

Vincent Colombel
IUT GEII de BORDEAUX 1



Mise à jour du serveur Internet et enrichissement de la base de données de SPERA.

Durée : 13/04/2003-18/06/2004

Tuteur : Patrice Kadionik



Mise à jour du serveur Internet et enrichissement de la base de données de SPERA.

Rapport de stage présenté par :
Vincent Colombel

Remerciements :

Je remercie **M. Patrice Kadionik**, Professeur au sein de l'ENSEIRB, pour l'aide et le soutien qu'il m'a apporté au cours de ce stage. Ses connaissances dans les domaines de la programmation et de l'informatique m'ont été d'un grand secours pour la mise à jour de SPERA.

Résumé :

Il y a cinq ans, des étudiants de l'ENSEIRB, aidé par les membres du radio club de l'école, ont pris l'initiative de créer un Service de Préparation à l'Examen Radioamateur : SPERA et de le mettre à la disposition de tous sur Internet.

Etant donné l'ancienne génération du serveur sur lequel il était implanté et le manque de questions dans la base de données, il m'a été confié la mission de mettre à jour le serveur "Apache" et ses différents logiciels "Java" ainsi que d'enrichir le questionnaire en questions techniques, en utilisant différents systèmes d'exploitation (Unix, Windows,...).

Mots-clefs :

Radioamateur
Serveur "Apache"
Base de données
Java
Unix

Abstract :

Five years ago, ENSEIRB's students and radioclub's members created a certicat's simulator : SPERA, and put in on internet for everybody.

Now, SPERA use an old génération serveur "Apache" and it miss technical questions. So, my mission was to update the serveur and its "Java " softwares and to enrich the data base with technical questions while using different exploitation systems (Unix, Windows, ...).

Key-Words :

radio ham
Serveur "Apache"
Data base
Java
Unix

SOMMAIRE

1 Introduction.....	7
2 Le contexte du projet.....	8
2.1 Présentation de l'école	8
3 DEROULEMENT DU PROJET	9
3.1 Le cadre du projet.....	9
3.1.1 Présentation de l'architecture générale de SPERA.....	9
3.1.2 Présentation des composants de SPERA	12
3.1.3 Présentation des systèmes d'exploitation utilisés.....	13
3.2 La mission en elle-même.....	15
3.2.1 Planning.....	15
3.2.2 Documentation et entraînement sur Unix.....	15
3.2.3 Première mise à jour	17
3.2.4 Enrichissement de la base de donnée de SPERA	19
3.2.5 Deuxième mise à jour.....	24
3.3 Les difficultés rencontrées.....	26
3.4 Les résultats obtenus.....	26
4 CONCLUSION	27
ANNEXES	
5.1 Documentation diverses.....	29
5.1.2 Unix	29
5.1.2 EMACS	33
5.1.3 VI	34
5.1.4 GNU	35
5.1.5 Tomcat	36
5.1.6 Apache	37
5.1.7 Radio amateur	37
5.2 Bibliographie et liens utiles	41
5.3 Index des illustrations	42

1 Introduction

Pour devenir radioamateur et pouvoir ainsi communiquer avec le monde entier par le biais des ondes hertziennes, il faut obtenir un certificat de radioamateur. Comme tout examen, ce certificat nécessite un entraînement mais les services de simulation de cet épreuve sont souvent payant ou peu ergonomique.

Pour palier à ce manque, des étudiants de l'ENSEIRB, aidé par les membres du radio club de l'école, ont pris l'initiative de créer un Service de Préparation à l'Examen Radioamateur : SPERA et de le mettre à la disposition de tous sur Internet.

SPERA est la copie conforme de l'examen , il met le candidat dans les mêmes conditions que lors du test réel : il se présente sous la forme d'un questionnaire à choix multiples, en temps limité, traitant différents domaines comme la réglementation, l'alphabet international, l'électronique, etc ...

Le barème est respecté : 3 points pour une bonne réponse, 0 point pour aucune réponse, -1 point pour une mauvaise réponse. A la fin de la simulation, les points sont comptés et le verdict (réussite ou échec) est prononcé.

D'un point de vue technique, SPERA est installé sur un serveur Web "Apache" et se présente sous la forme d'applets et de servlets, ce qui permet de bénéficier d'une interactivité rapide et ergonomique tout au long de l'utilisation du service.

A cause de la relative ancienneté du serveur sur lequel il était implanté et le manque de questions dans la base de données, ils se posaient deux problèmes récurrents : des utilisateurs malveillant tentaient de s'introduire illégalement dans le serveur afin de le pirater, tandis que les utilisateurs normaux étaient confrontés à des répétitions des questions techniques.

Il m'a donc été confié la mission de mettre à jour le serveur "Apache" et ses différents logiciels "Java" ainsi que d'enrichir le questionnaire en questions techniques, afin de résoudre ces problèmes.

La mise à jour devait permettre de garder les codes sources des différents programmes, tels que l'applet ou le servlet, intacts, mais aussi de renforcer la protection du serveur contre toute attaque extérieure.

Cette mise à jour ne fut pas sans poser des difficultés de compatibilité entre les logiciels ou entre les logiciels et les programmes obligeant donc à modifier en profondeur les fichiers de configurations des différents logiciels. Mais la documentation et l'aide fournie par M. Kadionik ainsi que les informations disponibles sur Internet m'ont aidé à résoudre les différents problèmes.

Les résultats obtenus, à savoir : une nette évolution dans la génération des différents composants de SPERA ainsi que le branchement du dernier serveur applicatif Java : Tomcat, ont permis de résoudre le problème de la protection du serveur tout en limitant le nombre de bugs au cours de l'utilisation, tandis qu'une augmentation de 216 % du nombre de questions à caractères techniques a, quand à elle, réduit le risque de répétition des questions.

2 Le contexte du projet

2.1 Présentation de l'école



Figure 1 L'ENSEIRB

Cette école a été créée dans les années 1920, elle s'appelait alors l'ERB : " *l'Ecole de Radiotélégraphie de Bordeaux* ". Après de multiples modifications et aménagements, elle devient l'ENSERB en 1975 : " *l'Ecole National Supérieur de Radioélectricité et d'Electronique de Bordeaux* ", ensuite, un pôle informatique puis un pôle télécommunication y sont implantés, elle devient alors l'ENSEIRB en 2000 : " *l'Ecole National Supérieur d'Electronique d'Informatique et de Radiocommunications de Bordeaux* " et y forme des ingénieurs de haute qualification.

L'ENSEIRB compte actuellement 65 enseignants, ainsi que 690 élèves dont 230 nouveaux chaque année et forme en moyenne 230 ingénieurs de toutes spécialités par an.

L'ENSEIRB est reconnue au niveau national pour être une très grande école d'ingénieur et représente la poursuite d'étude idéale pour tous les étudiants de l'IUT GEII de Bordeaux 1 désireux de continuer leur scolarité au sein d'une école de grande réputation.

3 DEROULEMENT DU PROJET

3.1 Le cadre du projet

3.1.1 Présentation de l'architecture générale de SPERA

Techniquement, l'architecture de SPERA est composée de deux parties : un côté serveur et un côté client.

→ Côté client, on trouve l'applet :

Une applet est une classe Java (*byte code*) résultant de la compilation d'un code source. Ce *byte code* est téléchargé par le navigateur du client (Internet Explorer, Mozilla, Netscape,...) quand une page HTML contient son URL (l'adresse Internet de l'applet), il est ensuite stocké sur le disque dur du client. Si le navigateur est compatible Java, c'est à dire, qu'il possède une *Java Virtual Machine* (JVM) : un décodeur de *byte code*, il traduit le *byte code* en une véritable petite application qui peut gérer un interface graphique, des interfaces sonores, des connexions,... Une fois exécutée ou le navigateur arrêté, l'applet est détruite.

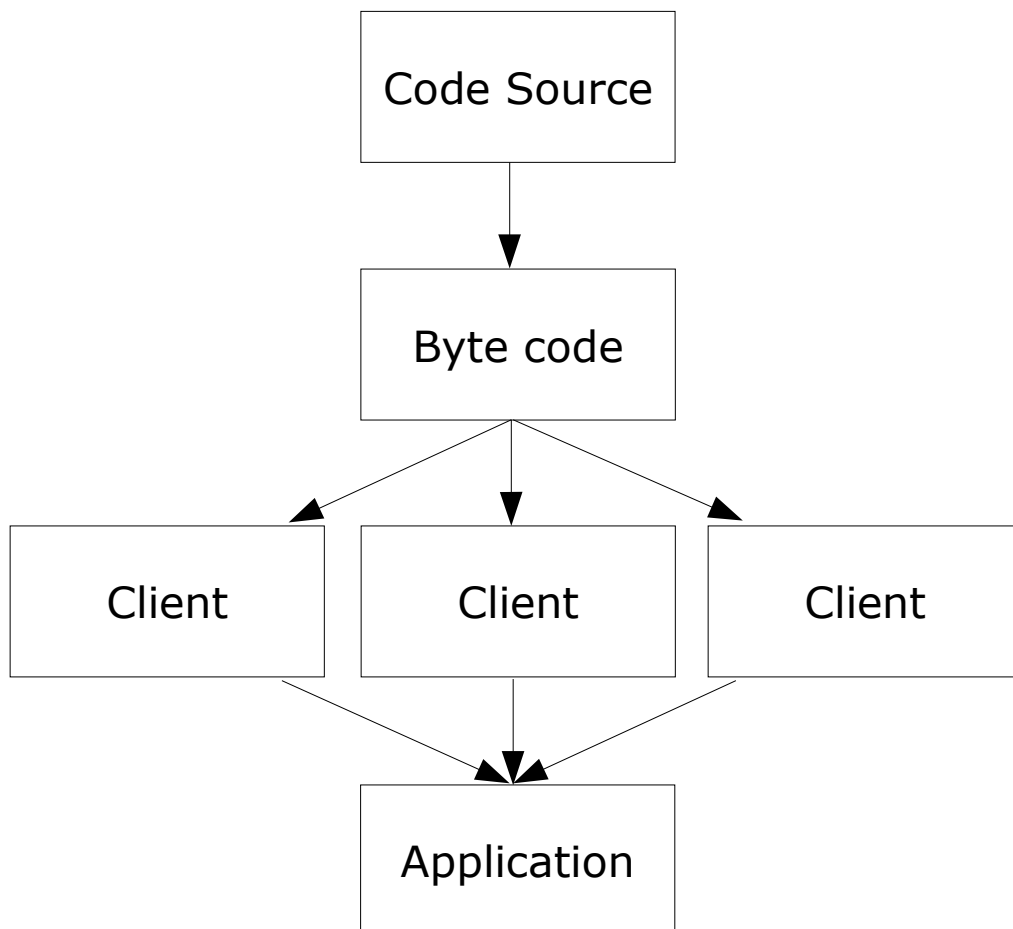


Figure 2 Evolution d'une applet

Avantages de l'usage d'applet :

- ✓ Possibilité de modifier le programme sans influencer la page HTML.
- ✓ Possibilité de développer sur Windows du code destiné à des Macintosh ou Unix et inversement.
- ✓ Rapidité de l'application une fois chargée.

Pour SPERA, l'applet a pour fonction de gérer l'interface graphique de la page Internet :

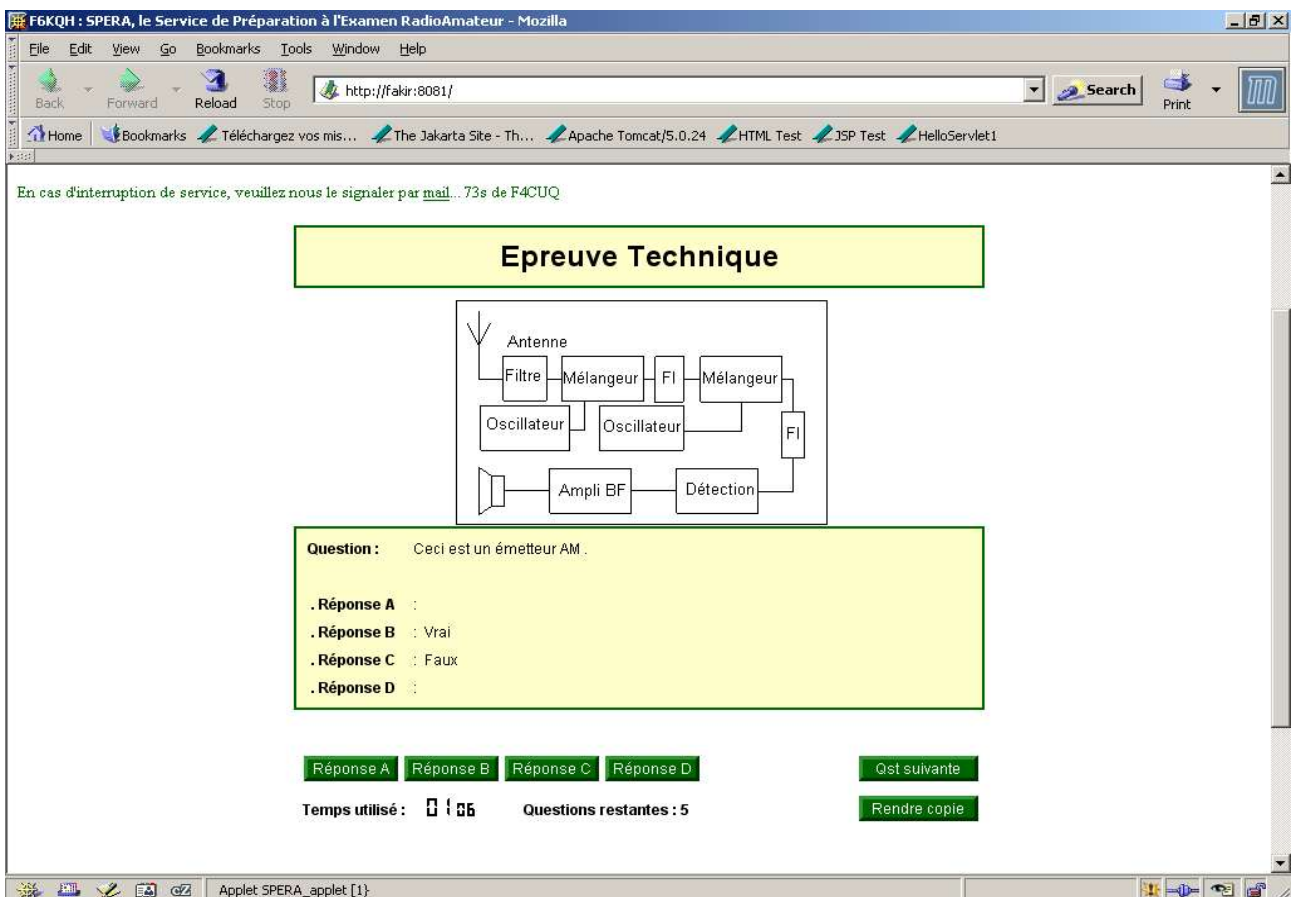


Figure 3 SPERA en fonctionnement

L'applet s'occupe :

- ✓ De la mise en page : l'organisation des différents secteurs (images, questions, boutons,...), et l'organisation de la zone des questions.
- ✓ De la gestion du temps par l'intermédiaire du compte à rebours.
- ✓ Du séquençement des évènements : choix du questionnaire, examen, calcul des points, résultat.

→ Côté serveur, on trouve la servlet :

Une servlet est une application semblable à une applet mais qui tourne sur le serveur Web. Elle réagit aux requêtes du client : elle reçoit des données entrantes ou des ordres et génère des données sortantes. Elle est souvent utilisée pour gérer des bases de données stockées sur le serveur : elle reçoit une requête du client, exécute une action dans la base de donnée et renvoie un résultat au client.

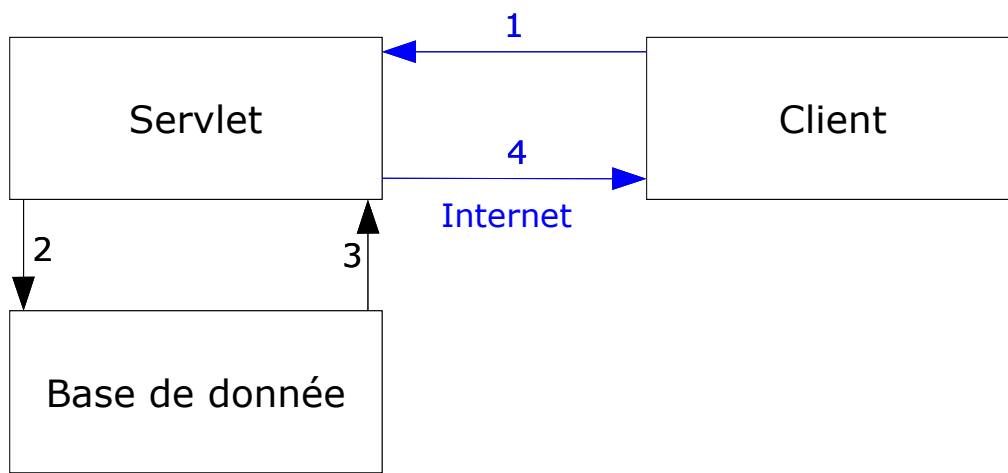


Figure 4 Fonctionnement d'une servlet

Une fois ces actions terminées, la servlet est détruite par le servlet engine (l'application qui transmet les informations du client à la servlet et inversement) afin de libérer l'espace mémoire qu'elle occupait.

Pour SPERA, la servlet a pour fonction la gestion de la base de données de question.

La servlet s'occupe :

- ✓ Du trie des questions en fonction du type d'examen.
- ✓ Du trie des schémas techniques en fonction des questions choisies.
- ✓ Du choix aléatoire de l'image en cas d'absence de schéma.

On peut schématiser SPERA comme ceci :

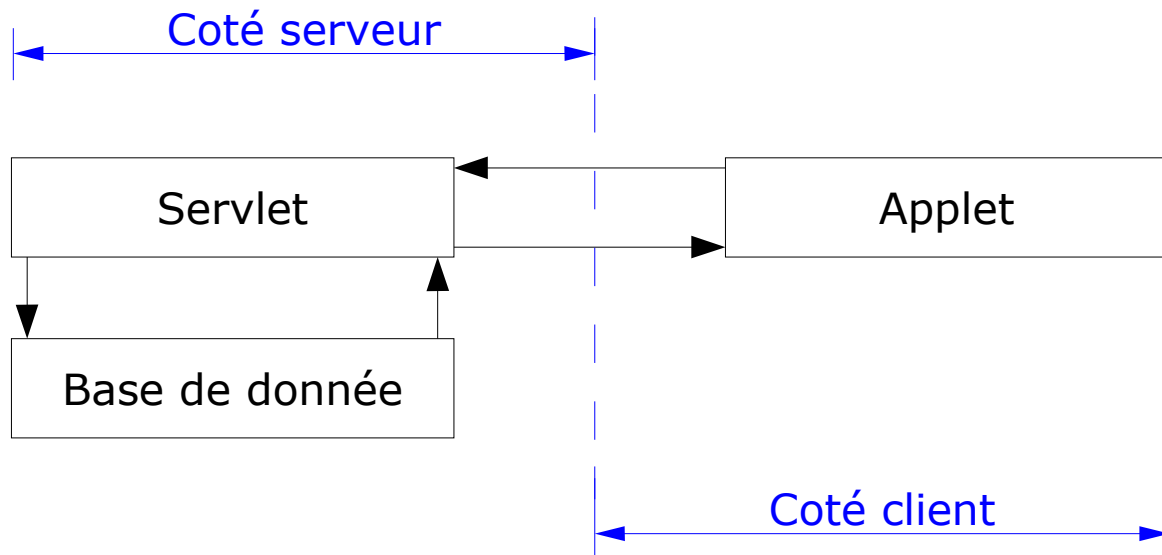


Figure 5 Schématisation de SPERA

On appelle cette architecture : une architecture 3 tiers, c'est à dire qu'elle est divisée en 3 niveaux :

- niveaux 1 : côté client (envoi de requêtes et réception de données).
- niveaux 2 : côté serveur (réception de requêtes et envoi de données).
- niveaux 3 : base de données.

3.1.2 Présentation des composants de SPERA

→ Le serveur Web Apache :

Le serveur Web qui supporte SPERA est un serveur "Apache". Ce serveur appartient à la famille des serveurs les plus utilisés au monde, en effet, près de 50 % des serveurs Web actuels sont des serveurs "Apache".

Ses principaux atouts sont :

- Un coût inexistant vu qu'il est en libre téléchargement.
- Une configuration aisée vue qu'elle est entièrement paramétrable par fichier texte.
- Une évolution permante afin de corriger toutes erreurs ou toutes failles de sécurité.
- Une maintenance facilitée par la présence de fichiers *.log* retraçant l'historique des connexions au site et des erreurs survenues au cours de son exécution.

Sa gratuité ainsi que sa fiabilité contribue certainement à sa grande popularité.

→ Les logiciels Java de Sun :

La totalité des logiciels utilisés pour développer ou tester des programmes Java

proviennent du développeur SUN dont le but est de fournir des applications de qualité identique aux produits commerciaux mais développées avec l'esprit d'ouverture du logiciel libre.

➤ **Le JDK :**

Le Java Development Kit est un ensemble d'outils servant à développer des programmes en code Java. JDK contient ces différents logiciels:

- **javac** : le compilateur qui permet de compiler le code source en *byte code*.
- **Java** : une machine virtuelle Java (JVM).
- **applet viewer**: un interpréteur d'applets.
- **jdb**: un débogueur pour rechercher les erreurs.
- **javap**: un décompilateur, pour décompiler du *bytecode* en codesource.
- **avadoc**: un générateur de documentation.
- **jar**: un compresseur de classes Java.

➤ **Le JSDK :**

Le Java Servlet Development Kit est un ensemble d'outils servant à développer des servlets. Le JSDK de Sun contient, en plus, un serveur web et un moteur de servlets permettant de les tester.

➤ **Les JSP :**

Les Java Server Pages sont des pages interactives. En effet, là où une page HTML aura continuellement le même contenu, une JSP pourra avoir un contenu différent suivant certains paramètres modifiables.

➤ **JServ :**

L'Apache JServ est un moteur de servlet très évolué spécialement dédié à l'exécution de servlet par l'intermédiaire d'un serveur Apache.

A l'heure actuelle, JServ n'évolue plus, il restera définitivement bloqué à sa version 2.0, car son successeur est arrivé : Tomcat.

➤ **Tomcat :**

Tomcat est le successeur de JServ. Il est à l'heure actuelle la référence officielle pour l'exécution de servlets ou de JSP et est donc le meilleur moteur de servlet développé, disponible sur Internet.

3.1.3 Présentation des systèmes d'exploitation utilisés

Afin de tester ou de modifier SPERA, il a été nécessaire d'utiliser différents systèmes d'exploitation.

➔ **Unix :**

C'est un système d'exploitation très prisé dans le milieu des informaticiens. En

effet, sa fiabilité et sa grande sécurité en font un logiciel très utilisé dans les universités et dans les milieux professionnels. Il a servi à exécuter toutes les modifications propres au serveur et à son environnement logiciel.

```

fakir.enseirb.fr - PuTTY
colombel:fakir% ls
/a/brahmane/home10/autre/colombel
F6KQH/      index.txt  nsmail/    packages/  pk@        tomcat/
bin/        mgc_tmp/   office97/  personnel/ tmp/        windows/
colombel:fakir% cd F6KQH/
colombel:fakir% ls
/a/brahmane/home10/autre/colombel/F6KQH
Client/     JSDK2.0/   Serveur/   jdk1.1.8/  packages/
colombel:fakir% cd Serveur/
colombel:fakir% ls
/a/brahmane/home10/autre/colombel/F6KQH/Serveur
ApacheJServ-1.1.2/  apache_1.3.29/  do_jserv_1.3.29*  scripts/
apache/            do_jsapa_1.3.29*  go-1.3.29*        servlets/
colombel:fakir% cd apache
colombel:fakir% ls
/a/brahmane/home10/autre/colombel/F6KQH/Serveur/apache
Jserv/        conf/      htdocs.org/  libexec/    proxy/
bin/          conf.org/  icons/       logs/
cgi-bin/      htdocs/    include/     man/
colombel:fakir% cd logs
colombel:fakir% ls
/a/brahmane/home10/autre/colombel/F6KQH/Serveur/apache/logs
access_log  error_log  jserv.log    mod_jserv.log
colombel:fakir%

```

Figure 6 Unix

→ **Windows :**

Logiciel phare de Microsoft, Windows est certainement le système d'exploitation le plus utilisé par les particuliers au monde. Sa simplicité d'utilisation ainsi que sa polyvalence font de lui une plate-forme populaire et appréciée de tous. Il a principalement servi à supporter les logiciels nécessaires à l'élaboration de nouvelles questions ainsi que de leurs schémas.

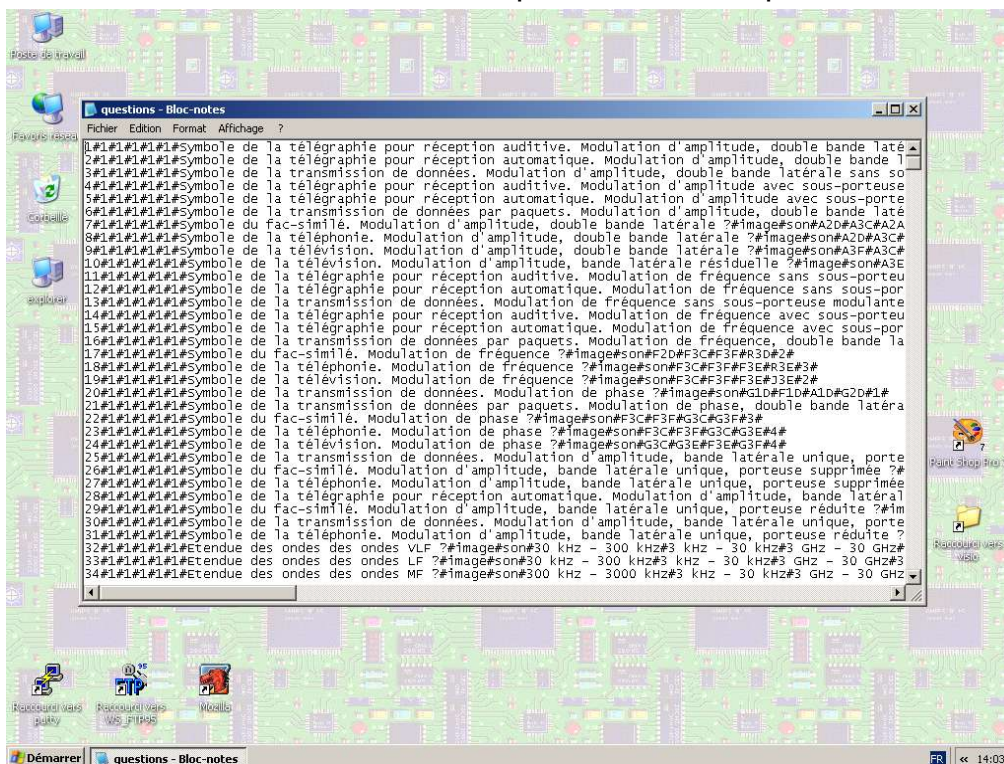


Figure 7 Windows

3.2 La mission en elle-même

3.2.1 Planning

Semaine :	Semaine 1	Semaine 2	Semaine 3	Semaine 4
Action :	Documentation et entraînement sur Unix			1 ^{ère} mise à jour

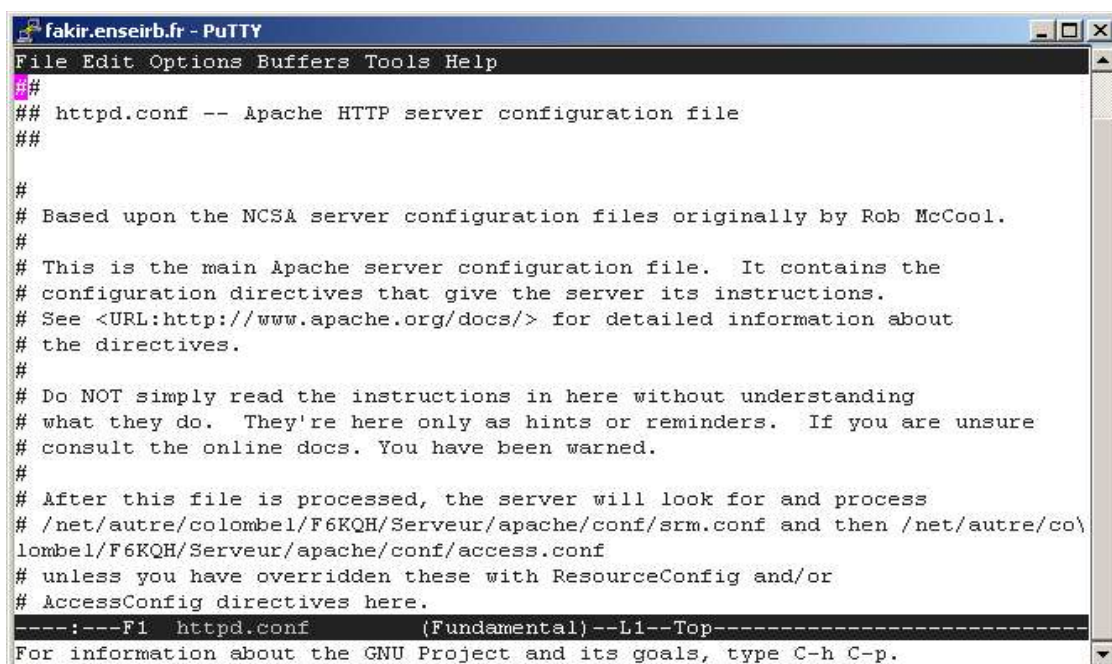
Semaine :	Semaine 5	Semaine 6	Semaine 7
Action :	Rédaction des nouvelles questions et documentation sur Tomcat		

Semaine :	Semaine 8	Semaine 9	Semaine 10
Action :	2 ^{ème} mise à jour		

3.2.2 Documentation et entraînement sur Unix

Trois semaines ont été nécessaires pour assimiler les bases de la programmation Unix et de ses principaux composants :

→ **EMACS :**

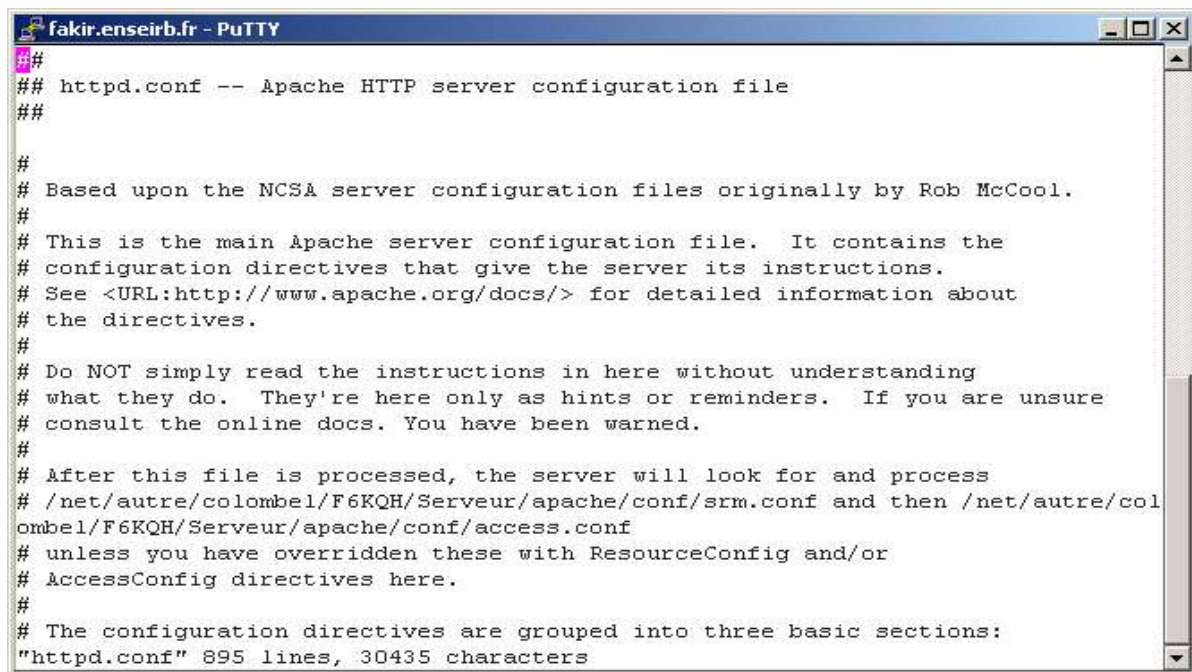


```
fakir.enseirb.fr - PuTTY
File Edit Options Buffers Tools Help
##
## httpd.conf -- Apache HTTP server configuration file
##
#
# Based upon the NCSA server configuration files originally by Rob McCool.
#
# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://www.apache.org/docs/> for detailed information about
# the directives.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs.  You have been warned.
#
# After this file is processed, the server will look for and process
# /net/autre/colombel/F6KQH/Serveur/apache/conf/srm.conf and then /net/autre/co\
lombel/F6KQH/Serveur/apache/conf/access.conf
# unless you have overridden these with ResourceConfig and/or
# AccessConfig directives here.
----:--F1 httpd.conf (Fundamental)--L1--Top-----
For information about the GNU Project and its goals, type C-h C-p.
```

Figure 8 Emacs

Emacs est un éditeur de texte puissant et peut remplir plusieurs fonctions. Son nom vient de l'acronyme de 'Editing MACroS' (éditeur de macro) et il est issu du projet GNU, c'est pourquoi Emacs est un logiciel libre. Il a été développé avec la philosophie Unix, ce qui explique pourquoi un grand nombre de fichiers de configuration ou de code source, destiné à être sous Unix, sont de simples textes bruts. En conséquence Emacs devient l'outil central pour le programmeur.

→ VI :



```
##  
## httpd.conf -- Apache HTTP server configuration file  
##  
  
#  
# Based upon the NCSA server configuration files originally by Rob McCool.  
#  
# This is the main Apache server configuration file. It contains the  
# configuration directives that give the server its instructions.  
# See <URL:http://www.apache.org/docs/> for detailed information about  
# the directives.  
#  
# Do NOT simply read the instructions in here without understanding  
# what they do. They're here only as hints or reminders. If you are unsure  
# consult the online docs. You have been warned.  
#  
# After this file is processed, the server will look for and process  
# /net/autre/colombel/F6KQH/Serveur/apache/conf/srm.conf and then /net/autre/col  
# ombel/F6KQH/Serveur/apache/conf/access.conf  
# unless you have overridden these with ResourceConfig and/or  
# AccessConfig directives here.  
#  
# The configuration directives are grouped into three basic sections:  
"httpd.conf" 895 lines, 30435 characters
```

Figure 9 VI

VI est l'éditeur de texte rivale d'EMACS. VI vient de 'Visual Interface' car il était conçu pour être un interface visuelle. D'utilisation compliquée, il est assez difficile à utiliser au début, mais il peut être très performant en utilisation normale.

De taille plus réduite qu'EMACS et de performances équivalentes, il est l'éditeur de texte le plus communément installé sous Unix.

3.2.3 Première mise à jour

Après la période d'adaptation nécessaire à la découverte et à l'utilisation d'Unix, la première mise à jour de SPERA a été effectuée.

La configuration de départ remontait à 1999, année durant laquelle a été installé SPERA.

Voici les versions des logiciels de l'époque :

- Apache Web Server 1.3.11
- Apache JServ 1.1
- Java Development Kit 1.1.5
- Java Servlet Development Kit 2.0

Dans un premier temps, nous avons récupéré les nouvelles versions de ces logiciels, puis nous les avons installées.

Tout d'abord, nous avons installé le Serveur Apache 1.3.29 accompagné de Apache JServ 1.1.2. Pour ce faire, nous avons dû modifier les scripts d'installation (programmes résumant des lignes de commandes) sous VI en leurs indiquant les nouvelles arborescences.



```
fakir.enseirb.fr - PuTTY
cd ApacheJServ-1.1.2

echo "Apache-JServ version 1.1.2 with :)"
echo "Apache version 1.3.29"

./configure --with-JSDK=/net/autre/colombel/F6KQH/JSDK2.0/lib/jsdk.jar --with-a
pache-src=/net/autre/colombel/F6KQH/Serveur/apache_1.3.29 --enable-apache-conf -
-prefix=/net/autre/colombel/F6KQH/Serveur/apache/Jserv/ --enable-compressed-jar

make
make install

cd ..

"do_jserv_1.3.29" 11 lines, 359 characters
```

Figure 10 Exemple de script

Puis, il a fallu remplacer JSDK 2.0 par JSDK 2.1 par l'intermédiaire du script `do_jserv1.3.29`.

Ensuite il a fallu remettre les fichiers de configuration générale qui se trouvaient dans le dossier `~/F6KQH/Serveur/apache.old/conf` dans le nouveau dossier `~/F6KQH/Serveur/apache/conf` et qui gèrent les accès au serveur ainsi que les propriétés de fonctionnement de celui-ci.

```
fakir.enseirb.fr - PuTTY
/a/brahmane/home10/autre/colombel/F6KQH/Serveur/apache/conf
access.conf          httpd.conf~        mime.types
access.conf.default jserv.conf         mime.types.default
example/            jserv.properties  servlets.properties*
example.properties* magic              srm.conf
httpd.conf          magic.default      srm.conf.default
colombel:fakir%
```

Figure 11 Les fichiers de configuration

Enfin, le remplacement du JDK 1.1 par le 1.2 put se faire mais fut problématique.

En effet, après avoir installé JDK 1.2 (appelé aussi JDK2), des problèmes de compatibilité (appelé aussi exception ou bug) entre JDK et la machine exécutant l'applet apparurent :

- JDK 1.1 : exception sur les machines récentes car cette version est trop vieille.
- JDK 1.2 : exception sur les machines anciennes et très différent du 1.1.

Un test exécuté avec la toute dernière version (JDK 1.4.2) montra encore des exceptions.

La solution intermédiaire qui a été trouvée fut donc d'installer la dernière version du JDK mais en technologie 1.1 : la 1.1.8. Elle représentait le meilleur compromis mais ne supportait pas JSDK 2.1. Il a donc fallu désinstaller JSDK 2.1 et réinstaller JSDK 2.0.

→ Après la 1ère mise à jour, le serveur avait donc ces logiciels:

- Apache Web Server 1.3.29
- Apache JServ 1.1.2
- Java Development Kit 1.1.8
- Java Servlet Development Kit 2.0

Un test de fonctionnement se révéla concluant : le serveur était stable, ne provoquait aucune exception chez le client, et ses fichiers *.log* ne relataient aucunes erreurs de fonctionnement.

Néanmoins, cette première mise à jour démontrait le manque de compatibilité entre les versions des logiciels Java, dépendant du fait qu'elles sont souvent très différentes les unes des autres.

Afin de moderniser et d'améliorer d'avantage SPERA, la décision fut prise d'intégrer au serveur Apache, le nouveau moteur de servlet : Tomcat, ceci allait donc faire l'objet d'une seconde mise à jour.

3.2.4 Enrichissement de la base de donnée de SPERA

→ Etude du programme de révision :

Dans un premier temps, une étude complète de la base de donnée existante fut nécessaire. Elle permit de souligner un manque important de questions dans la partie technique, ce qui à l'usage, provoquait une sorte de routine dans le questionnaire et qui n'était donc pas favorable à l'entraînement du candidat.

Dans un second temps, une étude du programme technique de l'examen réel fut utile afin de cibler les domaines sur lesquels les nouvelles questions devaient porter.

Les domaines retenus furent les suivants :

- Electronique (signaux, filtre, amplificateur, ...).
- Composants électroniques (résistance, condensateur, transistor, ...).
- Physique (propagation, conductivité, ...).
- Antenne (caractéristiques, gain, dénomination, couplage, ...).
- Schémas synoptiques d'émetteur ou de récepteur.
- Mesures (instrumentations, protocoles, ...).
- Modulation (FM, AM, BLU, ...).

Dans le même ordre d'idée, une étude d'un questionnaire proposé par une revue spécialisée mensuelle permit de juger parfaitement le niveau général des questions posées lors de l'examen.

→ Syntaxe du questionnaire :

Les questions sont stockées dans un fichier *.txt* : *questions.txt* , et peuvent être ainsi modifier à l'aide d'un simple éditeur de texte.

La syntaxe de ce fichier répond à une organisation rigoureuse, en effet, chaque question est organisée en 14 champs délimités par des '#'.

Exemple :

10294#2#0#0#1#1#5000 Hz = #image#son#50 kHz#5 MHz#0,005 MHz#0,05 MHz#3

- *10294* : numéro de la question.
- *2* : indique que c'est une question technique (1 pour la réglementation et 3 pour le code Q).
- *0* : indique que l'on ne peut pas poser cette question dans un examen de classe 1(*0* est remplacé par *1* dans le cas contraire).
- *0* : indique que l'on ne peut pas poser cette question dans un examen de classe 2(*0* est remplacé par *1* dans le cas contraire).
- *1* : indique que l'on peut poser cette question dans un examen de classe 3 (*1* est remplacé par *0* dans le cas contraire).

- 1 : indique que l'on peut poser cette question dans un examen de classe 4 (1 est remplacé par 0 dans le cas contraire).
- 5000 Hz = : texte de la question.
- image : signifie qu'il n'y a pas de schéma technique associé à cette question, une image aléatoire parmi 15 sera chargée.
- son : ce champ doit toujours contenir 'son', la fonction qui lui est associée n'est pas encore développée.
- 50 kHz : réponse A.
- 5 MHz : réponse B.
- 0,005 MHz : réponse C.
- 0,05 MHz : réponse D.
- 3 : numéro de la bonne réponse (0,005 MHz).

→ Protocole pour l'ajout de questions :

- ✓ **Etape 1** : Copie du fichier *questions.txt*, qui se trouve dans le répertoire : `~/F6KQH/Serveur/apache/htdocs/db`, sous un poste équipé de Windows.
- ✓ **Etape 2** : Edition du fichier *questions.txt* et rédaction de la question, accompagnée de ses propositions de réponses en s'appuyant sur un schéma au brouillon.

```

10656#2#0#0#1#1#F=145,500 MHz, F0l=134,800 MHz, Fi=10,7 MHz, quelle est la valeur de la fréquence image ?#
10657#2#0#0#1#1#F=145,900 MHz, F0l=135,200 MHz, Fi=10,7 MHz, quelle est la valeur de la fréquence image ?#
10658#2#0#0#1#1#F=145,100 MHz, F0l=134,400 MHz, Fi=10,7 MHz, quelle est la valeur de la fréquence image ?#
10659#2#0#0#1#1#sur 7,100 MHz, le ROS=1,5 et sur 7,050 MHz, le ROS=2, que pouvez-vous en déduire ?#Tech241
10660#2#0#0#1#1#sur 14,200 MHz, le ROS=1,7 et sur 14,350 MHz, le ROS=3, que pouvez-vous en déduire ?#Tech2
10661#2#0#0#1#1#sur 144,150 MHz, le ROS=3 et sur 144,750 MHz, le ROS=1,5, que pouvez-vous en déduire ?#Tec
10662#2#0#0#1#1#sur 7,150 MHz, le ROS=2,5 et sur 7,750 MHz, le ROS=1,5, que pouvez-vous en déduire ?#Tech2
10663#2#0#0#1#1#Cette image est issue d'un oscilloscope .#Tech242.gif#son# #vrai#Faux# #3
10664#2#0#0#1#1#Cette image est issue d'un analyseur de spectre .#Tech242.gif#son# #vrai#Faux# #2
10665#2#0#0#1#1#Cette image est issue d'un fréquencemètre .#Tech242.gif#son# #vrai#Faux# #3
10666#2#0#0#1#1#Cette image est issue d'un voltmètre .#Tech242.gif#son# #vrai#Faux# #3
10667#2#0#0#1#1#Cette image est issue d'un analyseur de spectre .#Tech243.gif#son# #vrai#Faux# #3
10668#2#0#0#1#1#Cette image est issue d'un oscilloscope .#Tech243.gif#son# #vrai#Faux# #2
10669#2#0#0#1#1#Cette image est issue d'un voltmètre .#Tech243.gif#son# #vrai#Faux# #3
10670#2#0#0#1#1#Cette image est issue d'un fréquencemètre .#Tech243.gif#son# #vrai#Faux# #3
10671#2#0#0#1#1#La diode Zener fournit une tension de référence à T .#Tech244.gif#son# #vrai#Faux# #2
10672#2#0#0#1#1#La diode Zener écrête le signal .#Tech244.gif#son# #vrai#Faux# #3
10673#2#0#0#1#1#La diode Zener filtre le signal .#Tech244.gif#son# #vrai#Faux# #3
10674#2#0#0#1#1#La diode Zener antiparasite le signal .#Tech244.gif# #vrai#Faux# #3
10675#2#0#0#1#1#La diode Zener antiparasite le signal .#Tech244.gif# #vrai#Faux# #3
10676#2#0#0#1#1#Ceci est un démodulateur .#Tech245.gif#son#FM#AM#BLU#numérique#3
10677#2#0#0#1#1#Ceci est un démodulateur FM .#Tech245.gif#son# #vrai#Faux# #3
10678#2#0#0#1#1#Ceci est un démodulateur BLU .#Tech245.gif#son# #vrai#Faux# #2
10679#2#0#0#1#1#Ceci est un démodulateur numérique .#Tech245.gif#son# #vrai#Faux# #3
10680#2#0#0#1#1#Ce pont de wheatstone est équilibré, le courant qui traverse A :#Tech246.gif#son#oscille#v
10681#2#0#0#1#1#Si la charge pure=100 ohms et que le ROS vaut 1, combien vaudra-t-il si la charge pure=20C
10682#2#0#0#1#1#Si la charge pure=50 ohms et que le ROS vaut 1, combien vaudra-t-il si la charge pure=150
10683#2#0#0#1#1#Si la charge pure=50 ohms et que le ROS vaut 1, combien vaudra-t-il si la charge pure=75 C
10684#2#0#0#1#1#Si la charge pure=150 ohms et que le ROS vaut 1, combien vaudra-t-il si la charge pure=45C
10685#2#0#0#1#1#Combien vaut le courant traversant la diode ?#Tech248.gif#son#5 mA#10 mA#20 mA#40 mA#2]

```

Figure 12 La base de données

✓ **Etape 3** : Saisie du schéma définitif sous le logiciel Visio Technical 4.1.

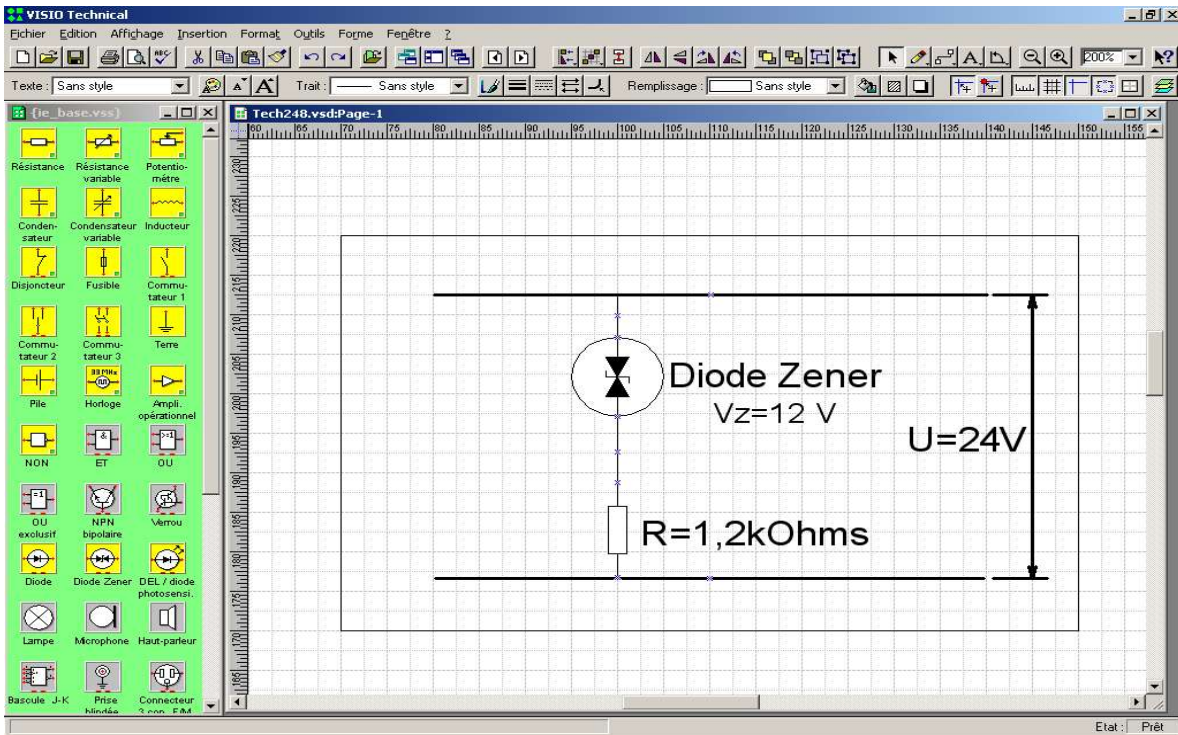


Figure 13 Saisie du schéma sous Visio

✓ **Etape 4** : Copier/Coller du schéma sous Paint Shop Pro 7.0 afin de l'enregistrer en .gif : format connue et courant sur Internet.

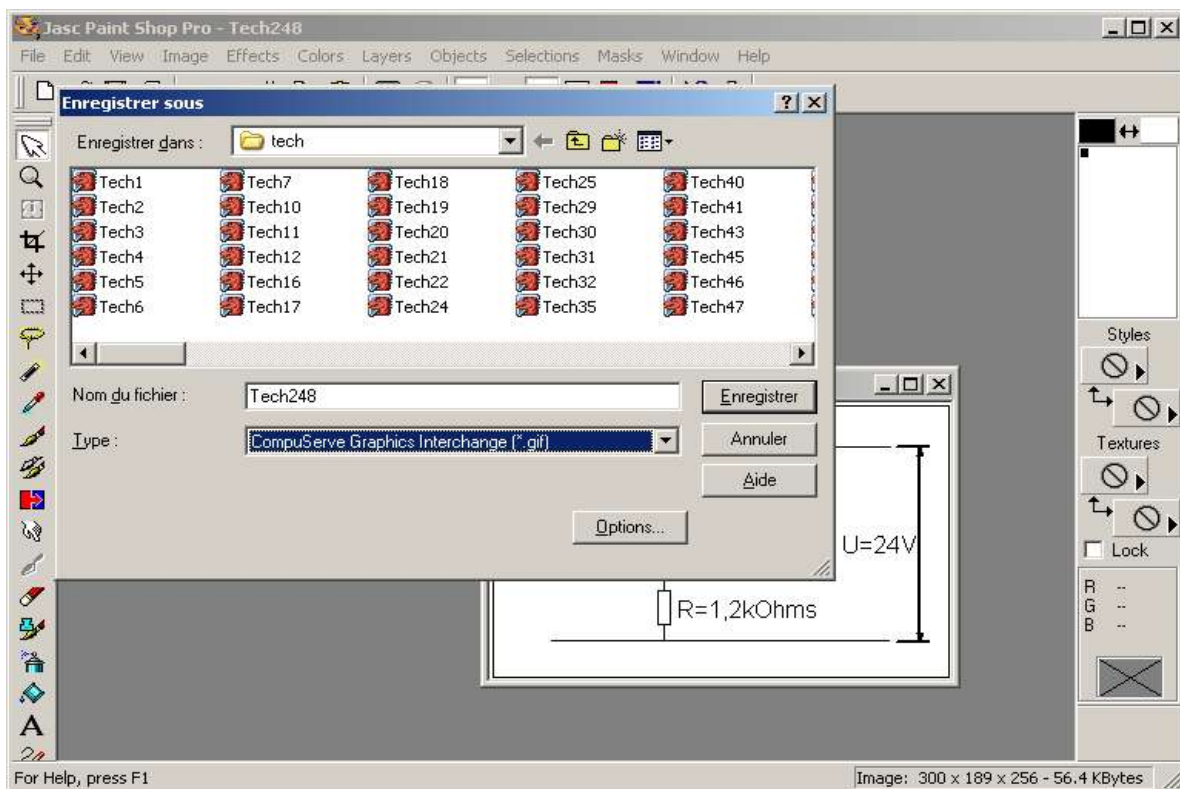


Figure 14 Enregistrement en .gif

✓ **Etape 5** : Remplacement de l'ancien fichier *questions.txt* par le nouveau et ajout du nouveau schéma au format *.gif* dans le répertoire *~/F6KQH/Serveur/apache/htdocs/Images*

✓ **Etape 6** : Vérification du bon fonctionnement de SPERA et surveillance des fichiers *.log* afin de déceler toute erreur.

→ **Cas particulier :**

Dans le cas de question de type 'Vrai ou Faux' ou 'Oui ou Non', la réponse A et la réponse D sont remplacées par des espaces vides pour n'afficher que deux réponses possibles centrées dans les propositions.

Exemple :

10609#2#0#0#1#1#Ceci est un émetteur AM .#Tech224.gif#son# #Vrai#Faux# #3

En cas d'interruption de service, veuillez nous le signaler par [mail... 73s](mailto:f6kqh@f4cuq.fr) de F4CUQ

Epreuve Technique

Question : Ceci est un émetteur AM .

.Réponse A :
.Réponse B : Vrai
.Réponse C : Faux
.Réponse D :

Réponse A Réponse B Réponse C Réponse D Qst suivante

Temps utilisé : 00:00 Questions restantes : 5 Rendre copie

Figure 15 Exemple de question

Toujours dans l'optique de limiter au maximum la mémorisation de la réponse par la reconnaissance du schéma présenté et donc de fausser le résultat du questionnaire, une solution a été trouvée : elle consiste à proposer plusieurs questions en s'inspirant d'une seule, mais en modifiant les valeurs numériques ou les propositions pour chacune.

Exemple :

A partir de cette question :

10272#2#0#0#1#1#La formule : $U = R \times I$, est-elle exacte ?#image#son# #Oui#Non# #2

En la modifiant, on peut la décliner en plusieurs variantes :

10273#2#0#0#1#1#La formule : $U = R / I$, est-elle exacte ?#image#son# #Oui#Non# #3

10274#2#0#0#1#1#La formule : $U = I / R$, est-elle exacte ?#image#son# #Oui#Non# #3

Obligeant ainsi le candidat à bien lire la question et non pas à se contenter d'un simple survole du schéma ou du texte.

Tout ceci a permis d'enrayer une certaine routine dans le questionnaire et de réduire la mémorisation des réponses systématiques.

En appliquant ces méthodes, le nombre de questions techniques est passé de 217 à 685, soit une augmentation de 468 nouvelles questions avec ou sans schéma technique.

3.2.5 Deuxième mise à jour

La 2ème mise à jour eu donc pour objet d'installer le moteur de servlet Tomcat et de l'intégrer au serveur Web Apache.



Comme dit dans les présentations, Tomcat succède à Apache JServ (qui a été stoppé à la version 2.0).

Tomcat est issu du projet Jakarta, projet qui sert à développer des logiciels pour serveurs en langage Java de qualité professionnel, mais en les mettant à la disposition de tous sur Internet.

Tomcat est donc un serveur complètement indépendant, il suffit juste de disposer une machine virtuelle Java pour le faire fonctionner.

Néanmoins, on peut l'intégrer à un serveur Apache et les faire fonctionner en complémentarité : Apache continue de gérer les pages HTML, PHP, ... mais délègue à Tomcat tout travail sur des applications Java (JSP, servlets, ...).

Pour arriver à un tel fonctionnement, il est tout d'abord nécessaire d'installer Tomcat et de le faire fonctionner en simple serveur.

→ Installation de Tomcat :

Avant d'installer Tomcat, il fut nécessaire d'installer une plate forme de développement et d'exécution Java récente. Pour ce faire, nous avons installer J2SDK 1.4.2 qui est la nouvelle dénomination des JDK récents.

Une fois J2SDK installé, c'est à dire décompacté, nous avons installé Tomcat 5.0.2. Pour que Tomcat sache où se trouve le répertoire d'installation du J2SDK, il a fallu éditer le fichier *.bash_export* qui est le fichier qui définit les variables d'environnements exportables, et faire pointer la variable d'environnement Java, sur le répertoire de J2SK 1.4.2.

Pour vérifier que la configuration est correcte, il suffit de taper (sous Unix) :
java -version pour que la version installée s'affiche :


```

fakir.enseirb.fr - PuTTY
/a/brahmane/home10/autre/colombel/pk/j2sdk1.4.2_04
COPYRIGHT          demo/
LICENSE            include/
README             jre/
README.html        lib/
THIRDPARTYLICENSEREADME.txt  man/
bin/               src.zip
colombel:fakir% java -version
java version "1.4.2_04"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_04-b05)
Java HotSpot(TM) Client VM (build 1.4.2_04-b05, mixed mode)
colombel:fakir%

```

Figure 16 Vérification de la version de Java

Une fois J2SDK 1.4.2 et Tomcat 5.0.2 correctement paramétré, nous avons vérifié le bon fonctionnement de celui-ci en allant (grâce à un navigateur internet) à l'adresse :

http://fakir.enseirb.fr:8080/

Cette adresse désigne le réseau local *fakir* de l'ENSEIRB et pointe l'adresse 8080 qui est l'adresse du serveur Tomcat en local.

En tapant cette adresse on obtient :

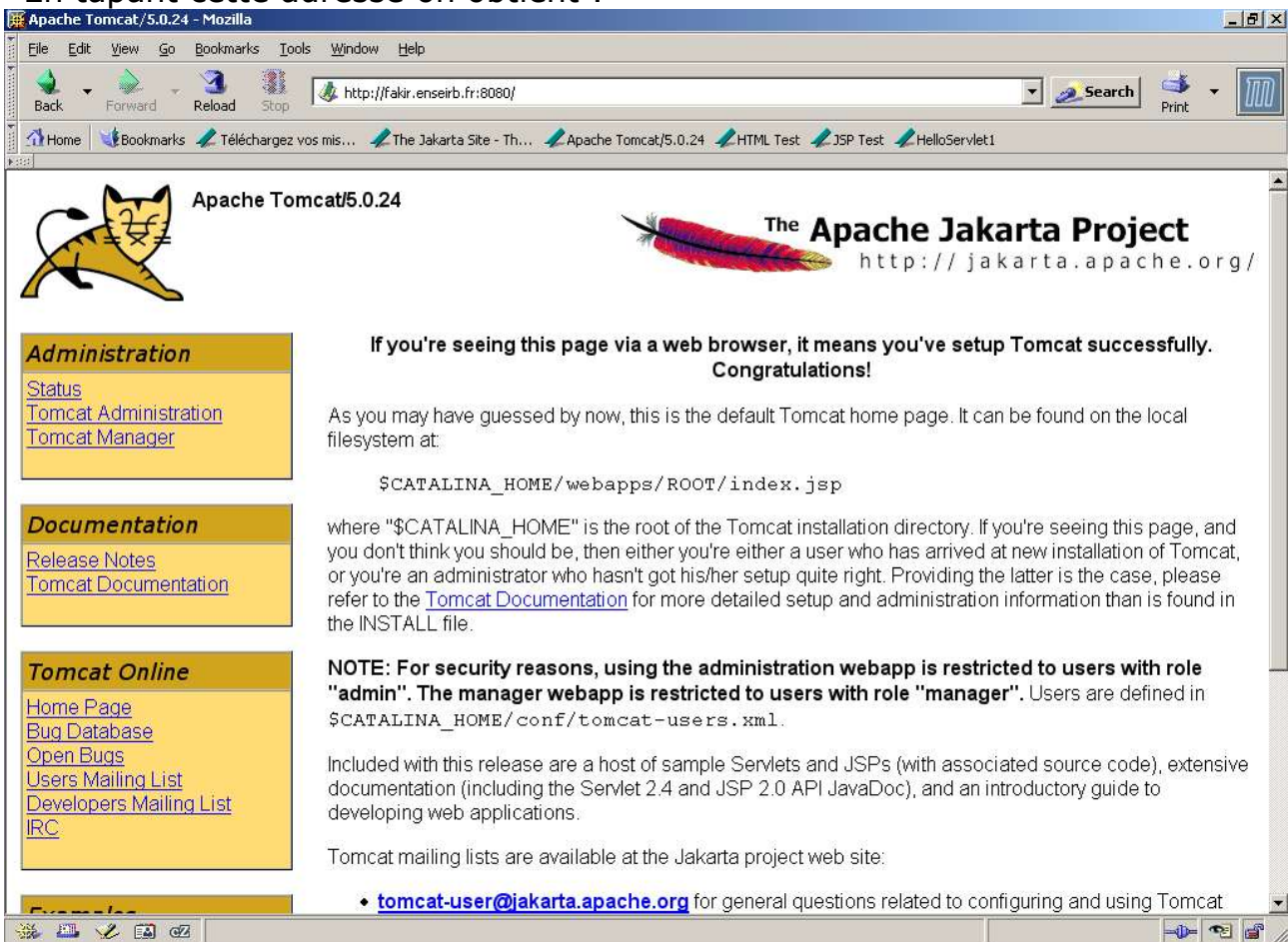


Figure 17 Page d'accueil de Tomcat

Après quelques tests de fonctionnement, l'installation s'avéra correcte et opérationnelle.

L'étape suivante fut le déplacement des applets et des servlets sur le serveur Tomcat. Une fois encore, des tests ont été effectués afin de vérifier la configuration de Tomcat et parfaire l'installation des applications importées.

La dernière étape de cette mise à jour fut la suppression de toute la partie serveur HTTP de Tomcat, vu que cette tâche était desservie uniquement par Apache, puis de connecter Tomcat à Apache par l'intermédiaire d'un module de connection. Ce module relia la partie HTTP de Apache à la partie Java de Tomcat par l'intermédiaire du port 8008. Une fois reliés, les deux logiciels pouvaient fonctionner en complémentarité.

3.3 Les difficultés rencontrées

Elles sont multiples et variées :

- Tout d'abord, la découverte et la prise en main de nouveaux logiciels et de nouveaux environnements, qui jusque là, m'étaient inconnus.
- Ensuite, les incompatibilités des différentes versions de logiciels entre eux, il a donc fallu homogénéiser leurs versions tout en restant dans des versions précédentes, afin de garantir un fonctionnement correct sur la plupart des PC.
- Puis, l'apparition de multiples erreurs de configuration rendant les systèmes inopérant jusqu'à résolution des problèmes.
- Et enfin, un problème d'hébergement du serveur car la partition du serveur sur laquelle SPERA était installé était pleine, empêchant toute installation de nouveau logiciel. M. Kadionik fut donc obligé de faire un lien pour héberger une partie de SPERA dans ses dossiers personnels.

3.4 Les résultats obtenus

En premier lieu le serveur est toujours en fonctionnement, SPERA fonctionne parfaitement et à profiter de ces nombreuses améliorations :

- Augmentation des versions des logiciels de développement et d'exploitation Java, ce qui modernise les applications.
- Modernisation d'Apache garantissant ainsi une meilleure sécurité face au piratage.
- Tomcat est installé et remplit ses fonctions : meilleure prise en charge des servlets tout en garantissant une plus grande stabilité et rapidité de l'ensemble.
- Augmentation conséquente du nombre de questions techniques : le nombre de questions techniques passent de 217 à 685 soit une augmentation de 468 nouvelles questions ou de +216 %, ce qui réduit le risque de mémorisation de la bonne réponse ainsi que l'impression de routine.

4 CONCLUSION

Ce stage m'a permis d'élargir ma palette de logiciels et de système d'exploitation utilisable : Unix, Open Office, Visio Technical, Paint Shop Pro 7.0, Emacs, Vi, ... Ce qui améliore ma polyvalence ainsi que mes compétences en Web-informatique.

Il m'a aussi permis de découvrir et de participer à des applications Internet de l'informatique et plus particulièrement à la maintenance et à la configuration de logiciels propres aux serveurs, mais aussi à mesurer la difficulté de la maintenance de systèmes informatisés notamment ceux destinés à être connectés au Web.

L'enrichissement de la base de donnée de SPERA à approfondie mes connaissances pratiques dans le domaine de la radioélectricité et des télécommunications par voies hertziennes, tout en mettant à contribution mes connaissances en électronique.

La satisfaction d'avoir élaboré quelque chose de fonctionnel et d'utile en quasi-autonomie, ainsi que les facultés d'adaptation dont j'ai dû faire preuve ont renforcé mon assurance personnel.

Au niveau professionnel, je pense pouvoir étendre mes compétences et mettre à profit mes acquis dans les domaines de la maintenance informatique et des radiocommunications au niveau de technicien supérieur.

ANNEXES

5.1 Documentations diverses

5.1.2 Unix

L'Histoire du système Unix :

Le premier système "Unix" a été mis au point par Ken Thompson dans les laboratoires Bell AT&T à Murray Hill dans le New Jersey aux Etats-Unis à partir de 1965. Le but de Ken Thompson était de mettre au point un système d'exploitation interactif simple, nommé "Multics" (Multiplexed Information and Computing System) afin de faire tourner un jeu qu'il avait créé (space travel, une simulation du système solaire).

Un consortium composé du MIT (Massachusetts Institute of Technology), de General Electric Co. et de Bell Lab se forma ainsi autour de Multics.

En avril 1969 les laboratoires AT&T décidèrent d'utiliser le système GECOS (General Electric Comprehensive Operating System) en lieu et place de Multics. Or Ken Thompson et Dennis Ritchie qui rejoignirent l'équipe eurent besoin de faire fonctionner le jeu space travel sur une plus petite machine (un DEC PDP-7, Programmed Data Processor possédant uniquement 4K de mémoire pour faire fonctionner les programmes des utilisateurs), c'est la raison pour laquelle il réécrivirent le système afin de créer une version réduite de Multics appelée UNICS (UNiplexed Information and Computing Service), rapidement contracté en Unix.

La date du 1^{er} janvier 1970 est considérée comme étant la date de naissance du système Unix, ce qui explique pourquoi toutes les horloges système des systèmes d'exploitation Unix démarrent à partir de cette époque.

Parallèlement à ces activités, D.Ritchie participa grandement à la définition du langage C (puisqu'il est considéré comme un de ses créateurs avec B.W.Kernighan), ainsi l'ensemble du système fût entièrement réécrit en langage C en 1973 et baptisé Unix Time-Sharing System (TSS).

Lorsque le système passa à la version 7 en 1979, l'évolution s'accompagna de nombreuses modifications notables telles que :

- la suppression du bridage lié à la taille des fichiers,
- une meilleure portabilité du système (fonctionnement sur de nombreuses plates-formes matérielles),
- l'ajout de nombreux utilitaires.

Un décret datant de 1956 empêchait l'entreprise ATT, dont dépendait Bell Labs, de commercialiser autre chose que des équipements téléphoniques ou télégraphiques, c'est la raison pour laquelle la décision fût prise en 1973 de distribuer les sources d'Unix dans les universités à des fins éducatives.

Dès fin 1977 des chercheurs de l'Université de Californie redéveloppèrent une version d'Unix à partir des sources fournies par AT&T afin de faire fonctionner le système sur leurs plates-formes VAX et la baptisèrent BSD pour Berkeley

Software Développement.

Ainsi deux branches de développement des sources virent le jour :

- La branche d'AT&T qui allait devenir System V de Unix System Labs (USL)
- BSD (Berkeley Software Développement) développé par l'Université de Californie

En 1977 AT&T mit les sources d'Unix à la disposition des autres entreprises, si bien qu'un grand nombre d'Unix-like furent développés :

- AIX, Unix commercial basé sur System V développé en février 1990 par IBM
- Sun Solaris, Unix commercial basé sur System V et BSD développé par SUN Microsystems
- HP-UX, Unix commercial basé sur BSD développé à partir de 1986 par Hewlett Packard
- Ultrix, Unix commercial développé par DEC
- IRIX, Unix commercial développé par SGI
- Unixware, Unix commercial développé par Novell
- Unix SCO, Unix commercial basé sur System V développé dès 1979 par Santa Cruz Operations et Hewlett Packard
- Tru64 Unix, Unix commercial développé par Compaq

En 1983 AT&T eût le droit de commercialiser son Unix, ce qui marqua l'apparition de Unix System V, la version commerciale de son système Unix. En 1985 un professeur hollandais, nommé Andrew Tannenbaum, développa un système d'exploitation minimal, baptisé Minix, afin d'enseigner la programmation système à ses étudiants.

En 1991 un étudiant finlandais, Linus Torvalds, décida de concevoir, sur le modèle Minix, un système d'exploitation capable de fonctionner sur les architectures de type 386.

Voici un schéma non exhaustif retraçant globalement l'apparition des principaux systèmes de type Unix :

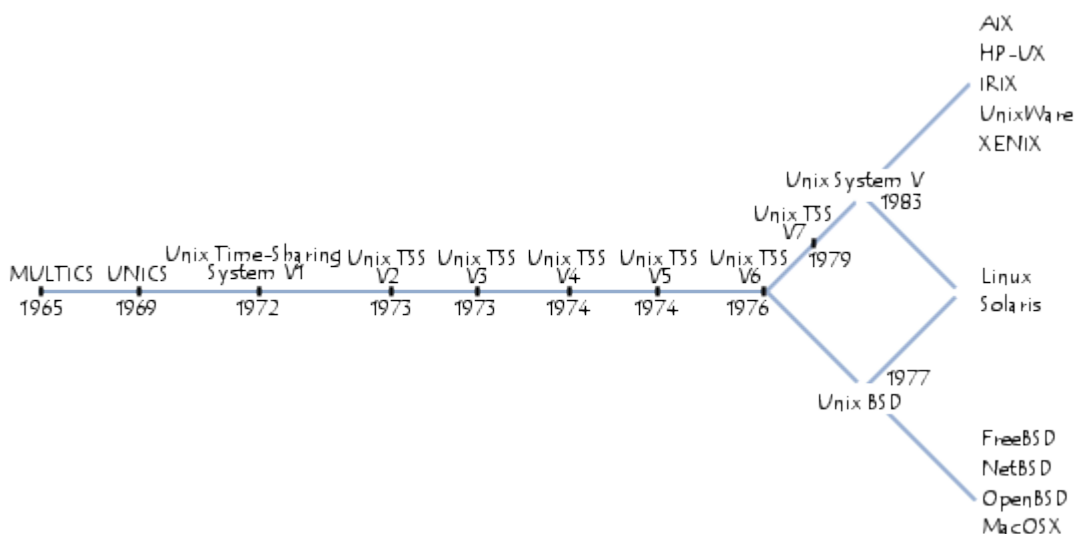


Figure 18 Evolution d'Unix

Le standard Unix :

Etant donné le grand nombre de systèmes Unix développés sur la base du System V de AT&T ou bien de BSD, la question d'un standard Unix s'est posé dès 1981 sur le groupe de discussion afin d'assurer une portabilité maximale entre les systèmes :

- en 1983, AT&T publie SVID (System V Interface Definition) décrivant le System V. Cette première définition est différente de POSIX
- en 1984 le groupe publie POSIX, une série de standards développés sous couvert de l'IEEE (Institute of Electrical and Electronics Engineers). POSIX est ainsi également connu sous le nom IEEE P1003.

à la même époque, un consortium de constructeurs (Sun, IBM, HP, DEC, AT&T, Unisys, ICL, ...) publie le standard le standard X/Open Portability Guide Issue 3 (XPG3). Ce standard s'occupe tout particulièrement des différences issues de la localisation géographique (date, alphabet, ...).

Le Système Unix :

Le système Unix est un système d'exploitation (noyau) multi-utilisateurs, multi-tâches, ce qui signifie qu'il permet à un ordinateur mono ou multi-processeurs de faire exécuter simultanément plusieurs programmes par un ou plusieurs utilisateurs. Il possède un ou plusieurs interpréteurs de commandes (shell) ainsi qu'un grand nombre de commande et de nombreux utilitaires (assembleur, compilateurs pour de nombreux langages, traitements de texte, messagerie électronique, ...). De plus il possède une grande portabilité, ce qui signifie qu'il est possible de mettre en oeuvre un système Unix sur quasiment toutes (?) les plates-formes matérielles.

De nos jours les systèmes Unix restent très présents dans les milieux professionnels et universitaires grâce à leur grande stabilité, leur niveau de sécurité élevé et le respect des grands standards, notamment en matière de réseau, dans leur implémentation.

Tableau des principales commandes Unix

Commande Unix	Description	équivalent DOS
ls	liste le contenu d'un répertoire	dir
cd	change de répertoire	cd
cd ..	répertoire parent	cd.. ou cd ..
mkdir	créé un nouveau répertoire	md
rmdir	supprime un répertoire	rd

cp	copie de fichier	copy, xcopy
mv	déplacement de fichier	move
rm	supprime le fichier	del
pwd	"print working directory" affiche le chemin complet du répertoire de travail en cours	del
passwd	change le mot de passe de l'utilisateur	
cat	commande permettant de concaténer deux fichiers	cat
more	affiche le contenu du fichier avec des pauses	more
file	affiche le type présumé du fichier spécifié	
man apropos	aide sur la commande demandée	help
lpr	imprime le fichier demandé	print
chmod	change l'attribut d'un fichier chmod XXX fichier XXX= Utilisateur Groupe Autres ou X représente un entier compris entre 1 et 7 Lecture=4, Ecriture=2, Execution=1 X=Lecture+Ecriture+Execution	attrib
chfn	change les informations personnelles vues avec finger	
chsh	change le shell : chsh user emplacement_du_shell	
clear	Efface les lignes affichées sur le terminal	cls
finger, who	liste des utilisateurs en ligne	
traceroute	trace le chemin entre la machine locale et la machine visée	tracert
ping	outil de diagnostic permettant de savoir si une machine répond sur le réseau	ping
ftp [machine] [port]	transfert de fichier entre la machine locale et la machine cible	
get	recupère un fichier	ftp
put	envoie un fichier	
quit	quitte la session FTP	
telnet [machine]	effectue un telnet	telnet
talk	permet de parler à un utilisateur connecté talk user	

	autorise ou non la commande talk	
mesg	mesg n : Empêche la réception de messages talk mesg y : Permet la réception de messages talk	
exit	déconnexion	exit

Ce document issu de CommentCaMarche.net est soumis à la licence GNU FDL. Vous pouvez copier, modifier des copies de cette page tant que cette note apparaît clairement.

© Copyright 2004 Jean-François Pillou - Hébergé par CD-Training, la formation Multimédia et Interactive par CD-Roms (CD-Training.fr) et Internet (Net-Training.fr). Le forum est hébergé en partenariat avec l'EpiTech.

Source : <http://www.commentcamarche.net/unix/unixintro.php3>

5.1.2 EMACS

Emacs est un éditeur de texte puissant et multifonction. Son nom est l'acronyme de Editing MACroS (« macros d'édition »). Mais les détracteurs de ce logiciel prétendent que Emacs est l'acronyme récuratif de Emacs makes all computers slow (« Emacs rend tous les ordinateurs lents »), à cause de sa réputation de lourdeur. Avec la puissance des ordinateurs actuels, cela n'est plus vrai. D'autres prétendent que Emacs est l'acronyme de Esc Meta Alt Control Shift à cause de certains de ses raccourcis claviers .

Emacs est un logiciel libre issu du projet GNU et qui a été développé à l'origine par Richard Stallman en collaboration avec James Gosling. Il est écrit en langage Lisp, et les fonctions d'édition de texte sont extensibles en Elisp (Emacs Lisp).

En 1991, suite à des divergences d'opinion sur le développement, un fork du développement eu lieu et donna naissance à Lucid Emacs (ensuite renommé Xemacs) qui se voulait alors offrir une version graphique et des fonctionnalités étendues. Cette version n'est pas approuvée par le projet GNU qui continue à développer GNU Emacs pour lequel bon nombres des fonctionnalités mises en avant dans XEmacs existent désormais.

Emacs est issu de la philosophie de conception Unix, qui comprend beaucoup de fichiers de configuration sous forme de texte brut, tout comme pour le code source des logiciels, ce qui explique que l'éditeur soit un outil central. Son rival en la matière est Vi, ce qui provoque des querelles légendaires (voir Troll (Internet)) entre les partisans de l'un et les promoteurs de l'autre.

Entre autres fonctionnalités d'Emacs :

- modes majeurs pour la plupart des langages (indentation, coloration

- syntaxique...), du texte brut au SGML, en passant par le Langage de programmation Perl, le HTML...
- Système de documentation intégré très performant avec navigation hypertexte ;
 - Hautement extensible : parmi les extensions on trouve la lecture du courrier électronique (vm,gnus...), un navigateur Web, des clients FTP ...

Source : <http://fr.wikipedia.org/wiki/Emacs>

Le contenu est disponible selon les termes de la licence GNU Free Documentation License. 'Wikipedia, l'encyclopédie libre'

5.1.3 VI

Vi est un éditeur de texte orienté écran écrit par Bill Joy sur une des premières versions de BSD. Le nom provient de Visual Interface (interface visuelle), car il était au départ conçu comme une interface visuelle contrairement à l'éditeur de ligne ex (ex est toujours disponible dans vi en appuyant sur : en mode commande).

Vi tend à frustrer extrêmement les nouveaux utilisateurs, car c'est un éditeur modal. En mode commande, la plupart des touches deviennent des commandes d'éditions, par exemple pour déplacer le curseur ou effacer du texte. Dans le mode édition, l'utilisateur peut taper du texte comme un éditeur classique, la touche échappement permettant de sortir de ce mode. Les premières versions de vi ne donnaient aucune indication sur le mode courant: typiquement, les utilisateurs appuyaient toujours sur échappement pour s'assurer d'être en mode commande. Les versions plus récentes de l'éditeur affichent le mode courant dans une ligne de statut en bas de l'écran. Un avantage de taper une commande dans le mode de commande est que plusieurs opérations d'édition peuvent être effectuées en ligne avec des combinaisons de touches très simples, sans avoir à tenir les touches alt, ctrl ou autres. Pour les utilisateurs expérimentés, ceci peut mener à une meilleure performance.

Vi est devenu l'éditeur standard de Unix et l'éditeur favori des hackers jusqu'à l'arrivée d'Emacs en 1984. En 2002, vi ou l'un de ses clones peut être trouvé dans presque toutes les installations de Unix (le Single Unix Specification inclut vi, donc tout système s'y conformant l'intègre).

Vi est encore très utilisé par les utilisateurs des variantes Unix (environ la moitié des répondants à un sondage Usenet en 1991 le préfère aux autres éditeurs). Il démarre plus vite que l'éditeur Emacs plus lourd et utilise moins de mémoire; conséquemment, mêmes quelques fans d'Emacs l'utilisent comme éditeur de courrier électronique ou pour de petites éditions. Sur les disquettes de sauvetage (rescue disk), c'est très souvent l'éditeur par défaut parce qu'il est assez compact pour tenir sur une disquette.

Vi et Emacs sont les deux opposants d'une guerre d'éditeurs.

Exemples de commandes

- a : insérer du texte après le curseur
- A : insérer du texte à la fin de la ligne courante
- c : permet de modifier un certain nombre de caractères, en fonction de la sous-commande (par exemple cl permet de changer le caractère courant, cw la fin du mot courant)
- i : insérer du texte au niveau du curseur
- O : insérer une nouvelle ligne au-dessus de la ligne courante
- o : insérer une nouvelle ligne en-dessous de la ligne courante
- :q : quitter vi
- s : permet de remplacer une chaîne de caractère par une autre. (par exemple on utilisera s/avant/après/g pour remplacer le mot "avant" par le mot "après" sur la ligne courante.
- w : positionne le curseur sur le premier caractère du mot suivant
- :w : enregistrer le fichier courant
- x : supprimer le caractère se trouvant sous le curseur
- / : permet de rechercher une chaîne de caractères

source : <http://fr.wikipedia.org/wiki/Vi>

Le contenu est disponible selon les termes de la licence GNU Free Documentation License. 'Wikipedia, l'encyclopédie libre '

5.1.4 GNU

Le projet GNU a été lancé par Richard Stallman en 1984 afin de créer un système d'exploitation libre et complet et, d'après ses mots, "ramener l'esprit de coopération qui prévalait dans la communauté informatique dans les jours anciens". Stallman demande que GNU soit prononcé gnou avec un g audible pour éviter l'horrible confusion avec le mot anglais new. Le symbole de GNU est un gnou, animal qui se prononce de la même manière que GNU en anglais.

GNU est un acronyme récursif pour "GNU's Not Unix" (littéralement, GNU n'est pas Unix). Cependant, étant donné qu'Unix était déjà largement répandu, et que son architecture globale avait fait techniquement ses preuves, le système GNU fut conçu pour être compatible avec lui.

L'association du projet GNU avec le noyau Linux s'est produite naturellement lorsque le noyau fut disponible et parfaitement fonctionnel dans l'environnement GNU. Le projet GNU avait prévu le développement du noyau Hurd pour compléter le système, mais au début des années 90, Hurd ne fonctionnait pas et son développement rencontrait d'importantes difficultés.

L'arrivée du noyau Linux a donc rendu disponible les logiciels du projet GNU sur les ordinateurs à base de microprocesseurs Intel et a permis une large diffusion du nouveau système d'exploitation GNU/Linux.

En 1985, Stallman a créé la FSF, Free Software Foundation, pour fournir la

logistique, le support légal et financier au projet GNU. La FSF a également financé des développeurs pour contribuer au projet GNU, une portion substantielle du projet ayant été (et étant encore) réalisée par des volontaires. Alors que la réputation de GNU grandissait, des entreprises intéressées ont commencé soit à contribuer au développement, soit à revendre les logiciels du projet GNU, et à offrir du support technique. La principale de ces sociétés étant Cygnus Solutions, faisant maintenant partie de Red Hat.

A partir de 1990, le système GNU disposait de son propre éditeur de texte (Emacs), d'un compilateur très performant (GCC), et de la plupart des bibliothèques système d'une distribution Unix standard. Le principal composant encore manquant étant le noyau.

Source : <http://fr.wikipedia.org/wiki/GNU>

Le contenu est disponible selon les termes de la licence GNU Free Documentation License. 'Wikipedia, l'encyclopédie libre '

5.1.5 Tomcat

Tomcat est un serveur d'applications *Java* capable de gérer des applications Web. Une application Web est une application accessible avec un navigateur Web et pouvant effectuer des opérations très complexes : agence de voyage en ligne, banque en ligne,... *Java* est une des technologies permettant de réaliser efficacement ce genre d'applications grâce à une combinaison de servlets (programmes *Java* exécutés par le serveur) et de JSP (*Java Server Page*) qui sont des pages HTML enrichies de morceaux de programme *Java*. Un serveur d'applications *Java* est donc un moteur capable d'exécuter des servlets et des JSP, ce que ne peut pas faire un simple serveur HTTP comme *Apache*. C'est pourquoi une architecture souvent rencontrée consiste en un serveur HTTP comme *Apache* pour "servir" les documents statiques (fichiers HTML, images) couplé à un serveur d'applications comme Tomcat pour les servlets et JSP. L'un des intérêts de cette technologie est son très haut niveau de spécification : on parle par exemple de la norme J2EE édictée par Sun et à laquelle se conforment les éditeurs de logiciel désireux de fournir un serveur d'applications. Tomcat constitue l'implémentation de référence de cette norme. L'intérêt de cette norme est qu'une application J2EE pourra être déployée tout aussi bien sur Tomcat que sur WebSphere (IBM) ou WebLogic (BEA). Il est possible sous Tomcat de déployer à chaud, c'est à dire sans devoir arrêter le serveur d'applications lui-même.

Source : <http://www.alcove.com/fr/alcove/newsletter/index>

5.1.6 Apache

Fin 1994 : le serveur web le plus utilisé dans le monde est celui du National Institute for Supercomputing Applications (NCSA httpd 1.3 -Université de l'Illinois). Le développement s'arrête suite au départ du développeur Rob McCool.

Début 1995 : naissance de la première équipe Apache (11 personnes) qui va coordonner la mise en commun de centaines de corrections et d'améliorations faite au serveur par des développeurs indépendants pour leurs propres besoins. Première version publique de Apache (0.6.2)

Début 1996 : suite à la sortie de la version 1.0 et moins d'un an après la création du groupe, Apache dépassa son "papa" en terme d'utilisation et devient le serveur Web le plus utilisé dans le monde.

Aujourd'hui : Apache est toujours le serveur Web le plus utilisé. La "Apache Software Foundation" compte plusieurs dizaine de membres, des centaines de contributeurs et développe plusieurs projets liés à la diffusion d'information sur Internet (serveur java, produits XML, langages de script...)

Source : <http://tecfa.unige.ch/guides/tie/pdf/files/inst-serverset.pdf>

5.1.7 Radio amateur

La 1ère liaison :

En avril 1922, le Français Léon Deloy ("F8AB") réussit plusieurs liaisons bilatérales avec l'Angleterre et le 28-11-1923 la première liaison avec l'Amérique.

Quelques radioamateurs célèbres :

Le Roi Hussein de Jordanie JY1A, le Roi Carlos d'Espagne, le sénateur Goldwater (USA) et Owen Garriott W5LFL : 1er astronaute radioamateur en 1983.

Les services amateurs :

L'instruction individuelle, l'intercommunication et les études techniques, sont des activités effectuées par des amateurs autorisés s'intéressant à la technique de la radioélectricité à titre personnel et sans intérêt pécuniaire, sous la tutelle de l'Autorité de régulation des télécommunications (ART) depuis le 1-1-1997.

L'autorisation d'émettre :

Elle est délivrée après :

- 1) Obtention du certificat d'opérateur radiotéléphoniste ou radiotéléphoniste radiotélégraphiste après examen.
- 2) Agrément de candidature par différents ministères. (Intérieur, Défense P&T)
- 3) Constatation de la conformité de l'installation aux conditions techniques prévues.

Les 3 groupes de licences sont :

Classe 1 HF/VHF/UHF, 250 watts.

Classe 2 : VHF/UHF, 120 watts.

Classe 3 (novices) : VHF 10 watts.

Redevance. Licence annuelle 45€.

Les fréquences :

Il y a 36 bandes autorisées en métropole, de 137,5 KHz à 250 000 MHz. Jusqu'à 10 000 MHz, les communications sont soit à très longue ou soit à très courte distance.

Au delà de 10 000 Mhz, on entre dans un domaine plutôt expérimental.

Les puissances autorisées :

Elles varient selon les groupes et les classes d'émission. Globalement, elles vont de 10 W à 500 W.

L'indicatifs :

Il est constitué d'un préfixe d'appartenance géographique (France continentale : F), suivi d'un suffixe (chiffre de 0 à 9), puis d'un groupe de 2 ou 3 lettres caractérisant le titulaire. Exemples : F6XZB, F9IV, ...

Les infrastructures :

Dans de nombreux pays, il y a :

Des réseaux de relais émetteurs récepteurs VHF (en France, plus de 100 relais VHF, bande 144 MHz, et 40 relais UHF 430 MHz).

Des balises émettant un signal fixe pour l'étude de la propagation et un réseau de transmission de données 90 BBS (Boîtes aux lettres informatiques.)

Dans l'espace, on trouve une dizaine satellites à l'usage exclusif des radioamateurs.

En France, par exemple :

Satellite Race (Radio Club de l'espace) avec le soutien du CNES.

Satellite SARA (satellite amateur de radioastronomie) lancé fin Juin 1991 par Esie-éspace (radio club de l'Esiee, Ecole Supérieure d'ingénieurs en électrotechnique et électronique) pour étudier les rayonnements radioélectriques de Jupiter.

Les modes de trafic :

Ils sont divers et variés : télégraphie, téléphonie, télex, télévision à balayage lent ou de type classique, fac-similé (télécopie), transmission de données par voies radio (packet-radio).

Le choix des fréquences et des modes se combinent suivant le type de liaison voulu, elle peut être en vue directe, à grande distance, intercontinentales ou bien par réflexion sur la lune ou essais de météorites ou même par satellites radioamateurs.

Le trafic radioamateur :

Le radioamateur doit au moins communiquer : son indicatif officiel, son prénom, sa localité, ses conditions de réception et d'émission et son matériel

utilisé (antenne, émetteur....).

Les informations doivent être uniquement de type technico-scientifique (ni messages privés, ni politique, ni religion ou langage cru...).

Il se peut qu'ils utilisent des termes abrégés directement issus du code télégraphique, comme par exemple : QRA : domicile, QRG : fréquence, QRM : interférence, QSB : affaiblissement passager, QSL : carte accusé de réception qui confirme la liaison radio, OM : Old Man : opérateur, YL : Young Lady : opératrice.

L'écoute des radioamateurs :

Elle était très facile, avec des récepteurs ordinaires, dans les bandes 7, 14, 21 et 28 MHz, à l'époque où les radioamateurs utilisaient la modulation d'amplitude (AM).

Actuellement le trafic s'effectue presque entièrement en modulation à bande latérale unique (BLU, SSB pour les Anglo-Saxons), une petite portion de bande 28 MHz est utilisée par certains radioamateurs (en particulier Américains) en modulation de fréquence (FM).

Le décryptage de la BLU (bande latérale inférieure ou supérieure) ne peut s'effectuer que par un récepteur de "trafic" équipé d'un oscillateur local créant une porteuse artificielle dont est dépourvue l'émission BLU.

Le nombre de personnes à l'écoute :

Il y a environ 6000 SWL (Short Wave Listener) qui pratiquent l'écoute avec virtuosité, ils éditent souvent une carte QSL pour confirmer l'écoute des stations entendues. Et ils peuvent même obtenir un code de reconnaissance du type F-12345 (100F pour 5 ans), en s'adressant au CNERA.

Les autres activités :

Le réseau des radioamateurs peut aussi être un réseaux d'urgence : exercices ou alertes réelles dans le cadre du service de protection civile et du plan ORSEC. Ils peuvent être amenés à remplacer des réseaux de télécommunications détruits par une catastrophe naturelle ou par une guerre, en vue d'aider la population.

Les compétitions :

Le but d'un concours est d'effectuer un maximum de liaisons en un temps déterminé (exemple : Championnat de France).

Les diplômes :

Ils sont obtenus pour des liaisons réussies avec des correspondants déterminés dans des conditions précises.

Les Radio Clubs :

Ils assurent l'échange des connaissances théoriques et pratiques et préparent à l'examen pour l'obtention de la licence.

Nombre de radioamateurs :

Monde : environ 4 200 000

Japon => 1 200 000

USA => 600 000
Allemagne => 65 000
France :
France en 1925 => 355
France en 1939 => 650
France en 1950 => 3000
France en 1960 => 4000
France en 1970 => 6000
France en 1980 => 11000
France en 1991 => 17000
France en 1995 => 18500
dont classe 1 : 11200 classe 2 :6800
Radio clubs => 500

Source : <http://www.urc.asso.fr/def.htm>

5.2 Bibliographie et liens utiles

"Unix pour l' impatient 2ème édition"

de Paul W. Abrahams et Bruce R. Larson aux éditions VUIBERT.

Ouvrage très complet sur Unix, regroupant toutes les fonctionnalités de ce système d'exploitation et de ses composants.

<http://jakarta.apache.org/tomcat/index.html>

Site du projet Jakarta sur Apache Tomcat (en anglais).

<http://java.sun.com>

Site de Sun Microsystems : site général de tous les logiciels Java (en anglais).

<http://www.coreservlets.com>

Site regroupant des tutoriaux et des exercices d'entraînement sur Java et Tomcat (en anglais).

<http://www.developpez.com>

Site du Club des Développeurs, regroupant toutes sortes d'informations sur les différents langages de programmation.

<http://www.apachefrance.com>

Le site francophone sur Apache et les technologies Internet.

<http://fr.wikipedia.org/wiki/Accueil>

Le site de l'encyclopédie libre.

<http://www.radioamateur.org>

Portail francophone du radioamateurisme.

<http://perso.wanadoo.fr/f6crp/elec/index.htm>

Cours et traités de radioélectricité et d'électronique pour le radioamateur.

Et bien sûr :

<http://www.enseirb.fr/f6kqh/present.html>

Le site de F6KQH : le radioclub de l'ENSEIRB et de SPERA.

5.3 Index des illustrations

L'ENSEIRB.....	8
Elaboration d'une applet.....	9
SPERA en fonctionnement.....	10
Fonctionnement d'une servlet.....	11
Schématisation de SPERA.....	12
Unix.....	14
Windows.....	14
Emacs.....	15
VI.....	16
Exemple de script.....	17
Les fichiers de configuration.....	18
La base de données.....	20
Saisie du schéma sous Visio.....	21
Enregistrement en .gif.....	21
Exemple de question.....	22
Vérification de la version de Java.....	24
Page d'accueil de Tomcat.....	25
Evolution d'Unix.....	30