin/Sfgate-5.111/applifiles/. Ce fichier est en fait un modèle ("template") qui décrit d'une façon générique et unique les champs de chaque base. Pour chaque base, on crée ensuite un fichier (que l'on place également dans le répertoire applifiles/) faisant la correspondance entre le champ générique, et le champ spécifique à la base.
Ce système permet par exemple d'interroger, à l'aide d'un seul champ, trois champs "année" dénommés différemment selon la base, l'un s'appelant "AN", l'autre "année" et le dernier "year".

4. Installation et configuration logicielles

a. *Présentation de FreeWAIS-sf et choix d'une version*

Cette distribution contient des programmes client et serveur, ainsi que des programmes auxiliaires pour le protocole TCP/IP connu sous le nom de WAIS (Wide Area Information Services). Un système WAIS est constitué de clients qui dialoguent avec le programme serveur `waisserver`, via un réseau TCP/IP utilisant le protocole WAIS. Le serveur répond à une requête en utilisant des index créés à partir du document original par le programme `waisindex`. Une requête pourra par exemple être adressée par l'intermédiaire du programme client `waissearch`.

Un juste milieu doit être observé pour le choix d'une version : la toute dernière évolution d'un logiciel libre est plus susceptible de présenter des bogues encore inconnus, contrairement à une version ancienne. En revanche, cette dernière risque de ne pas posséder certaines améliorations majeures. Notre cas était plus délicat puisqu'à ces considérations se superposent des problèmes de compatibilité avec Linux, produit en pleine expansion mais encore très récent. D'autant plus que l'application entière comporte quatre modules `freeWAIS-sf`, `Perl`, `Wais.pm` et `SFgate`.

Une solution semblait avoir été trouvée, grâce à la cellule MathDoc de l'Université Joseph Fourier de Grenoble, qui proposait des versions de `wais.pm` et `SFgate` spécialement arrangées pour Linux (ftp://mathdoc.ujf-grenoble.fr/pub/mathdoc/SFgate/) J'avais donc choisi une configuration logicielle similaire à celle de Grenoble qui tournait sous Linux :

- `FreeWAIS-sf-2.1.2`
- `Perl 5.004_04` (installé en standard avec la Red Hat 5.2)
- `Wais-2.307-patched` (module `wais.pm`)
- `SFgate-5.111-patched`

Il s'est avéré que la compilation du module `wais.pm` posait de sérieuses difficultés. J'ai utilisé par la suite les versions proposées par l'informaticien avec qui j'ai établi un contact[14] pour résoudre ce problème, `FreeWAIS-sf-2.2.1` et `Wais-2.311`.

b. *Installation et Configuration de FreeWAIS-sf 2.2.1*

➲ La première étape consiste à dépaqueter la distribution dans le répertoire `/usr/local` :

```
% gunzip -cd freeWAIS-sf-2.2.1.tar.gz | tar xvf -
```

La distribution contient les répertoires suivants (dans `/usr/local/freeWAIS-sf-2.2.1/`) :

- **`FIELD-EXAMPLE`** : données et procédures qui vérifient le système une fois que freeWAIS-sf est construit.
- **`bin`** : scripts shell servant pour la plupart à afficher les documents non textuels suite à une requête.
- **`ctype`** : code pour les jeux de caractères 8 bits.
- **`doc`**: documentation qui accompagne freeWAIS (répertoires `CNIDR` et `original-TM-wais`) et freeWAIS-sf (répertoire `SF`).
- **`ir`** : code principal pour la plupart des indexeurs et pour le serveur.
- **`lib`** : quelques fonctions C qui auraient pu être oubliées sur certains systèmes.
- **`regexp`** : code pour la gestion des expressions régulières.
- **`ui`** : quelques clients standards.
- **`x`** : le client standard X11.

[14] Voir page 50.

➲ Création des Makefiles grâce au script `configure` :

`% ./configure`

Suite à l'exécution de ce script, plusieurs questions sont posées à l'administrateur, relatives à différentes propriétés du système, au volume de données à traiter, etc. Les paramètres par défaut ont été acceptés. Il est conseillé en particulier d'accepter le paramètre proposé pour la dernière question ('Use your ctype ?') portant sur le jeu de caractères à utiliser, ce qui permet de rentrer tous les caractères accentués manuellement, et de ne pas avoir de problèmes par la suite.

➲ Création de l'exécutable par la commande make :

`% make` (construit le système et exécute certains tests)

➲ Lancement des tests

`% cd FIELD-EXAMPLE`
`% ../ir/waisserver -d . -p 9565`
`% make test`
`% kill waisserver_pid` (`waisserver_pid` est le numéro du processus `waisserver` lancé)

Des requêtes diverses sont posées grâce au script `test`, et les résultats sont comparés aux résultats attendus contenus dans la distribution. Certains tests peuvent échouer selon les paramètres rentrés suite à l'exécution du script `configure`.

Liste des tests réussis :

Test	Requête posée
PLAIN	Probabilistic Indexing
BOOLEAN	au=(pennekamp or fuhr) and processing
FIELD	au=pfeifer
NUMERIC	py==1995
COMPLEX	py==1995 and (ti=(Retrieval freeWAIS) or au=pfeifer)
PARTIAL	Pfeif*
DATE	Ed>930101
LITERAL	'Enhanced Retrieval'

Liste des tests ayant échoué :

PROXIMITY	Durant la configuration, il a fallu choisir entre les opérateurs de proximité et les opérateurs booléens. Ces derniers ont été préférés, le test a donc normalement échoué.
GERMAN	Ce test ne réussit que lorsqu'on déclare le caractère "β" comme un caractère légal (ce qui est le cas pour la langue allemande).

➲ Puisque la phase de tests est concluante, l'on peut procéder à l'installation du système :

`% make install` (installe les scripts et les binaires)
`% make install.man` (installe les pages du manuel)
`% make clean` (supprime les fichiers temporairement copiés sur le disque pour l'installation)

Ci-dessous, un extrait de l'arborescence installée, reprenant certains répertoires et fichiers importants :

1er niveau	2nd niveau	3e niveau
/usr/local/	bin/	waisindex
		waissearch
		waisserver
	lib/	
	man/	

➲ Ajouter le chemin `/usr/local/bin/` dans la variable PATH afin de pouvoir lancer `waisindex`, `waissearch` et `waisserver` depuis n'importe quel répertoire.

c. *Présentation de SFgate 5.111*

SFgate est une passerelle entre le World Wide Web et WAIS. Ce module est particulièrement bien adapté à FreeWais-sf puisqu'il supporte toutes ses extensions ; en outre, tous les serveurs comprenant le protocole WAIS peuvent être contactés.

SFgate est en fait un script CGI, et en temps que tel, sera utilisable par tout serveur Web habilité à exécuter ce type de scripts. Un formulaire HTML constitue l'interface utilisateur, et remplace la ligne de commande formant l'équation de recherche. L'interrogation via SFgate peut porter simultanément sur autant de bases WAIS que le formulaire en propose (par exemple, une liste de bases sera placée en début de page, et l'utilisateur aura comme choix d'interroger toutes les bases en même temps, ou seulement celles cochées). L'ensemble des documents répondant à une requête est regroupé dans une page HTML sous la forme d'une ligne (entièrement paramétrable) par document. L'accès au document primaire s'effectue ensuite par double-click sur la ligne désirée. Les résultats peuvent également être directement présentés sous forme de documents primaires.

d. *Mise en place de SFgate 5.111*

➲ Cette mise en place nécessite en premier lieu l'installation préalable du module Perl `Wais.pm`, qui fournit un accès aux bibliothèques FreeWAIS-sf.

Ce module, indispensable au fonctionnement de SFgate, a posé de sérieux problèmes d'installation. Des erreurs de compilation demandaient en effet de se plonger au cœur du programme pour adapter le module à ma plate-forme. SFgate constituait l'aboutissement de l'application la plus importante devant être installée sur le serveur : un outil de recherche documentaire souple, puissant, et libre. Les forums de discussions sur Perl, ainsi que la liste de diffusion à laquelle j'étais abonné ont apportés plusieurs solutions ; malheureusement, aucune d'entre elles n'a résolu le problème. L'envoi de courriers électroniques aux concepteurs de SFgate, ou à certaines personnes ayant donné des cours d'installation de ce module (pas sur Linux, malheureusement), s'est aussi soldé par un échec. La spécificité de mon problème (rareté, pour l'instant, des plates-formes Linux tournant avec ce type d'application) explique la difficulté à trouver de l'aide. J'ai aussi contacté les différents services informatiques de l'INRA, qui disposent d'un réseau de compétences étendu. Il s'est avéré qu'une seule personne pouvait s'occuper de mon cas, l'Ingénieur d'Etudes Christophe Caron, du centre de Jouy-en-Josas. Il lui a donc été ouvert un compte disposant des droits du super-utilisateur. Le problème a finalement pu être résolu : en annexe se trouvent les opérations effectuées.

Le module `wais.pm` ayant été installé en fin de stage, SFgate a pu alors être mis en place et configuré (sommairement, par manque de temps, à l'heure où est rédigé ce rapport) :

➲ Désarchivage et décompactage de la distribution SFgate dans le répertoire `Sfgate-5.111/` :

```
gunzip -c SFgate-5.111.tar.gz | tar xvf -
```

➲ Editer le 'Makefile' et paramétrer la variable PERL avec le chemin correct menant au Perl de la plate-forme, dans notre cas : `/usr/bin/perl`.

➲ Lancer la configuration de SFgate en tapant :

```
% make
```

Des propriétés du système sont déterminées, puis certaines questions relatives par exemple à l'emplacement de différents répertoires et fichiers sont posées :

Where should the Perl libraries go ?	`/usr/lib/perl5/site_perl`
Where should the Perl utilities go ?	`/usr/bin`

What is the default path to the local wais database ?	/usr/local/sfbases
What is your http proxy ?	None
Where do your html pages resides ?	/usr/www/fsite/htdocs/
Where should the documentation go ?	Sfgate
What is the virtual name of the document directory ?	/sfgate
Where is your real CGI dir ?	/usr/local/apache/cgi-bin/SFgate-5.111
What is the virtual name of your cgi directory ?	/cgi-bin/sfgate
In which directory should the application files go ?	/usr/local/apache/cgi-bin/SFgate-5.111/applifiles
In which directory should the logfile SFgate.log go ?	/usr/local/apache/cgi-bin/SFgate-5.111/log
Send registration mail to sfgate@ls6.informatik.uni-dortmund.de ?	Y[15]

Après avoir répondu aux questions, un fichier `config.sh` est créé. Il est alors possible de le modifier en lançant la commande :

```
% make configure
```

Si l'on désire juste revoir les paramètres que l'on a rentré, lancer la commande :

```
% make show
```

➲ Lorsque `config.sh` est paramétré correctement, il reste à créer l'exécutable `SFgate` en lançant la commande :

```
% make
```

Quelques tests sont réalisés en se connectant sur un serveur FreeWAIS-sf à l'adresse : http://ls6-www.informatik.uni-dortmund.de/. L'échec de ces tests n'est pas forcément rédhibitoire (de nombreuses raisons peuvent exister : changement de place de la base de données, indisponibilité du serveur,…). Dans notre cas, ces tests n'ayant pas été concluants, il a été procédé à une deuxième série d'essais, sur la base `bdmail` nouvellement créée. La distribution SFgate fournit dans le répertoire `SFgate-5.111/doc/` un formulaire d'exemple qu'il suffit d'adapter, `demo.html`, lequel reprend un grand nombre des fonctionnalités de SFgate. Sont reportées ici les lignes modifiées pour le besoin du test, qui est extrêmement simple : il ne porte que sur l'interrogation du champ "catégorie" de la base `bdmail`, avec comme paramètre par défaut "linux" :

```
<FORM METHOD="POST" ACTION="/cgi-bin/SFgate-5.111/SFgate">
      <DL>
        <DT> <INPUT NAME="database"
            TYPE="checkbox"
            VALUE="xxx.xxx.xxx.xxx/sfgate/bdmail" CHECKED>
          <B>DEMO</B>
        <DD> Base de test <BR>
      </DL>
<HR><H2>Entrez votre requête</H2><HR>
      <DL>
        <DT> Categorie
        <DD> <INPUT TYPE="text"
            NAME="ca"
            VALUE="linux">
      </DL><HR>
```

Le succès de cet essai a autorisé l'installation complète de SFgate (installation des fichiers aux emplacements spécifiés dans `config.sh`) par la commande :

```
% make install
```

Restent à vérifier les permissions, en particulier celles de l'exécutable `SFgate`, et à rajouter une ligne au fichier de configuration d'Apache, `httpd.conf`. Il est en effet conseillé de ne pas mettre les bases de données dans l'arborescence de documents placée sous `htdocs`. Chaque base WAIS créée est

[15] Une réponse positive provoque l'envoi automatique d'un courrier électronique aux concepteurs de SFgate. Cet enregistrement n'est réalisé qu'à titre informatif.

placée sous `/usr/local/sfbases/`, dans un sous-répertoire spécifique (`listdiff/` pour la base `bdmail`). De la même façon que l'on a créé un alias pour l'emplacement des scripts CGI (afin que celui-ci reste transparent aux yeux des utilisateurs), on rajoute un alias pour le chemin d'accès aux bases de données :

```
Alias /sfgate/ "/usr/local/sfbases/"
```

Le succès très tardif de l'installation du module SFgate n'a pas laissé le temps d'étudier de façon approfondie sa configuration. L'exploitation des nombreuses possibilités du module sera abordé suite au stage.

B. Installation d'un moteur de recherche : Xavatoria

☞ http://www.xav.com/scripts/xavatoria/index.html

1. Présentation

Xavatoria est un moteur de recherche qui autorise une recherche booléenne complexe avec troncature à gauche et à droite. Le paramétrage permet une restriction sur les pages à rechercher, et plusieurs paramètres de pertinence sont modifiables. Les metadata tels que *keywords* et *description* sont prises en compte. Enfin, ce moteur peut gérer efficacement une masse de donnée allant jusqu'à une vingtaine de mégaoctets.

2. Potentialiser la recherche d'informations déjà présentes sur le site

Un moteur de recherche était initialement prévu pour effectuer indépendamment deux types de recherches, sur le site entier et sur l'Index Synonymique de la Flore de France de Michel Kerguélen[16]. Xavatoria, application Perl libre, comprenait les facilités d'interrogation désirées et offrait une vitesse d'exécution parfaitement adaptée à la masse des données à traiter.

Cependant, les difficultés rencontrées initialement avec l'interface Web de FreeWAIS-sf ont amené à étendre le domaine de recherche de Xavatoria aux notices bibliographiques issues de Texto. Seule contrainte, le résultat d'une requête était présenté sous forme de pages HTML : il fallait donc morceler le fichier texte de notices en autant de pages HTML que de notices. La recherche puis l'apprentissage du langage le plus approprié à ce type de tâche ont été entrepris. L'objectif fixé a finalement été atteint à l'aide de scripts écrits en Perl.

a. *Notes préliminaires : le langage PERL*[17]

Présentation et explication du choix de ce langage pour la réalisation et l'exécution de scripts CGI

Perl est un langage interprété (avec une phase interne de précompilation) optimisé pour traiter des fichiers textes, mais qui peut également être utilisé pour diverses tâches d'administration-système. Sa syntaxe s'inspire très largement de celles de C, sed, awk et sh. On le trouve librement sur Internet. Son utilisation touche de nombreux domaines : traitement de fichiers texte, extraction d'informations, écriture de scripts d'administration système, prototypage rapide d'applications, etc. Il permet aussi l'écriture d'applications

[16] Cet index est une liste alphabétique des taxons de la flore spontanée et cultivée française, leurs synonymes et leurs hybrides. Il comporte environ 62 000 citations de taxons.
[17] Practical Extraction and Report Language

puissantes qui peuvent tourner immédiatement sur plusieurs plates-formes différentes. En outre, les nombreuses bibliothèques[18], ainsi qu'une grande quantité de modules déjà disponibles permet de développer rapidement des applications touchant à des domaines divers (CGI, Tk, Gtk, Msql, POSIX, Curses, NNTP, etc.).
Perl apparaît donc parfaitement adapté aux tâches à réaliser : création de pages HTML dynamiques, extraction et reformatage de données etc. D'autant plus que le module Perl est fourni en standard dans la distribution Linux Redhat 5.2 (l'interpréteur Perl est installé dans le répertoire `/usr/bin`).

Le réseau CPAN (Comprehensive Perl Archive Network) a été mis en place pour centraliser tous les documents et fichiers relatifs à Perl :

ftp.funet.fi (site principal)
ftp://ftp.jussieu.fr/pub/perl/CPAN/ (sites miroir en France)
ftp://ftp.lip6.fr/pub/perl/CPAN

Deux sites, sources d'informations précieuses, le premier étant plus technique, et le second maintenu par un groupe d'utilisateurs :

http://www.perl.com/
http://www.perl.org/

Enfin, les groupes de news et les listes de discussion (en français) :

fr.comp.lang.perl
perl-fr@oghma.com
perl@u-strasbg.fr

Les expressions régulières[19]

Les expressions régulières sont une des caractéristiques qui font de Perl un langage parfaitement adapté au traitement de fichiers texte[20]. Ce sont des suites de caractères disposés selon une syntaxe spécifique, afin de décrire le contenu d'une chaîne de caractères. Ceci permet de tester si cette chaîne correspond à un motif, pour en extraire des informations ou encore y effectuer des substitutions. Ci-dessous sont présentés à titre d'exemple, la signification de quelques caractères spéciaux (appelés métacaractères) puis de quelques expressions régulières simples.

`?`	Indique que l'expression précédente doit être présente une fois.
`*`	Indique que l'expression précédente peut être présente zéro fois ou plus.
`+`	Indique que l'expression précédente doit être présente une fois ou plus.
`[]`	Classe de caractères.
`.`	N'importe quel caractère mis à part le retour chariot.
`^`	Début de ligne.
`$`	Fin de ligne.
`\`	Change un métacaractère en caractère normal. Le motif "`\.`" Correspond par exemple à un point normal.

`[abc]`	Lettres "a", "b" ou "c" autorisées.
`[a-z]`	Toutes les lettres minuscules de "a" à "z". Ce motif ne correspond qu'à une lettre.
`[a-z]+`	Toute suite de lettres.
`ca[s]+e\.$`	Correspond à toute ligne terminée par "`case.`" ou "`casse.`".

[18] Les deux bibliothèques Perl les plus connues sont celle de l'Anglais Steve Brenner, `cgi-pl.pl` qui fonctionnait au temps de la version 4 du langage PERL et qui continue d'ailleurs de très bien tourner sous PERL5 ; et celle de l'Américain Lincoln D. Stein, `CGI.pm`, qui fonctionne à partir de PERL5.003 et qui est beaucoup plus puissante et élégante.
[19] Sont également appelées expressions rationnelles.
[20] Le système d'expressions régulières de Perl provient directement du monde Unix, plus précisément de Sed et Awk.

Remarque - Premiers pas en Perl

L'apprentissage du langage de script Perl, apparaît beaucoup plus ésotérique que JavaScript ou Visual Basic (seules connaissances en programmation, avec HTML). Ce langage étant en pleine croissance, il a été heureusement aisé de trouver différents tutoriels (en français ou en anglais) ; en outre, les ouvrages en texte intégral et accessibles gratuitement sur Internet (http://itlibrary.com) ont constitué une aide particulièrement précieuse. Il est à noter que les listes de diffusion françaises Perl sont peu actives, au contraire du forum français, qui a fourni rapidement des solutions aux problèmes rencontrés.

Remarque sur les scripts reportés dans ce mémoire

La réalisation de certaines tâches très spécifiques n'autorisait pas l'utilisation de scripts prêts à l'emploi, et a nécessité l'apprentissage du langage Perl. Les seuls scripts reportés dans ce mémoire seront ceux que j'ai écrits, les autres provenant principalement de sources libres sur Internet. Dans ces scripts, des sections de configuration commentées permettent d'adapter chaque programme à son utilisation propre. Cependant, la personnalisation de ces programmes peut demander de se plonger dans le cœur du script, la compréhension (au moins partielle) du langage dans lequel il est rédigé devenant alors indispensable. C'est pour cette raison que je n'ai cherché sur Internet que des scripts CGI écrits en Perl (d'ailleurs numériquement très largement dominants dans les banques de scripts explorées).

Le transfert de ces scripts Perl doit s'effectuer exclusivement en mode ASCII, sous peine de déclencher des erreurs d'exécution ("Internal Server Error"). En mode binaire, le transfert peut endommager des caractères cachés de fin de ligne et rendre le script non-exécutable, même si le fichier semble extérieurement n'avoir subi aucune modification.

b. *Traitement des notices Texto*

- La base bibliographique des ouvrages du centre est élaborée sous Texto. La sortie de l'ensemble de la base fournit un fichier texte de notices bibliographiques correspondant au format suivant (le nombre de champs n'est pas fixe) :

```
REF     .00048
CA      .Groupement pour l'Avancement des Méthode Spectrographiques GAMS
CG      .Journées internationales d'études des méthodes de séparation
        .immédiate et de chromatographie;Paris (FRA);1961/06/13-15
DA      .1961
PG      .350 p.
LA      .eng;fre;ger
NT      .absent
ED      .Groupement pour l'Avancement des Méthodes
        .Spectrographiques;PARIS;
        .(FRA)
LO      .5J GAM 54;DOCDJ
MC      .CHROMATOGRAPHIE
```

- Le résultat d'une requête est constitué d'un ensemble de pages HTML, chacune correspondant à une notice. J'ai donc écrit un script Perl qui extrait chaque notice d'un fichier texte issu de Texto, construit une page HTML par notice, et nomme cette page à partir de sa cote (la référence `00048` ci-dessus, de cote `5J GAM 54;DOCDJ`, donne le fichier `5JGAM54.html`). L'utilité de cette notation est développée par la suite.

```
#!/usr/bin/perl -w

# Vérifier que la première ligne du fichier à formater correspond
# à la première ligne de la première notice
```

```
sub Date {
 local(@DATE)=("Janvier","Février","Mars","Avril","Mai","Juin","Juillet",
"Août","Septembre","Octobre","Novembre","Décembre");
 ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst)=localtime(time);
 $date="$mday $DATE[$mon] 19$year  --  $hour:$min:$sec";
 }

print "\nNom du fichier Texto à formater (sans l'extension) ?  ";
$nom=<STDIN>;
chop $nom;
$nomtxt = $nom.".txt";
$cpt=-1;    # compteur nb lignes fichier
$cptnot=-1;  # compteur nb lignes notice
$nbnotices=0;
$date = &Date ();

open(TEXTO,"<$nomtxt") or die "Ce fichier n'existe pas !\n\n";
@tab=<TEXTO>;
print "\n";

foreach (@tab) {
 ++$cpt;
 ++$cptnot;
 if ($_ =~ /LO /) {
       # sans espace après /LO, sélectionne aussi MICROBIOLOGIE
       $save = $_;

       # exemple de cote : LO     .SL LAS 1161;DOCDJ
       s/;.+$//;   # supprime tout à partir de la première parenthèse
rencontrée
       s/LO *\.//;
       s/ //g;
       chop $_;
       $singlecote = $_.".html";
       $cote ="/usr/www/fsite/htdocs/notices/$_.html";
       $_ = $save;
       }

 if ($_ !~ /[A-Za-z]/) {
       ++$nbnotices;
       open(NOTI,">$cote");
       print         NOTI         "<html><head><title>Notices</title><body
background=images/bg524.jpg>\n";
       print      NOTI      "<font      size=+3      color=#AA0000>Notice
demandée</font><hr><br>\n";
       print NOTI"<blockquote>\n";
       for ($i=($cpt-$cptnot); $i<$cpt; ++$i) {
             print NOTI "$tab[$i]<br>\n";
             }

       $sommaire="/usr/www/fsite/htdocs/sommaires/".$singlecote;
# vérifie si un sommaire correspondant à la notice existe,
# et dans ce cas crée en bas de la page HTML un hyperlien vers ce
sommaire.
```

```
        if (!open(SOMM,"<$sommaire")) {
```

```
        }
```

```
        else {
```

```
            print      NOTI      "<br><br><blockquote>Consulter      <a
href=/sommaires/$singlecote>le sommaire</a><br>\n";
            }

       print NOTI "<br><h5 align=center><a href=/index.html>Retour page
principale</a>\n";


       print NOTI "</blockquote></body></html>\n";
             $cptnot=-1;
       close (NOTI);
       print "Fichier $cote créé.\n"
       }
 }
close (TEXTO);
print "\n";
print "------ Opération réalisée le  $date\n";
print "Nombre de notices rentrées :  $nbnotices\n";
open (NBNOTI,"<nbnot");
 @tabnb=<NBNOTI>;
 @deblign = split(/ /,$tabnb[$#tabnb]);
 $nbtot=$deblign[0]+$nbnotices;
 print "Nombre total de notices    :  $nbtot\n";
 print   "-------------------------------------------------------------
\n\n";
close (NBNOTI);

open (NBNOTI,">>nbnot");
 print NBNOTI "$nbtot notices  --  $date\n";
close (NBNOTI);
```

Session d'exemple (les données rentrées par l'utilisateur sont en gras) :

```
./notice
Nom du fichier Texto à formater (sans l'extension) ? FichierTexto
Fichier /usr/www/fsite/htdocs/notices/Notice1.html créée
Fichier /usr/www/fsite/htdocs/notices/Notice2.html créée
Fichier /usr/www/fsite/htdocs/notices/Notice3.html créée
Etc.

"------ Opération réalisée le 28 Juin 1999  -  10:18 ------
Nombre de notices rentrées : 129
Nombre total de notices    : 1852
"-----------------------------------------------------------
```

En outre, chaque opération est consignée dans le fichier texte `nbnot` dont le contenu, après les deux premières opérations, est le suivant :

```
0
2669 notices  --  21 Juillet 1999  --  10:58:30
2755 notices  --  26 Juillet 1999  --  19:6:41
```

3. Mise en ligne de ressources auparavant seulement disponibles sur papier à l'UPE-Doc

Il est apparu utile de compléter la base de notices bibliographiques déjà existante en lui associant une base des sommaires correspondants. L'idée avait le mérite d'être attrayante, mais occasionnait un surcroît de

travail considérable : il fallait, pour chaque monographie, numériser le sommaire, le passer à l'OCR[21], effectuer les corrections requises, et insérer le texte du sommaire dans une page HTML. Une méthodologie a donc été mise au point afin d'automatiser le travail autant qu'il était possible.

a. *Traitement des sommaires*

L'objectif était de proposer, suite à la sélection d'une notice, l'affichage du sommaire correspondant. Réciproquement, suite à une recherche sur les sommaires, il fallait être en mesure de fournir les notices correspondantes.

La majeure partie du travail consiste à préparer une source de données, à partir de sommaires numérisés puis passés à l'OCR, sous forme d'un fichier texte. Le script suivant, voisin du précédent, extrait de ce fichier chaque ligne de sommaire, et la place dans un tableau. Le résultat final est une page HTML.

```
#!/usr/bin/perl -w

sub Date {
 local(@DATE)=("Janvier","Février","Mars","Avril","Mai","Juin","Juillet",
"Août","Septembre","Octobre","Novembre","Décembre");
 ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst)=localtime(time);
 $date="$mday $DATE[$mon] 19$year  --  $hour:$min:$sec";
 }

$nbsommaire=0;
$nbtot=0;

open (NBSOMM,"<nbsom");
 @tabnb=<NBSOMM>;
 @deblign = split(/ /,$tabnb[$#tabnb]);
 $nbtot=$deblign[0]+$nbsommaire;
 print "\n-----------------------------------------------------------------
\n";
 print "Nombre total de sommaires  : $nbtot\n";
 print   "-----------------------------------------------------------------
\n\n";
close (NBSOMM);


do
{

print "\nNom du fichier à formater (sans l'extension) ?  ";
$nom=<STDIN>;
chop $nom;
$nomtxt = $nom.".txt";
$nomhtml = "/usr/www/fsite/htdocs/sommaires/".$nom.".html";
$date = &Date ();

open(OCR,"<$nomtxt") or die "Ce fichier n'existe pas !\n\n";
open(SOMM,">$nomhtml");
 print            SOMM            "<html>\n<title>Sommaire</title>\n<body
background=images/bg524.jpg>\n";
 print       SOMM       "<font       size=+3       color=#AA0000>Sommaire
demandé</font><hr><br><center>\n";
 print SOMM "<table border=2>\n";
       @tab=<OCR>;
       foreach (@tab) {
             s/\t/<td>/g;
             print SOMM "<tr>";
```

[21] Optical Characters Recognition

```
            print SOMM "$_\n\n";
            }
 print    SOMM    "</table></center>\n<br><br><blockquote>Consulter    <a
href=/notices/$nom.html>la notice bibliographique</a></blockquote>\n";
 print    SOMM    "<h5    align=center><a    href=/index.html>Retour    page
principale</a></h5>\n";
 print SOMM "<br><br><br></body>\n</html>\n";
close (SOMM);
close (OCR);

++$nbtot;
open (NBSOMM,">>nbsom");
 print NBSOMM "$nbtot sommaires  --  $date\n";
close (NBSOMM);



print "\n------ Opération réalisée le  $date\n";
print "Fichier $nomhtml créé.\n\n";


print "\nUn autre sommaire (<ENTER> pour oui, <n> pour non)  ";
 $autre = <STDIN>;
 chop $autre;


} while ($autre ne "n");

print "\n-----------------------------------------------------";
print "\nNombre total de sommaires  : $nbtot\n";
print "-----------------------------------------------------\n\n";
```

b. *Liens entre notices et sommaires - Protocole d'exploitation*

➢ *Les notices*

Le transfert des fichiers Texto, de l'ordinateur ayant accès à la base de données jusqu'à la plate-forme Linux, occasionnait des problèmes d'accentuation lorsqu'il était effectué via les disquettes et sous un format texte quelconque. Des caractères spéciaux étaient en effet substitués aux caractères accentués. J'ai remédié à ce problème en utilisant la démarche suivante :

➲ Enregistrer le fichier Texto (format texte MS-DOS) en format "Texte seulement" sous Word, et transférer le fichier par FTP (en mode ASCII) sur le serveur de centre.

➲ Sur la plate-forme Linux, récupérer le fichier par FTP en mode binaire (curieusement, le problème d'accentuation réapparaît si le transfert est effectué en mode ASCII).

Le fichier Texto est placé dans le répertoire `/usr/www/fsite/work/notices/`. Puis le script de formatage est lancé par : `./notice`. L'ensemble des pages HTML créées est alors automatiquement placé dans le sous-répertoire `/notices` du répertoire principal du site (`htdocs`).

Remarque - format de la cote d'une monographie

Les cotes affectées aux monographies ont une structure de la forme "`AGR SEB 2627`" : le thème de l'ouvrage est l'agronomie, le nom de l'auteur est Sébillote, et le numéro de saisie sous Texto est 2627.

➢ *Les sommaires sont numérisés puis passés à l'OCR (OmniPage)*

Les quelques erreurs sont corrigées manuellement, et l'on obtient un fichier texte du type :

```
METHODS IN ENZYMOLOGY VOLUME II
TABLE OF CONTENTS
```

CONTRIBUTORS TO VOLUME II v
OUTLINE OF ORGANIZATION, VOLUME II ix
OUTLINES OF VOLUMES I, III, IV xviii
ERRATA YFOR VOLUME I xx
Section I. Enzymes of Protein Metabolism
1. Swine Pepsin and Pepsinogen ROGER M. HERITIOTT 3
2. Chymotrypsinogens and Chymotrypsins M. LASKOWSKI 8
3. Trypsinogen and Trypsin M. LASKOWSKi 26
4. Naturally Occurring Trypsin Inhibitors M. LASKOWSKi 36
etc.

Le format de ligne reconnu par le script est ainsi : tabulation - texte - tabulation - pagination - saut de ligne. Ce format correspond au format de sortie généralement observé après l'étape de numérisation et d'OCR.

- *Chaque notice doit être reliée à son sommaire et inversement.*

Le fichier texte prêt pour le script sommaire (représentant un seul sommaire) est enregistré sous le nom "`cote.html`", où cote représente la cote de l'ouvrage dont le sommaire vient d'être numérisé. La notice correspondant au sommaire possède le même nom, la liaison est donc aisée : chacun des deux scripts construit en bas de la page HTML un hyperlien renvoyant au même nom de fichier, mais avec une localisation différente : répertoires `/usr/www/fsite/htdocs/notices` ou `/usr/www/fsite/htdocs/sommaires`.
Avant de créer dans la page HTML de notice un hyperlien vers le sommaire correspondant, le script vérifie que ce dernier existe. Dans le cas contraire, aucun hyperlien n'est créé. La vérification n'a lieu que dans ce sens : si la création des pages de notices est presque instantanée, il en va autrement pour la constitution de la banque de sommaires.

4. Installation et configuration de Xavatoria

Deux scripts sont à installer dans le répertoire `/cgi-bin/`, puis à éditer : `build.pl` et `search.pl`.

Le premier construit `index.txt`, le fichier d'index du site (ou de la partie de site) sur lequel s'effectuera la recherche. Le second réalise la recherche à partir de l'équation de recherche et du fichier d'index. Quelques autres fichiers utiles à l'exploitation de Xavatoria seront décrits ensuite.
La configuration reportée ci-dessous est adaptée pour une recherche sur la base de notices.

- *Build.pl*

`$Index_File "/usr/www/fsite/htdocs/notices/index.txt";`	Emplacement du fichier d'index.
`$baseurl = "http://xxx.xxx.xxx.xxx/notices";`	Emplacement (URL) du répertoire principal où se trouve les pages soumises au moteur.
`$basedir = "/usr/www/fsite/htdocs/notices";`	Idem, en chemin absolu.
`$extensions = "\.html\.htm\.";`	Extensions des fichiers sur lesquels sera effectuée la recherche.
`$DMZ .="/usr/www/fsite/htdocs/notices/summaries.html "; $DMZ .="/usr/www/fsite/htdocs/notices/search.html ";`	Répertoires ou fichiers qui doivent être exclus des recherches

Xavatoria classe les documents trouvés par ordre décroissant de pertinence, selon le nombre d'occurrences des termes recherchés et leurs places dans le document. Chaque occurrence d'un terme

compte un point. Un coefficient multiplicateur est appliqué si le terme fait partie de l'URL, du titre de la page HTML, ou des métadata *keywords* ou *description* (respectivement 4, 10, 10, 4).
Le paramétrage ci-dessous représente les valeurs par défaut.

`$Filename_Rank = 4;`	Coefficient pour les termes trouvés dans l'URL.
`$Title_Rank = 10;`	Idem pour le titre.
`$Keyword_Rank = 10;` `$Description_Rank = 4;`	Idem pour les métadata.

Ce dernier paramètre permet de multiplier par le "`$CRANK_FACTOR`" le poids d'un document. Quelle que soit la valeur de "`$CRANK_FACTOR`" pour un document, celui-ci n'apparaîtra bien entendu que s'il correspond à l'équation de recherche.

`$CRANK_FACTOR = 18`	Ce facteur est un entier compris en 2 et 20.
`$CRANK .=` `"/usr/www/fsite/htdocs/index.html ";`	Documents dont le poids est augmenté. Cette ligne n'est donnée qu'à titre d'exemple, il n'est pas utile de donner plus de poids à l'index du site.

- *Search.pl*

```
$Index_File = "/usr/www/fsite/htdocs/notices/index.txt";
```

Emplacement du fichier d'index.

```
$Ignore .= " what how who which when where do you find site get ";
$Ignore .= "and or if not a the for an it of from by the one two to he ";
$Ignore .= "most all about i me search is are be been with why ";
$Ignore .= "quel quels quelle quelles sont est combien qui quoi ";
$Ignore .= "les de des du pour quand où et ou si ne un une le la ";
```

Liste de mots vides à ignorer dans une requête.

```
$Search_Page =
"http://xxx.xxx.xxx.xxx/usr/www/fsite/htdocs/notices/search.html";
```

Emplacement du formulaire de recherche

```
$Hits_Per_Page = 10;
```

Nombre de résultats par page

```
($Link_URL,$Link_Title)     =     ("http://xxx.xxx.xxx.xxx","Retour     page
principale");
```

Définit l'hyperlien en bas de chaque page de résultats.

```
$summary_file = "/usr/www/fsite/htdocs/notices/summaries.html";
```

Emplacement du fichier `summaries.html`

- *Summaries.html*

Ce fichier contient l'historique des requêtes : il permet de se faire une bonne idée de ce qui a été demandé (le demandeur restant anonyme), puisqu'il n'est pas réinitialisé. En voici un échantillon :

Search by :
Raw Query was: 1998 and bacter*

Recherche sur : `1998, bacter*`.
Visualisation des documents 1-2 sur 2, par ordre de pertinence.

Search by :
Raw Query was: las*

Recherche sur : `las*`.
Visualisation des documents 1-10 sur 14, par ordre de pertinence.

Un script Perl pourra par la suite être créé afin d'exploiter et présenter ces données dans un format aisément consultable.

5. Performances et limites

➢ *Recherche complexe*

Les équations de recherche sous Xavatoria sont voisines de celles de grands moteurs comme Alta Vista. Ci-dessous sont reportées la plupart des possibilités offertes.

+CGI +scripting

Trouve "CGI Scripting", "CGI scripting", et "scripting of type CGI"
Ne trouve pas "cgi scripting", ni "CGI script"

-CGI +scripts

Trouve "Perl scripts", "Hollywood scripts", et "cgi scripts"
Ne trouve pas "CGI definitions", ni "CGI scripts"

or CGI and scripts

Trouve "Perl scripts", "CGI scripts", et "CGI Scripts"
Les documents avec les deux termes apparaissent en premier dans la liste.

"CGI scripts"

Trouve "CGI scripts"
Ne trouve pas "CGI Scripts", ni "scripts of type CGI"

and cgi and script*

Trouve "CGI scripts", "cgi scripting", et "Cgi scripter"

not planet and Venus and pictures or images

Liste d'abord "pictures and images of Venus", puis "Venus pictures"
Ne liste pas "venus picture", ni "picture of planet Venus"

Where is the *frog*?

Trouve "frog", "frogleg", et "bullfrog"

Note sur la casse

Une recherche sur "usa" pourra donner "Usa", "USA", et "usA", alors qu'une recherche sur "USA" ne donnera que "USA"

Note sur la troncature

La troncature peut s'utiliser à gauche et à droite mais pas centralement ("po*ier" ne marchera pas). Elle ne peut représenter au maximum que 4 caractères, et ne peut être employée qu'avec des mots d'au minimum trois caractères ("le*" sera ignoré).

➢ *Limites*

Xavatoria gère efficacement la recherche sur un site dont l'indexation des pages fournit un fichier de l'ordre de 15 à 20 mégaoctets. Pour un site de taille nettement supérieure, le ralentissement des performances peut nécessiter l'utilisation d'un moteur plus puissant.

Un test a été réalisé une fois le site de Dijon transféré sur la plate-forme Linux, pour vérifier que le moteur était adapté à la taille du site.

(1) Indexation

Section indexée	Nombre de fichiers indexés	Nombre de mégaoctets indexés	Durée de l'opération
Site entier	3037	10,355	1 min 27 s
Section notices	2589	2.051	12 s

(2) Recherche

Section sur laquelle s'effectue la recherche	Terme recherché	Nombre de pages sélectionnées	Durée de l'opération

Site entier	poisson*	22	5 s
Section notices	poisson*	16	1 s

D'après ces essais, Xavatoria apparaît plus que suffisant pour ce site, lequel pourra encore être enrichi de nombreuses pages (en particulier celles de sommaires) sans voir chuter ses performances de façon préjudiciable.

VII. L'avenir du site : assurer une dynamique de développement

A. En maintenant une qualité de services par une gestion adéquate du site : WebTester et Authorization Gateway

Un dysfonctionnement prend de plus en plus d'ampleur sur le Web : la présence d'une quantité croissante de liens morts au sein de sites mal gérés. Je sais par expérience qu'il est très frustrant de tomber sur la fatidique "Error 404" ; une mesure élémentaire en matière de qualité de services consiste à éradiquer ce type de désagrément. Ce qui est facile pour un petit site comportant peu de pages, l'est beaucoup moins pour un site d'importance, en particulier lorsque ce site regroupe les pages de plusieurs services. J'ai par conséquent recherché un outil susceptible d'automatiser cette tâche. L'application WebTester s'est révélée répondre pleinement aux attentes.

1. Liens et cartographie du site : WebTester 1.05

Http://awsd.com/scripts/webtester/index.shtml

a. *Présentation*

WebTester offre des rapports concernant la cartographie du site et la validité de ses liens (liens internes et externes, des hyperliens classiques à ceux utilisant les images maps ou résultant d'une page générée par un script CGI, etc.).

➢ Informations relatives à la cartographie générale du site

- Documents locaux - Pour chaque document du serveur (page HTML, script CGI) sont listés lorsqu'ils existent, les documents locaux ayant un lien avec ce fichier, les liens valides et les liens locaux "morts" trouvés dans ce fichier.
- Téléchargements - Pour chaque fichier : taille (en octets) et durée théorique de téléchargement selon différentes liaisons (14.4 Kbps, 28.8 Kbps, ISDN, T-1).
- Répertoires.
- Binaires (images, graphiques, fichiers .zip etc.) : liste de ce type de fichiers, et des pages incluant un lien vers ces fichiers.
- Fichiers proposés en téléchargement : même principe.
- Mailto : même principe.
- FTP : même principe.
- Telnet : même principe.
- Gopher : même principe.
- News : même principe.
- URL externes - Liste de tous les liens externes, avec les pages contenant ces liens.

➢ Liens "morts" et autres problèmes

- Fichiers non trouvés - Avec la liste des documents incluant des liens vers ces fichiers.
- Fichiers non lisibles par l'ensemble de la communauté Internet.
- Fichiers trouvés mais non référencés (inutilisés dans les pages du site).
- Répertoires non trouvés - Avec la liste des documents incluant des liens vers ces répertoires.
- Liens dont l'ancre n'existe pas - Avec la liste des documents incluant ces liens.
- Liens externes "morts" - Liste des documents incluant ces liens, statut de ces documents : 301 (Moved Permanently), 302 (Moved Temporarily), 404 (Not Found) etc.).

Remarque - Tous les fichiers et liens listés le sont en hyperliens, ce qui permet un accès direct au document souhaité. La totalité de ces rapports est réunie dans une page HTML unique, `sitecheck.html`, située dans le répertoire `/usr/www/fsite/htdocs/sitetools/`.

➢ Cartographie du site

Ci-dessous est reporté un extrait de la cartographie du site réalisée le 3 août 1999 (l'ensemble de la cartographie se trouve dans la page HTML `sitemap.html`, située dans le répertoire `/usr/www/fsite/htdocs/sitetools/`).

……

- Unites de Recherches - INRA de Dijon - 2 Jul 1999
  - STATION D'AGRONOMIE - 27 Jul 1999
    - Thèses de la Station d'Agronomie - 27 Jul 1999
      - CORPS STATION D'AGRONOMIE - 27 Jul 1999
      - COLBACH RESUME FRANCAIS - 27 Jul 1999
      - COLBACH RESUME ANGLAIS - 27 Jul 1999
      - MEUNIER RESUME FRANCAIS - 27 Jul 1999
      - MEUNIER RESUME ANGLAIS - 27 Jul 1999
  - LABORATOIRE D'AMÉLIORATION DES PLANTES - 27 Jul 1999
    - Thèses de l'Unité de Génétique et d'Amélioration des Plantes - 27 Jul 1999
      - AMELIORATION DES PLANTES DIJON - 27 Jul 1999
      - PAGE RESUME FRANCAIS - 27 Jul 1999
      - PAGE RESUME ANGLAIS - 27 Jul 1999
  - LABORATOIRE DE RECHERCHES SUR LES AROMES de DIJON - 2 Jul 1999
  - Plate-forme de Prédéveloppement en Biotechnologie - 27 Jul 1999
    - Accueil - 27 Jul 1999
    - Communication - 27 Jul 1999
    - Logiciels - 27 Jul 1999
    - Moyens - 27 Jul 1999
    - Organisation - 27 Jul 1999

……

Remarque - fichiers nouveaux ou récemment modifiés

Ces fichiers (voir la section paramétrage) sont signalés par le terme "NEW!", comme dans l'exemple suivant :

- Stats du serveur Web de Serveur Apache 1.3.6 / INRA Dijon - 2 Aug 1999 - **NEW!**

b. *Installation et configuration*

L'application est installée dans le répertoire: `/usr/local/apache/cgi-bin/webtester_files`

```
#!/usr/bin/perl
```

Chemin absolu du compilateur Perl.

```
$InFile = "/usr/www/fsite/htdocs/index.html";
```

Chemin absolu de la page principale du site.

```
$OutFile = "/usr/www/fsite/htdocs/sitetools/sitecheck.html";
```

Nom et localisation de la page HTML contenant les rapports décrits ci-dessus ("Informations relatives à la cartographie générale du site", et " Liens morts et autres problèmes").

```
$MapFile = "/usr/www/fsite/htdocs/sitetools/sitemap.html";
```

Même chose pour la page HTML contenant la cartographie du site.

```
$LocalPath = "/usr/www/fsite/htdocs";
$LocalURL = "http://xxx.xxx.xxx.xxx";
```

Chemin et URL absolus du répertoire principal devant être analysé dans le site.

```
$CGIPath = "/usr/local/apache/cgi-bin";
$CGIURL = "http://xxx.xxx.xxx.xxx/cgi-bin";
```

Chemin et URL absolus du répertoire hébergeant les scripts CGI.

```
$ImageMapPointer = "/cgi-bin/imagemap";
$ImageMapPath = "/usr/foo";
```

Variables utilisées pour aider le script à localiser correctement les fichiers image map. Ce paramètre n'est pas pris en compte ici, car le site n'emploie pas ce type de fichiers.

```
$SiteName = "Site INRA de Dijon";
```

Nom du site.

```
$Avoid = "notices\/.*\.html";
$Avoid = "sommaires\/.*\.html";
```

Cette variable représente une expression régulière qui identifie tous les fichiers qui ne doivent pas être analysés. Ici sont exclus tous les fichiers de notices et de sommaires (puisqu'ils constituent une base de données, et n'ont pas de liens directs avec le site).

```
$ParseCGI = "";
```

Représente tous les scripts CGI qui doivent être analysés. Sans valeur, comme ici, l'existence de chaque script sera notée, mais aucun d'entre eux ne sera exécuté. Ce paramètre est utile dans le cas où un script génère une page HTML devant être incluse dans la cartographie du site.

```
$ListBinaryLinks = 0;
```

Si la variable est initialisée à 0, les fichiers de type binaire seront exclus des listes "liens vers". Si par exemple de nombreux boutons de navigation apparaissent sur chaque page, la taille du rapport en sera nettement diminuée. Pour inclure ces fichiers, mettre la variable à 1.

```
$MissingLinks = 1;
```

Cette variable sera mise à 0 si l'analyse ne doit pas porter sur les fichiers existants mais non référencés. Ce qui est utile lorsque l'arborescence principale contient de nombreux fichiers sans relation avec le site.

```
$IgnoreExternals = 0;
```

Cette variable est initialisée à 0 afin que l'analyse inclue les liens externes.

```
$ShowOnlyErrors = 0;
```

Avec ce paramétrage, le script montre les erreurs et tout le reste.

```
$PrintDates = 1;
```

Avec ce paramétrage, la cartographie du site inclut pour chaque fichier la date de dernière modification.

```
$DaysNew = 7;
```

Initialisée à 7, la variable indique au script d'apposer la mention "NEW!" devant la date de modification (ou le nom du fichier si cette date n'est pas rapportée) de tout fichier ayant changé depuis 7 jours.

```
$MinLevel{'/usr/www/fsite/htdocs/document_test.html'} = 3;
```

Autorise un certain contrôle de la façon dont sera construite la carte du site. Le fichier `index.html` est placé au niveau 1, les fichiers référencés par lui sont au niveau 2, chacun pouvant référencer des fichiers alors au niveau 3 etc. Le "`minimum level`" pour un fichier sera son plus haut niveau d'apparition autorisé dans la carte du site. Par exemple, si la page `doc_niveau2.html` est référencée par `index.html`, mais doit apparaître dans la carte sous `page_niveau2.html`, elle aussi référencée par `index.html`, on assigne un "`minimum level`" égal à 3 à `doc_niveau2.html`.

Donner une valeur très élevée à cette variable permettra de ne pas faire apparaître certains fichiers dans la cartographie du site.

```
$Verbose = 1;
```

Paramétré ainsi, le script dispense des commentaires lors de son exécution. Une valeur 0 pour cette variable inactive les commentaires, mais laisse bien sûr apparaître un éventuel message d'erreur.

c. *Protocole d'utilisation*

Les mises à jour sont effectuées localement par le daemon cron. Celui-ci lance hebdomadairement le script `/usr/local/apache/cgi-bin/webtester_files/config.pl`, lequel écrit les fichiers `sitecheck.html` et `sitemap.html` dans le répertoire `/usr/www/fsite/htdocs/sitetools/`.

d. *Conclusion*

WebTester s'est avéré être un outil très précieux, puisqu'il a permis de repérer de nombreux fichiers et répertoires absents, après le transfert intégral de l'Intranet INRA Dijon sur le site de Dijon :

- 5 répertoires manquants.
- 61 fichiers manquants.
- 51 liens "morts".

La plupart de ces problèmes n'est due qu'au transfert, et à la modification de l'arborescence d'accueil.

2. Gestion du site via le Web

Assurer cette gestion régulièrement demandait d'adapter les outils mis à disposition à une utilisation aisée. Dans cette optique, la très grande majorité des applications installées a été rendue accessible directement depuis le site, qu'il s'agisse de l'administration ou de l'exploitation de ces outils. Naturellement, certaines applications ne concernent que l'administrateur, aussi une section a été mise en accès réservé : y pénétrer nécessite l'emploi d'un login et d'un mot de passe. La gestion des accès réservés est effectuée via la distribution Authorization Gateway.

Http://www.ray.org.hk/perl/

Erreurs d'exécution et mode de transfert des fichiers

Plusieurs autres scripts (Account Manager Lite, CheckMe) ont été essayés sans succès. Après installation et configuration, leur exécution sur le serveur produisait une "Internal server error", et une exécution locale via l'interpréteur Perl donnait le message suivant : "`Illegal character \015 (carriage return) at line 2`". Il est probable que ce problème est dû à un mauvais mode de transfert (un transfert ASCII est obligatoire pour télécharger des scripts). Pour ces fichiers, la compression était de type .zip (seul choix proposé). Authorization Gateway n'a en revanche pas présenté de problème, et était téléchargeable sous forme d'une archive compressée .tar.gz.

a. *Présentation d'Authorization Gateway*

Comme toutes les applications installées au cours de ce stage, Authorization Gateway fait partie de la famille des logiciels et scripts libres. Cette distribution présente la particularité de pouvoir être installée par un utilisateur "normal" (sans les droits de *root*). Il permet de protéger[22] un nombre illimité de répertoires et fichiers par un accès login - mot de passe, cache l'URL de la page source, et dispose d'une fonction d'éjection automatique.
La distribution est constituée des fichiers suivants :

- `auth.conf` : fichier de configuration.

[22] Les scripts libres proposés avouent la plupart du temps leurs limites en matière de protection . Si l'utilisateur "normal" est bloqué, stopper un utilisateur averti nécessitera d'utiliser des scripts plus élaborés (et payants).

- `asetup` : script shell qui assigne la permission adéquate à chaque fichier.
- `check.pl` : script qui vérifie et valide le fichier de configuration.
- `user.db` : base de données qui stocke tous les noms d'utilisateurs et les mots de passe associés.
- `html.db` : base de données qui stocke tous les noms de pages devant être protégées.
- `index.html` : page HTML demandant l'authentification de l'utilisateur.
- `main.html` : page HTML principale : sommaire proposant l'accès à toutes les pages protégées.
- `fail.html` : page HTML retournée lorsque l'identification d'un utilisateur échoue.
- `logoff.html` : page HTML retournée lorsqu'un utilisateur se déconnecte.
- `notf.html` : page HTML retournée lorsqu'une demande d'accès à une page n'aboutit pas.
- `auth.pl` : script principal.

b. *Installation et configuration*

➲ Editer le fichier `auth.conf` et y apporter les modifications nécessaires :

```
$userdb='/usr/local/apache/cgi-bin/auth-0.7.1/user.db';
```

Chemin absolu (fichier `user.db`).

Procéder de la même façon avec tous les autres fichiers de la distribution.

```
$expire=2;
```

Détermine le temps d'inactivité (en minutes) au bout duquel la déconnexion automatique se produit.

➲ Lancer le script `asetup`.

➲ Lancer le script `check.pl`.

➲ Editer le fichier `user.db` et entrer le nom de tous les utilisateurs ainsi que les mots de passe correspondants selon le modèle :

```
user1|password1
user2|password2
etc.
```

➲ Editer le fichier `html.db` et entrer les pages devant être protégées, selon le modèle :

```
alias1|chemin_absolu_fichier1
alias2|chemin_absolu_fichier2
etc.
```

L'alias est le nom présenté dans la page de sommaire "`main.html`".

➲ Enfin, éditer les pages HTML `index.html`, `main.html`, `fail.html`, `loffof.html` et `notf.html` et les personnaliser à volonté.

Dans le fichier `main.html`, un lien à une page protégée se fait par une ligne du type :

```
<A HREF="/cgi-bin/auth-0.7.1/auth.pl?alias1">Affichage du fichier1</A><P>
```

Le lien qui entraîne une déconnexion de l'utilisateur se fait par la ligne suivante :

```
<A HREF="/cgi-bin/auth-0.7.1/auth.pl?logoff">Log off</A><P>
```

c. *Conclusion*

De manière générale, les scripts payants offrent un niveau de protection plus élevé. Néanmoins, Authorization Gateway se révèle très simple à installer et à utiliser, avec en outre la fonction très intéressante d'auto-déconnexion, et la protection d'un nombre illimité de répertoires ou de pages HTML.

Cependant, un défaut majeur est mis en évidence lorsque l'utilisateur, à la fin de sa session, se déconnecte (le script prévoit une fonction de déconnexion manuelle) et retourne à d'autres pages. En effet, si un deuxième utilisateur passe derrière le premier et effectue un retour dans l'historique des pages visitées, il peut accéder à la page d'entrée du mot de passe. Et sans voir ce mot de passe (les caractères sont remplacés par des étoiles), il peut se reconnecter, puisque les étoiles n'ont pas été effacées. Deux modifications sont à apporter pour pallier ce problème :

➲ Dans l'en-tête de la page `index.html` (renommée `pass.html`), rajouter la ligne :

```
<META http-equiv="Jon Eyrick" content="no-cache">
```

De cette manière, cette page ne figurera pas dans le cache de la machine.

➲ Utiliser du JavaScript dans le code HTML afin d'ouvrir la session sur un deuxième navigateur, que l'on refermera en fin de session : l'historique du navigateur initial ne comportera donc aucune des pages de la session (ni celle d'entrée du mot de passe). Dans la page d'index réservée à la section Intranet du site (voir la partie consacrée au site INRA de Dijon), rajouter la ligne :

```
<INPUT              TYPE=BUTTON              VALUE="Identification"
onClick="window.open('pass.html')">
```

Le fichier `pass.html` contenant le formulaire de saisie du login et du mot de passe doit dans ce cas se trouver au même endroit que l'index.

B. En offrant de nouveaux services : traitement en ligne des commandes et des inscriptions

D'un point de vue général, la vie d'un site dépend en partie de l'intérêt que lui porteront ses visiteurs. Un moyen dont dispose l'administrateur pour entretenir cet intérêt est de montrer que l'on s'occupe du site, qu'on le fait évoluer. Nous ne rejoindrons évidemment pas des méthodes visant à modifier régulièrement et automatiquement la présentation d'un site alors que son contenu reste identique. En revanche, la création de nouveaux services destinés à faciliter le quotidien des utilisateurs ne peut être accueillie que favorablement.

Deux grandes catégories d'utilisateurs se détachent en ce qui concerne le site du centre, selon leur caractère extérieur ou non à l'INRA ; nous porterons en particulier notre attention sur la sous-classe des utilisateurs de Dijon, puisque nous avons toute latitude pour installer les applications de notre choix sur la plate-forme Linux.

Deux services nouveaux ont été créés sur le site, et automatisés par l'emploi de scripts CGI en Perl :

- Un service d'inscriptions, par exemple à un congrès ou à un séminaire organisé par le centre de Dijon.
  Cette application, jusqu'alors utilisée uniquement à des fins de tests, sera mise en fonction très prochainement à l'occasion du congrès DEFI'99 prévu en décembre. Le projet complet (dont le service d'inscriptions et les pages HTLM à installer sur le site) est cours d'élaboration.
  Cette application doit permettre la consultation en ligne des réponses à un formulaire, et envoyer la liste de tous les participants à chaque inscrit après la clôture des inscriptions. En outre, les données seront reformatées afin d'automatiser la création d'étiquettes au nom de chacun des participants, pour faire des badges.
- Un service de commandes (de photocopies ou de prêts d'ouvrages) en ligne.

1. Service d'inscriptions

a. *Protocole et conditions d'utilisation*

Du coté utilisateur, seul le formulaire d'inscription est accessible. Une fois la personne inscrite, elle reçoit un message de confirmation.
Du coté administrateur (accessible après identification) il est possible par simple click sur un hyperlien :

- d'envoyer la liste de tous les inscrits à chaque participant grâce au script `/cgi-bin/inscriptions/sendm.cgi`. Cette liste ne reprend que certains des champs du formulaire.
- de consulter cette liste sur le site par le script `/cgi-bin/inscriptions/liste.cgi`.
- de consulter la liste complète des inscrits, avec tous les champs par le script `/cgi-bin/inscriptions/formresult1.cgi`

Enfin, il a été prévu d'automatiser la création d'étiquettes (ou badges), grâce au script `etiq`, et aux macros Word "`étiquettesPréparation`" et "`étiquettes`" décrites en annexes.

Remarque - La distribution bnbform

Cette distribution (qui offre un formulaire, un service mail, et la possibilité de rapatrier les réponses dans un fichier texte) a été téléchargée via Internet, et constitue le "pilier" de ce service. Les formulaires d'inscriptions et de commandes sont issus de `bnbform`, de même que le fichier `form.log` qui regroupe les données rentrées par l'intermédiaire de ces formulaires. L'utilisation de cette distribution est décrite à la fin du chapitre. Les scripts qui suivent n'en font pas partie, et ont été conçus pour traiter, reformater, et présenter les données du fichier `form.log` à l'utilisateur et/ou à l'administrateur.

b. *Reformatage de la liste des participants*

Programme **`liste`**

La liste des participants est contenue dans le fichier texte `form.log`, créé par le script `bnbform.cgi`, et constitué d'autant de lignes que de réponses au formulaire. Une ligne est formée de champs séparés par le caractère "|".
Le programme `liste` se situe dans le répertoire `/usr/www/fsite/work/inscriptions/` ; son exécution en local a pour résultat de créer le fichier `liste.txt` à partir de `form.log`. Ce fichier est en fait une liste simplifiée des participants (comprenant leur nom, prénom, service, organisme, ville et e-mail), et servira de source de données pour les scripts `sendm.cgi` (application de messagerie électronique) et `liste.cgi` (affichage de la liste simplifiée sur le site).

```
#!/usr/bin/perl -w

open(DATA,"</usr/local/apache/cgi-bin/inscriptions/form.log");
@tab= <DATA>;
close(DATA);

@tabdef= grep(($_=~m/Monsieur/ || $_=~m/Madame/),@tab);

print "\n\n";
open(LISTE,">/usr/local/apache/cgi-bin/inscriptions/liste.txt");

print LISTE "NOM PRENOM SERVICE ORGANISME VILLE E-MAIL\n";
print LISTE "\n";
foreach (@tabdef) {
 s/Madame\||Monsieur\|//;
 }
@tabdefClass=sort @tabdef;
foreach (@tabdefClass) {
 @ligne= split(/\|/,$_);
 $ligne[0] =~ y/a-z/A-Z/;
 $ligne[0] =~ s/é|è/E/g;
```

```
 $ligne[3] =~ s/ /_/g;
 print "$ligne[0] $ligne[1] $ligne[3] $ligne[4] $ligne[7] $ligne[2]\n";
 print   LISTE   "$ligne[0]   $ligne[1]   $ligne[3]   $ligne[4]   $ligne[7]
$ligne[2]\n";
 }
 print   "\nCes   lignes   ont   été   enregistrées   dans   le   fichier
/usr/local/apache/cgi-bin/liste.txt.\n";

#@tabdefClass=sort @tabdef;
#foreach (@tabdefClass) {
#       s/Madame\||Monsieur\|//;
#}
#foreach (@tabdefClass) {
#print "\n$_";
#}


close(LISTE);
print "\n\n";
```

c. *Envoi de la liste de participants à chaque inscrit*

Script CGI **`sendm.cgi`**

Le script `sendm.cgi` exploite les données du fichier texte `liste.txt` pour envoyer à chaque inscrit la liste simplifiée de tous les participants dans un courrier électronique. Contrairement au script `liste`, `sendm.cgi` peut être lancé depuis le serveur.

```
#!/usr/bin/perl -w

open(LISTE,"<liste.txt");
@tab=<LISTE>;
@tabdef=grep($_=~m/\@/,@tab);
foreach (@tabdef) {
 print $_;
 @ligne = split(/ /,$_);
 print "$ligne[5]\n";
 $destmail = $ligne[5];
 $destmail =~ s/\@/\\\@/;
 print "$destmail\n";

 open (MESSAGE, "|mail $destmail");
 print MESSAGE "To : $destmail\n";
 print MESSAGE "Reply to : \n\n";
 print MESSAGE "Liste des participants :\n";
 print MESSAGE "_____________________\n\n";
 foreach (@tabdef) {
       print MESSAGE "$_";
       }
 print MESSAGE "_____________________\n\n";
 print MESSAGE "  Merci pour votre participation et à bientôt !\n\n";
 close (MESSAGE);
 }
close(LISTE);
```

d. *Mise en ligne de la liste simplifiée des inscrits*

Script CGI **`liste.cgi`**

De la même façon, ce script utilise `liste.txt`, et crée dynamiquement une page HTML présentant sur le site la liste simplifiée.

```
#!/usr/bin/perl -w

use CGI;
$html= new CGI;

print $html->header;
open(FORM,"<liste.txt");

print <<EOF;
<HTML>
<HEAD>
<TITLE>Insriptions</TITLE>
</HEAD>
<BODY BGCOLOR=#EEEADC>
<H1        ALIGN=CENTER><FONT        COLOR=#AA0000>        Liste        des
participants</FONT></H1><HR><BR>

<TABLE BORDER=2 ALIGN=CENTER CELLPADDING=5 CELLSPACING=1>
EOF
while (<FORM>) {
 print "<tr><td align=center>\n";
 s/ /<td align=center>/g;
 s/_/ /g;
 print;
 }

print <<EOF;
</TABLE>

<BR><BR>
<H5 ALIGN=CENTER>
<A HREF="http://xxx.xxx.xxx.xxx">Retour page principale</A>
<HR SIZE=1 NOSHADE WIDTH=50%>

</BODY>
</HTML>
EOF
close(FORM);
exit;
```

e. *Mise en ligne de la liste complète des inscrits*

Script CGI **`formresult1.cgi`**

Ce script, très voisin de `liste.cgi` dans sa conception comme dans son utilisation, ne sera pas reporté ici. Il utilise les données de `form.log` pour construire dynamiquement une page HTML présentant la totalité des données entrées par chaque participant via le formulaire.

f. *Création automatique d'étiquettes*

Cette application est réalisée à l'aide des outils de publipostage de Word, et au langage de Macros associé à Word. La source de données est préparée par l'intermédiaire du script `etiq` situé dans le répertoire `/usr/www/fsite/work/inscriptions/`.

Deux étapes succèdent à cette phase préparatoire : créer le format d'étiquettes désiré, puis réaliser le publipostage proprement dit. Ces deux dernières étapes sont reportées en annexe afin de ne pas alourdir le mémoire.

Programme **etiq**

La source de données pour le script est encore une fois le fichier `form.log`. Suite à l'exécution de ce programme, un fichier `etiq.txt` est créé, et constituera la matière première pour la suite des opérations.

```
#!/usr/bin/perl -w

open(DATA,"</usr/local/apache/cgi-bin/inscriptions/form.log");
@tab= <DATA>;
close(DATA);

@tabdef= grep(($_=~m/Monsieur/ || $_=~m/Madame/),@tab);
print "\n\n";
open(ETIQ,">etiq.txt");
foreach (@tabdef) {
 @ligne= split(/\|/,$_);
 $ligne[4] =~ s/ /_/g;
 print "$ligne[1] $ligne[2] $ligne[4] $ligne[8]\n";
 print ETIQ "$ligne[1] $ligne[2] $ligne[4] $ligne[8]\n";
 }
 print "\nCes lignes ont été rentrées dans le fichier
/usr/www/fsite/work/inscriptions/etiq.txt.\n";
close(ETIQ);
print "\n\n";
```

Concrêtement, la marche à suivre pour créer une série d'étiquettes est très simple :

➲ Lancer localement le script `etiq` en tapant :

```
% etiq
```

Il n'est pas nécessaire de se placer dans le répertoire où se trouve le script, puisque son chemin d'accès a été rajouté à la variable d'environnement PATH.

➲ Il ne reste alors plus qu'à rapatrier le fichier obtenu sur la plate-forme où se trouvent installés Word et ses macros "`étiquettesPréparation`" et "`étiquettes`", puis lancer successivement ces deux dernières.

g. *La clé de voûte de l'application : le script CGI* ***bnbform.cgi***

☞ Http://bignosebird.com/carchive/bnbform.shtml

➢ Présentation

Le script `bnbform.cgi` concatène les réponses à un formulaire dans un fichier texte selon le format suivant : chaque réponse forme une ligne ; chaque élément du questionnaire (zone de texte, case à cocher, etc.) correspond à un champ ; enfin, le séparateur de champ est constitué du caractère "|".

Ce fichier, après deux réponses au questionnaire, pourrait se présenter ainsi :

Ligne 1 `|NOM|PRENOM|ENVOYE PAR|SERVICE/LABO|ORGANISME|ADRESSE_____|CODE POSTAL|VILLE|TELEPHONE_|FAC-SIMILE_|DEJEUNER|ATELIER 1|ATELIER 2|DATE_D'ENVOI|`

Ligne 2

Ligne 3

Ligne 4 `Monsieur|Liénard|Christophe|lienard@dijon.inra.fr|URD|INRA|17 rue de Sully BV 1540|21000|Dijon|03 80 69 32 03|12 12 12 12 12|Oui||Présent|Mon Aug 9 14:56:44 1999|`

Ligne 5
```
Madame|Untel|Alice|alice@mailfictif.com|néant|néant|rue de la Liberté|21000|Dijon|03 86 34 34 34|45 45 45 45 45 |Non|Présent |Présent|Mon Aug 9 15:03:40 1999|
```

Chaque nouvelle inscription aura pour effet de rajouter une ligne en fin de fichier. Deux courriers électroniques sont aussi envoyés, l'un au responsable des inscriptions (contenant les données rentrées par le nouvel inscrit), et l'autre au nouvel inscrit pour lui signaler que son inscription a bien été enregistrée.

Exemple de message envoyé au responsable :

On Mon Aug 9 14:56:44 1999,
The following information was submitted:
Host: xxx.xxx.xxx.xxx
etat = Monsieur
nom = Liénard
prenom = Christophe
submit_by = lienard@dijon.inra.fr
service = URD
organisme = INRA
adresse = 17 rue de Sully BV 1540
code = 21000
ville = Dijon
tel = 03 80 69 32 03
fax = 12 12 12 12 12
dejeuners = Oui
atelier1 =
atelier2 = Présent

Chaque champ obligatoire fait l'objet de tests de saisie : rentrer "lienard@dijon" dans le champ "E-mail" provoque l'envoi d'un message d'erreur (page HTLM définie par l'administrateur).

➢ Installation et exploitation

➲ Le script est installé dans le répertoire `/cgi-bin/inscriptions` du serveur, et ne nécessite pas de modification.

Le fichier texte contenant les inscriptions est réenregistré automatiquement à chaque nouvelle inscription sous le nom "`form.log`" dans le même répertoire.

➲ L'ensemble du paramétrage s'effectue par modification de la page HTML contenant le formulaire d'inscription. Ci-dessous se trouve la page réalisée pour le formulaire d'inscription ; les caractères en rouge correspondent aux modifications à apporter pour rajouter un champ (non obligatoire) dans le formulaire. La suppression ou la modification de champs existants s'effectuent également à ces endroits.

```
<HTML>
<HEAD><TITLE>Formulaire d'inscription</TITLE></HEAD>
<BODY BACKGROUND="../images/bg524.jpg">
<TABLE>
<TR><TD><IMG SRC="../images/blmail.gif"></TD>
<TD VALIGN=CENTER><FONT COLOR=#AA0000 SIZE=+4> Section Inscriptions</FONT></TD></TR>
</TABLE><HR><BR>

<FORM METHOD="POST" ACTION="/cgi-bin/inscriptions/bnbform.cgi">
<PRE>
 <INPUT TYPE="RADIO" NAME="etat" VALUE="Monsieur" CHECKED> Monsieur
<INPUT TYPE="RADIO" NAME="etat" VALUE="Madame"> Madame

 Nom              <INPUT TYPE="TEXT" NAME="nom" SIZE=35 MAXLENGTH=50>
```

```
  (obligatoires)

 Pr&eacute;nom                  <INPUT  TYPE="TEXT"  NAME="prenom"  SIZE=35
MAXLENGTH=50>
  (obligatoire)

 Adresse   E-Mail          <INPUT   TYPE="TEXT"   NAME="submit_by"   SIZE=35
MAXLENGTH=50>
  (obligatoire)

 Service/Labo      <INPUT TYPE="TEXT" NAME="service" SIZE=50>

 Organisme         <INPUT TYPE="TEXT" NAME="organisme" SIZE=50>

 Adresse           <INPUT TYPE="TEXT" NAME="adresse" SIZE=50>

 Code postal        <INPUT TYPE="TEXT" NAME="code" SIZE=5>   Ville <INPUT
TYPE="TEXT" NAME="ville" SIZE=50>

 T&eacute;l :             <INPUT TYPE="TEXT" NAME="tel" SIZE=20>

 Fax :             <INPUT TYPE="TEXT" NAME="fax" SIZE=20>

 D&eacute;jeuners                   <INPUT  TYPE="RADIO"  NAME="dejeuners"
VALUE="Oui"   CHECKED>   Oui      <INPUT   TYPE="RADIO"   NAME="dejeuners"
VALUE="Non"> Non

 Participation   &agrave;        <INPUT   TYPE="CHECKBOX"   NAME="atelier1"
VALUE="Pr&eacute;sent"> Atelier 1
                    <INPUT         TYPE="CHECKBOX"         NAME="atelier2"
VALUE="Pr&eacute;sent"> Atelier 2

                         <INPUT  TYPE="SUBMIT"  VALUE="Envoyer">    <INPUT
TYPE="RESET" VALUE="R&eacute;initialiser">
</PRE>

<INPUT TYPE="HIDDEN" NAME="required" VALUE="nom,prenom,submit_by">
<INPUT  TYPE="HIDDEN"  NAME="data_order"  VALUE="etat,nom,prenom,submit_by,
service,organisme,adresse,code,ville,tel,fax,dejeuners,atelier1,atelier2"
>
<INPUT TYPE="HIDDEN" NAME="submit_to" VALUE="lienard@dijon.inra.fr">
<INPUT TYPE="HIDDEN" NAME="autorespond" VALUE="yes">
<INPUT TYPE="HIDDEN" NAME="automessage" VALUE="mymessage.txt">
<INPUT TYPE="HIDDEN" NAME="outputfile" VALUE="form.log">
<INPUT TYPE="HIDDEN" NAME="form_id" VALUE="Inscription">
<!-- <INPUT TYPE="HIDDEN" NAME="ok_url"
     VALUE="http://xxx.xxx.xxx.xxx/cgi-bin/thanks.html">
<INPUT TYPE="HIDDEN" NAME="not_ok_url"
     VALUE="http://xxx.xxx.xxx.xxx/cgi-bin/oops.html"> -->
<!--  END OF SCRIPT CONFIGURATION SECTION -->

</FORM></BODY></HTML>
```

L'explication de l'utilisation des balises <INPUT> est reportée ci-dessous :

```
<INPUT TYPE="HIDDEN" NAME="required" VALUE="nom,prenom,submit_by">
```

Liste des champs obligatoires.

```
<INPUT TYPE="HIDDEN" NAME="data_order" VALUE="etat,nom,prenom, … ">
```

Liste (dans l'ordre) de l'ensemble des champs du formulaire.

```
<INPUT TYPE="HIDDEN" NAME="submit_to" VALUE="lienard@dijon.inra.fr">
```

Adresse électronique du responsable à qui seront envoyées les réponses.

```
<INPUT TYPE="HIDDEN" NAME="autorespond" VALUE="yes">
```

Lorsque le paramètre VALUE est sur "yes", une réponse prédéfinie est envoyé au nouvel inscrit.

```
<INPUT TYPE="HIDDEN" NAME="automessage" VALUE="mymessage.txt">
```

Nom (et emplacement) du fichier texte contenant la réponse prédéfinie. Dans ce cas, le fichier se trouve dans le même répertoire que le script.

```
<INPUT TYPE="HIDDEN" NAME="outputfile" VALUE="form.log">
```

Nom (et emplacement) du fichier texte où sont concaténées toutes les réponses au formulaire.

```
<INPUT TYPE="HIDDEN" NAME="form_id" VALUE="Inscription">
```

Texte constituant le sujet du message envoyé au nouvel inscrit.

2. Le service de commandes

L'application relative au service de commandes est en fait une adaptation simplifiée de celle du service d'inscriptions, puisque seules trois tâches devront être effectuées suite à l'envoi d'une commande : concaténer les données à un fichier texte, et envoyer deux courriers électroniques, l'un à l'administrateur du service de commandes, l'autre au demandeur pour lui confirmer la prise en compte de sa commande. Cette application ne sera donc pas décrite ici.

VIII. Considérations générales sur la sécurité

A. Sauvegardes du site

- L'ensemble du travail effectué durant ce stage se trouve dans l'arborescence du répertoire `/usr/.`

Une première sauvegarde de la totalité des répertoires et fichiers contenus dans `/usr/` a été effectuée après la fin des tests par la commande :

```
tar -cvf /usr/ | gzip -9c > usrsave.tar.gz
```

Pour la syntaxe des commandes `tar` et `gzip`, se reporter au chapitre relatif à l'exploitation de Linux".

- Un deuxième type de sauvegarde est réalisé quotidiennement (la nuit), sur les fichiers récemment modifiés. Cette opération est effectuée grâce à la commande :

```
find ./repertoire -mtime 0
```

Le rajout d'une ligne dans la "crontable" permet d'automatiser cette tâche.
Les fichiers sauvegardés sont dans un premier temps envoyés par FTP sur le serveur de centre. Par la suite, ils seront enregistrés sur bandes, sur une plate-forme Sun prévue à cet effet, qui pose malheureusement quelques problèmes de configuration matérielle pour l'instant.

B. Estimation des risques et solutions mises en œuvre

Les véritables numéros IP sont masqués pour des raisons de sécurité. En résumé, deux numéros IP sont en fait attribués à la plate-forme Linux, un numéro routable (yyy.yyy.yyy.yyy) et un numéro non routable (xxx.xxx.xxx.xxx).

L'adressage du centre de Dijon est un adressage IP avec pseudo-classe A : tous les appareils du centre ont un numéro IP non routable dont les deux premiers octets sont communs. Pour sortir du centre, le routeur affecte dynamiquement une autre adresse qui, elle, est routable (translation d'adresse).
Ce qui a été effectué au départ pour le serveur de l'UPE-Doc, a été de créer une correspondance dans le routeur entre le numéro IP de la plate-forme (`xxx.xxx.xxx.xxx`) et une adresse IP fixe et routable (`yyy.yyy.yyy.yyy`).
Une personne appartenant au centre pouvait alors contacter le site hébergé par le serveur de l'UPE-Doc à l'adresse http://xxx.xxx.xxx.xxx/, et une personne extérieure, elle, devait utiliser http://yyy.yyy.yyy.yyy/.

Une restriction d'accès au serveur pour les seules machines de l'INRA avait été paramétrée à partir des numéro IP des différentes plates-formes. Des tests ont été réalisés (des tentatives de connexion de différents endroits), validant ainsi cette mesure de sécurité. Cependant, compte tenu de la faible fiabilité de cette mesure, la décision a été prise de restreindre l'accès du serveur aux seules machines du centre de Dijon, en supprimant la correspondance dans le routeur. Le serveur se trouve dès lors à l'abri des "turpitudes du monde extérieur", protégé par les systèmes de sécurité du centre.

Conclusion générale et bilan du stage

Construire un système d'information fiable et performant sur la seule base de l'exploitation de logiciels dits "open sources", pouvait sembler fantaisiste voire utopique si l'on remonte à quelques années. Ce qui est beaucoup moins vrai actuellement. D'une part, il ne faut pas perdre de vue qu'Internet, l'"autoroute" de l'information, repose en grande partie sur le concept de logiciel libre : DNS, le système qui relie un nom de domaine à son adresse IP, et Sendmail, pour la transmission du courrier, sont deux projets libres ; un des sites Web les plus visités au monde, Yahoo!, utilise un serveur libre, Apache, sur un système d'exploitation libre, FreeBSD, ainsi que le langage de script libre Perl pour ses scripts CGI [10]. D'autre part, le renouveau du logiciel "open source" se précise, en particulier depuis l'expansion sans précédent d'un système d'exploitation libre, Linux, pour lequel des éditeurs prestigieux (Corel, IBM, Oracle, Sybase) ont annoncé des plans de portage de leurs produits. D'autant plus qu'Apple, Netscape ou Sun Microsystems ont adopté le modèle "open source" pour leur propres produits.

J'ai éprouvé une grande motivation à découvrir ce courant dit "open source", et ainsi utiliser des produits en pleine ascension et à très fort potentiel comme Linux ou Apache. Tout d'abord pour toutes les possibilités qu'apportent ces logiciels. Ensuite, pour l'indépendance qu'assure le concept même du logiciel libre, lequel constitue une philosophie de l'informatique radicalement opposée à celle prônant l'approche commerciale à outrance. Enfin pour l'enthousiasme et l'activité remarquables que l'on peut observer autour des projets de type "open source", mêlant aujourd'hui utilisateurs "normaux", passionnés désintéressés et hommes d'affaires.

Il est ressorti de ce stage que l'utilisation de logiciels libres demandait, en particulier pour le néophyte, un certain temps d'adaptation. Au contraire d'un produit commercial, où cliquer à trois ou quatre reprises au bon endroit suffit généralement à l'installation, le produit "open source" peut demander un investissement en temps non négligeable. Même si les sources d'aides, pour peu qu'on sache les trouver, sont considérables dans le cadre des grands projets. Cependant, cette approche ouvre une perspective tout-à-fait intéressante. La découverte de l'application se trouve scindée en deux parties distinctes : d'un côté, l'apprentissage "classique" visant à maîtriser l'utilisation de l'application, et de l'autre côté l'apprentissage de son installation proprement dite. Un apprentissage qui selon le niveau de l'utilisateur, peut s'avérer relativement long, mais a le mérite d'aboutir sur la connaissance approfondie d'un logiciel, au fur-et-à-mesure que l'on arrive à affiner son paramétrage.

Inconvénient, un problème d'installation sérieux est susceptible de paralyser l'utilisateur moyen, qui n'aura alors d'autre recours que de trouver une solution de rechange. A ce sujet, le cas de SFgate a suscité une réflexion. Il a été constaté que l'accès à une même information pouvait découler de techniques très dissemblables. Au gré des aléas rencontrés pendant ce stage, le projet de recherches documentaires sur les notices des monographies a évolué d'une recherche sur champs structurés au sein de fichiers texte, vers une recherche en texte intégral sur pages HTML. Le résultat final étant des plus satisfaisants.
Par ailleurs, la solution comportant FreeWAIS-sf et le module SFgate s'est révélée relativement lourde à mettre en place, même si l'on fait abstraction des problèmes de compilation qui ont paralysé l'application jusqu'à l'intervention d'un informaticien professionnel. Les ressources utilisées en terme de stockage de données sont également plus importantes, puisque FreeWAIS-sf utilise tout un jeu d'index dans le but de potentialiser la qualité de recherche. De son côté, le moteur Xavatoria s'est avéré particulièrement simple et rapide à mettre en place. Il est clair que dans notre cas, la solution de facilité passait par un moteur de recherche, et ceci, sans nuire à la qualité du service proposé.

Si l'on élargit cette réflexion, il apparaît évident que la constitution d'une base de données formée de textes intégraux, pour laquelle un moteur est parfaitement adapté, est sans commune mesure avec celle d'une base de notices bibliographiques. Aussi le choix d'une technique peut s'avérer délicat, d'autant plus que les capacités de traitements des plates-formes informatiques croissent exponentiellement. Quelle est la limite à partir de laquelle la vitesse d'indexation de documents en texte intégral est préférable à la pertinence d'un système reposant sur une indexation manuelle par champs ?

On peut considérer le bilan du stage comme positif, puisque toutes les applications devant être installées l'ont finalement été avec succès. Cependant la durée relativement courte du stage n'a pas permis l'exploitation complète des nouvelles ressources ; aussi le travail effectué appelle une continuité dans plusieurs directions :

- Transfert sur le serveur de Jouy-en-Josas de certaines applications installées le serveur de l'UPE-Doc, dont le service d'inscription au congrès.
- Développement de l'indexation par FreeWAIS-sf (seule une base de test a pour l'instant été expérimentée) et exploitation approfondie de l'interface Web SFgate. Le forum doit lui aussi être concrètement utilisé.
- Poursuite de la mise en place de pages HTML, en particulier pour préparer le congrès DEFI'99.
- Affinement de la hiérarchie au sein de l'arborescence du serveur de l'UPE-Doc. Les diverses applications installées, le remaniement du site et son développement ont entraîné un ajout important de répertoires et de fichiers. La majeure partie des modifications du site étant maintenant effectuées, il est plus facile d'améliorer la cohérence de l'ensemble. Les évolutions futures du site en seront ainsi facilitées.
- Régler des détails pratiques pour la mise en service du serveur, comme par exemple le verrouillage de la plate-forme Linux pour les utilisateurs.
- Enfin, cette plate-forme étant dédiée à Linux, il semble logique d'explorer certains des outils phare du monde Linux, tels que la suite StarOffice.

IX. Bibliographie thématique et ressources Internet

A. Bibliographie thématique

Ne sont répertoriés ici que les articles électroniques ou papier, et les ouvrages papier. Les sites sont reportés au fur-et-à-mesure de leur exploitation dans le mémoire.

1. Linux

[1] http://www.lemonde.fr/article/0,2320,dos-2372-4340-QUO-1-2031-,00.html
"Comment Microsoft se prépare à affronter le système d'exploitation Linux", *Le Monde*, 10 novembre 1998
[2] http://www.lemonde.fr/article/0,2320,dos-2372-19161-QUO-1-2031-,00.html
"Linux connaît une croissance spectaculaire", *Le Monde*, mercredi 18 août 1999
[3] TACKETT J., BURNETT J.S. - Linux, Le Macmillan Edition 1999 - Paris : Simon & Schuster Macmillan, 1999, 866 p., 2-74440-0567-3
[4] WELSH M., KAUFMAN L. - Le système Linux - Paris : Editions O'Reilly, 1999, 595 p., 2-84177-033-8

2. Apache

[5] http://www.cmpnet.fr/se/directlink.cgi?INF19990521S0021
"Quatre serveurs Web pour l'entreprise", Informatiques Magazine, 21 Mai 1999
[6] http://apache.org/ABOUT_APACHE.html
"About the Apache HTTP Server Project", février 1999
[7] LAURIE B., LAURIE P. - Apache, The Definitive Guide - Sebastopol (USA) : 1999, Edtions O'Reilly, 375 p., 1-56592-528-9

3. FreeWais-sf et Z39-50

[8] http://www.acctbief.org/avenir/z3950.htm
"L'AVENIR DES FORMATS DE COMMUNICATION - Z39.50 et l'échange de données bibliographiques en ligne", Jean Marc Czaplinski, Direction de l'informatique et des nouvelles technologies, Bibliothèque nationale de France, 8 octobre 1996

4. Langages de scripts

[9] SCHWARTZ R.L., CHRISTIANSEN T. - Introduction à Perl - Paris : 1998, Editions O'Reilly, 305 p., 2-84177-041-9
[10] MEDINETS D. - Perl 5 By Example - 1996, Que, 658 p., 0-78970-866-3
http://www.itlibrary.com/reference/0789708663.html
[11] CHALEAT P., CHARNAY D. - Programmation HTML et JavaScript - Paris : 1999, Eyrolles, 454 p., 2-212-09024-2

5. Le logiciel libre

[12] TRAN P., "Le logiciel retourne à ses sources", *PC Expert*, Mai 1999

[13] http://www.lemonde.fr/article/0,2320,dos-2372-20677-QUO-1-2031-,00.html
"Logiciels libres : l'offensive de Sun Microsystems", *Le monde*, mardi 31 août 1999

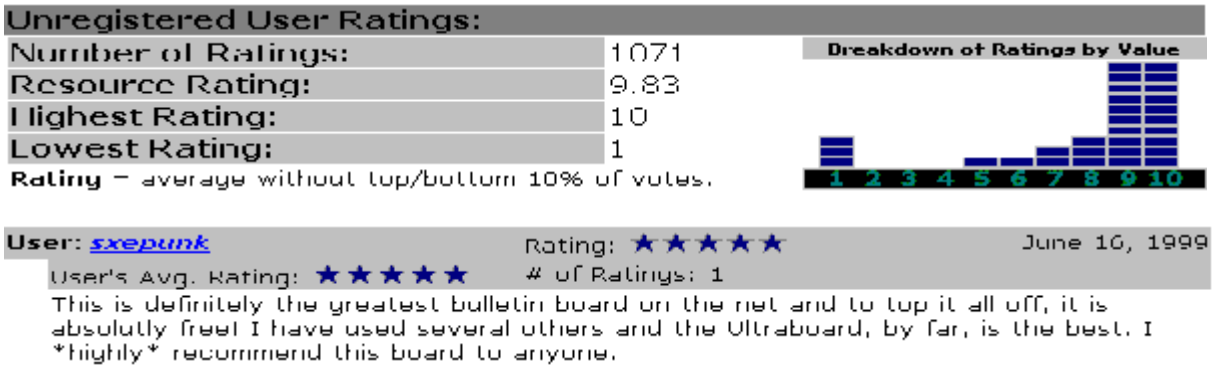
B. Ressources générales sur Internet

Ces ressources ont été abondamment utilisées au cours du stage.

1. Scripts CGI

http://cgi.resourceindex.com/Programs_and_Scripts/
The CGI Resource Index / Programs and Scripts

Plus de 700 scripts C, C++, Remotely Hosted, AppleScript, AppleScript, Tcl, Unix Shell ou Visual Basic, et près de 1600 scripts en Perl, classés par catégories. Celles ayant été utilisées (parmi les scripts Perl) sont Bulletin Board Message Systems, Form Processing, Miscellaneous, Password Protection et Searching. Des commentaires et une notation issue des utilisateurs sont disponibles pour de nombreux scripts. L'échantillon suivant concerne le script UltraBoard.



http://scriptsearch.internet.com/
Scripts Search

Près de 4300 scripts Applescript, C/C++, Combination, Java, JavaScript, PHP/FI, Remotely Hosted, Tcl, Unix Shell, VBScript, ou Visual Basic, et plus de 1200 scripts Perl, rangés par catégories. Celles utilisées ont été BBS, Miscellaneous et Security

http://freecenter.com/cgi.htm
FreeCenter / Free CGI Scripts

Recense 18 banques de scripts libres notées par le FreeCenter (dont The CGI Resource Index et Scripts Search, les deux plus importantes) soit un ensemble d'environ 7500 scripts, la plupart écrits en Perl.

2. Librairie informatique en ligne

http://itlibrary.com/
Un choix très important de livres en ligne (langue anglaise), accessibles gratuitement et en texte intégral, sur les rubriques suivantes :
Programming Languages, Databases, Security, Web Services, Network Services, Middleware, Components, Operating Systems, User Interfaces, Groupware & Collaboration, Content Management, Productivity Applications, Hardware, Fun & Game.
Un site qui s'est révélé particulièrement précieux pour résoudre différents problèmes relatifs à Perl, Javascript.

3. Le logiciel libre

http://www.opensource.org/

Organisation indépendante créée il y a un an, dans le but d'élargir le mouvement Open Source. Définition officielle du logiciel libre, liens vers des projets importants.

http://opensource.oreilly.com/

L'éditeur O'reilly s'est spécialisé dans les ouvrages sur le logiciel libre, et propose entre autres des liens vers les projets "open sources" majeurs.

http://www.aful.org

Association francophone des utilisateurs de Linux et des logiciels libres.

X. Annexes

A. Le site de Dijon

1. La page d'accueil

L'ancienne page est accessible à l'adresse: http://www.inra.fr/Dijon/index_bis.html
Pour des questions de place, seule la nouvelle version est présentée ici.

2. Règles éditoriales du serveur

Règles d'édition du serveur du centre INRA de Dijon

- Mettre le logo INRA (`mini-inr3.gif`) en haut de page à gauche sur toutes les pages "INRA".
- Utiliser une "barre" pour séparer le titre de la page du corps de la page. Utiliser la même barre en fin de page.
- Faire un lien de retour sur chaque page vers la page principale du répertoire hiérarchique immédiatement supérieur et/ou vers la page d'accueil du serveur comme suit avec l'image `toit.gif`, son adresse étant : http://www.inra.fr/Dijon/
- Mettre ces liens sous la barre de fin de page.
- Signer chaque page avec le nom et le prénom du rédacteur et/ou son adresse e-mail (avec lien sur la procédure "mailto").
- Sur les pages strictement INRA, ne pas oublier la mention : Copyright © 1999, INRA, Tous droits réservés.
- Respecter les règles de droit d'auteur, de propriété intellectuelle et de reproduction (articles soumis ou publiés, photos, dessins, ...)
- Mettre la date de dernière mise à jour sur chaque page.
- Ne jamais se référer ou faire de lien vers des pages du serveur non remplies.
- Ne pas mettre des informations de nature à nuire à l'image de l'INRA, d'ordre privé ou de nature commerciale.
- Prendre en compte les machines bas de gamme dans l'édition des pages.
- Vérifier ses pages sur plusieurs types de navigateurs (Netscape, Internet Explorer, ...).
- Eviter de répéter des informations déjà présentes sur le serveur du centre ou de l'INRA national. Mettre plutôt un pointeur (lien) vers ces informations qui sont mises à jour régulièrement.
- Eviter les pages trop longues. Sinon, utiliser des menus (ou sommaires) avec des "ancres" pour que l'utilisateur trouve rapidement l'information qu'il cherche. Mettre un titre significatif à chaque page, ainsi que des mots-clés significatifs (en français et éventuellement en anglais) pour une meilleure visibilité du serveur sur le Web.

B. Création d'étiquettes à l'aide de macros Word

Remarque - Convention d'écriture pour la description des macros

Taper sur <CTRL a> signifie appuyer sur les touches "contrôle" et "a".

- Macro" `étiquettesPréparation`"

Création de la macro "`étiquettesPréparation`":

➲ Dans Word, sélectionner le menu *Outils/Macro/Nouvelle macro…* .

➲ Nommer la macro "`étiquettesPréparation`" et valider. L'enregistrement de la macro commence alors, et toutes les actions de l'utilisateur sont transcrites en langage VBA (Visual Basic for Applications).

➲ Sélectionner le menu *Fichier*, ouvrir le fichier `etiq.txt` en format "Texte seulement".

➲ Taper sur <CTRL a>, ce qui sélectionne l'ensemble du texte, puis aller dans le menu *Tableau/Convertir texte en tableau…* . Dans la boîte de dialogue, au niveau de "Séparer le texte au niveau des", choisir "Autres" et remplacer le caractère existant dans la zone de texte par un espace. Le nombre de colonnes et de lignes doit apparaître automatiquement dans la boîte de dialogue.

➲ Valider puis sélectionner le menu *Edition/Remplacer*. Effectuer le remplacement du caractère "_" par un espace. Aller dans le menu *Fichier/Enregister sous…* et choisir le format Word (le format texte ne permettrait pas sauvegarder la mise en forme en tableau).

➲ Fermer la fenêtre du document et appuyer sur le bouton d'arrêt d'enregistrement des macros (dans la petite fenêtre qui a été présente durant tout l'enregistrement).

Echantillon du fichier etiq.txt (en entrée de macro)

```
nom prénom labo
Liénard Christophe Unité_Présidence_Equipe_Documentation
Nom1 Prénom1 Labo1 Provenance1
Nom2 Prénom2 Labo2 Provenance2
```

Echantillon du fichier `etiq.doc` (en sortie de macro)

Nom	Prénom	Labo	Provenance
Liénard	Christophe	Unité Présidence Equipe Documentation	Dijon

- Macro "étiquettes"

Création de la macro "`étiquettes`"

➲ Lancer l'enregistrement d'une nouvelle macro, que l'on nommera "`étiquettes`"

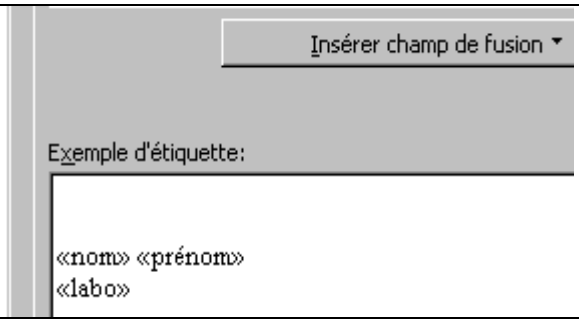
➲ Taper <CTRL n> pour ouvrir un nouveau document, et aller dans le menu *Outils/Publipostage…* .

➲ Dans la nouvelle fenêtre, aller dans *Créer/Etiquettes de publipostage*, et choisir *Fenêtre active*.

➲ Aller dans *Obtenir les données/Ouvrir la source de données…* . Sélectionner le fichier `etiq.doc`.

➲ Valider *Préparer les données*: une nouvelle fenêtre s'ouvre. Dans la section "Imprimante", sélectionner *Laser et jet d'encre*, et *Alimentation : Tray 1*. Dans la section "Numéro de référence", sélectionner le format d'étiquette *etiq - personnaliser*.

➲ La fenêtre "Etiquette" s'ouvre. Sélectionner *Insérer un champ de fusion*, et insérer successivement les quatre champs disponibles (nom, prénom, labo et provenance) en adoptant la disposition suivante:

	Format adopté : - Deux lignes vides - Champ nom séparé par un espace du champ prénom - Champ labo. - Champ provenance Cette disposition est indispensable pour que l'ensemble texte-image tienne dans l'étiquette.

➲ Sélectionner *Fusionner…*, puis également *Fusionner* dans la fenêtre qui s'ouvre.

➲ On revient à un document Word, que l'on ferme sans en enregistrer les modifications. On aboutit sur un autre document Word qui constitue le modèle de publipostage des étiquettes. Sa modification permettra la mise en forme des étiquettes, et l'ajout du logo.

Remarque - Modèle des étiquettes

Le modèle, dans notre cas, comporte 10 étiquettes : on insère donc 20 logos (10 de l'INRA et 10 du centre de Dijon). Ce modèle servira ensuite pour l'ensemble des étiquettes à créer : si le tableau `etiq.doc`

comporte 53 personnes, 6 pages d'étiquettes seront créées, soit 53 étiquettes complètes, et 7 étiquettes vierges ne comportant que les 2 logos.

➲ Taper <CTRL a>, centrer l'ensemble du texte, et le mettre en gras avec une police Times New Roman, de taille 12.

➲ On insère le logo dans chacun des 10 enregistrements du modèle (au premier caractère pour la première étiquette, et juste après le champ "Enregistrement suivant" pour les 9 autres) : aller dans le menu *Insertion/Image/A partir du fichier…*, puis sélectionner le logo de l'INRA : fichier `logoinra.jpg`. Réitérer l'opération avec le logo du centre de Dijon (fichier `toit.gif`)

➲ Dans la barre d'outil de publipostage, sélectionner l'icône *Fusionner vers un nouveau document*, puis arrêter l'enregistrement de la macro. Enfin, enregistrer le document (qui réunit l'ensemble des étiquettes dans leur état final) sous le nom désiré et fermer sans enregistrer les modifications les autres documents (qui ont servi au publipostage).

Echantillon du fichier en sortie de macro (le nom du fichier est laissé à l'appréciation de l'utilisateur).

  **Liénard Christophe** Unité Présidence Equipe Documentation Dijon	  **Untel Antoine** Néant Paris

C. Intervention de M. Caron

Ci-dessous est reporté le compte rendu rédigé par M. Caron suite à son intervention pour l'installation du module `wais.pm`. Ce compte rendu a été laissé dans l'état afin de ne pas risquer de modifier des paramètres.

1. Freewais-sf

```
./Configure
make depend  (ne marche pas)
faire:
touch ./bin/Jmakefile
touch ./doc/original-TM-wais/manl/Jmakefile
touch ./indexer/Jmakefile
touch ./lib/ctype/Jmakefile
touch ./lib/ftw/Jmakefile
touch ./lib/ir/Jmakefile
touch ./lib/regexp/Jmakefile
touch ./lib/Jmakefile
touch ./server/Jmakefile
touch ./test/Jmakefile
touch ./ui/Jmakefile
touch ./Jmakefile
```

et relancer ./Configure
=>genre config.h et confmagic.h : pas de modifs
=>config.sh contient les PATH..., compilo : en gros les reponses et de
detections faites par Configure
make depend
make
make test
make install
make install.man
=>install sous /usr/local/bin

```
-r-xr-xr-x   1 root     root       658244 Aug 23 11:53 waisindex
-r-xr-xr-x   1 root     root       707524 Aug 23 11:53 waisserver
-r-xr-xr-x   2 root     root       725372 Aug 23 11:53 waisq
-r-xr-xr-x   1 root     root       704908 Aug 23 11:53 waissearch
-r-xr-xr-x   2 root     root       725372 Aug 23 11:53 waisping
drwxr-xr-x   3 root     root         1024 Aug 23 11:53 .
-r-xr-xr-x   1 root     root         3963 Aug 23 11:53 catalog
-r-xr-xr-x   1 root     root         1098 Aug 23 11:53 ws
-r-xr-xr-x   1 root     root         1191 Aug 23 11:53 check-sources
-r-xr-xr-x   1 root     root         2437 Aug 23 11:53 dictionary
-r-xr-xr-x   1 root     root          547 Aug 23 11:53 getaddrs
-r-xr-xr-x   1 root     root         3866 Aug 23 11:53 inverted_file
-r-xr-xr-x   1 root     root        22391 Aug 23 11:53 makedb
-r-xr-xr-x   1 root     root         6817 Aug 23 11:53 mkfmt
-r-xr-xr-x   1 root     root         1168 Aug 23 11:53 server_stats
-r-xr-xr-x   1 root     root          440 Aug 23 11:53 stats.awk
-r-xr-xr-x   1 root     root           48 Aug 23 11:53 wais-gif-display
-r-xr-xr-x   1 root     root           59 Aug 23 11:53 wais-html-display
-r-xr-xr-x   1 root     root           48 Aug 23 11:53 wais-jfif-display
-r-xr-xr-x   1 root     root           48 Aug 23 11:53 wais-jpeg-display
-r-xr-xr-x   1 root     root          213 Aug 23 11:53 wais-pict-display
-r-xr-xr-x   1 root     root           48 Aug 23 11:53 wais-ppm-display
-r-xr-xr-x   1 root     root           55 Aug 23 11:53 wais-tiff-display
-r-xr-xr-x   1 root     root         1825 Aug 23 11:53 waisretrieve
-r-xr-xr-x   1 root     root       751228 Aug 23 11:53 swais
```

et installer libwais.a en /usr/local/lib/freeWAIS-sf
renommer libwais.a
et /usr/local/man

2. Wais.pm 2..311

```
cd /usr/local/src/Wais-2.311
cp /usr/local/src/freeWAIS-sf-2.2.11/lib/libwais.a /usr/local/lib/libwais.a
cp /usr/local/src/freeWAIS-sf-2.2.11/lib/wais.h /usr/include/wais.h

Pour éviter toute confusion sur le .h : /usr/include/wais.h

mv ./lib/wais.h ./lib/wais.h.old
mv ./ui/wais.h ./ui/wais.h.old

modifier le Makefile.PL pour pointer sur libs et include
'LIBS'        => "-L/usr/local/lib -L$Config{ldflags} -lwais",
'INC'         => "-I/usr/include -DWAIS_USES_STDIO -I$Config{cppflags}",
perl Makefile.PL
make
make test
make install
make clean
```