Le Hublot

Journal des tuteurs — Numéro 5 — Mars 2000

Pour toutes questions, idées d'articles, etc, n'hésitez pas à nous écrire : `tuteurs@clipper`. *Vous pouvez aussi mettre un mot dans le casier d'Émilia Robin. Certaines pages de notre site sont indiquées dans les articles : on suppose que toutes les adresses indiquées commencent par* `http://www.eleves.ens.fr:8080/tuteurs/`. *Certains mots sont suivis d'un astérisque*; ils renvoient au lexique en fin de numéro. Vous pouvez retrouver les anciens numéros du* Hublot *sur notre site :* `http://www.eleves.ens.fr:8080/tuteurs/hublot.html`

Pratique

Typographie : les citations

Le mois dernier, on a étudié l'espacement à laisser autour des ponctuations. Ce mois-ci, on va aborder la façon de faire les citations. Comme d'habitude, ces informations proviennent du *Lexique des règles typographiques en usage à l'Imprimerie nationale*, Imprimerie nationale, 1990.

Il existe des usages en matière de citations, mais aucune règle précise, associant une façon de procéder à un type de citations. On peut en effet citer du texte en le mettant entre guillemets, en le détachant du reste du texte, ou en le mettant en italique.

Tout dépend de l'équilibre général de la page, de la présence ou non de citations dans la citation elle-même (citation de second niveau), de la présence d'italique dans la citation, etc. Comme toujours, le plus important est d'adopter une présentation uniformisée dans le document.

— **Citations ordinaires** — Une citation courte apparaît le plus souvent entre guillemets[1]. La référence est donnée après la citation, entre parenthèses, en dehors des guillemets.

Ainsi, dans le dernier *BOcal*, il est écrit qu'« une dizaine de tickets de cocktail » récompenseront les meilleurs déguisements (*BOcal*, n° 327).

— **Longues citations** — Les passages longs sont composés en romain d'un corps inférieur à celui du texte, sans guillemets, et sont isolés du texte par des blancs :

> Dans la plaine rase, sous la nuit sans étoiles, d'une obscurité et d'une épaisseur d'encre, un homme suivait seul la grande route de Marchiennes à Montsou, dix kilomètres de pavé coupant tout droit, à travers les champs de betteraves. Devant lui, il ne voyait même pas le sol noir, et il n'avait la sensation de l'immense horizon plat que par les souffles du vent de mars, des rafales larges comme sur une mer, glacées d'avoir balayé des lieues de marais et de terres nues. Aucune ombre d'arbre ne tachait le ciel, le pavé se déroulait avec la rectitude d'une jetée, au milieu de l'embrun aveuglant des ténèbres. (Émile ZOLA, *Germinal*.)

Les vers sont cités de la même façon : pas de guillemets, caractères plus petits, blancs avant et après la citation.

— **Lettres** — Reproduits *in extenso*, les textes épistolaires sont composés le plus souvent en italique ; on rencontre parfois aussi des lettres citées en romain d'un corps (taille) inférieur. Ils sont séparés du texte par des blancs.

— **Langues étrangères** — Les citations en latin ou en langue étrangère sont citées en italique. *A fortiori*, l'italique s'emploie *ad libitum* pour toutes les locutions latines, *id est* non francisées.

— **Citations de second niveau** — Le *Lexique* ne dit rien de bien clair sur ce sujet... Le cas ne se pose évidemment que pour les citations guillemetées ; dans le cas de l'italique ou d'un texte détaché, on met des guillemets normaux.

Dans le cas des citations entre guillemets, il est interdit de mettre des guillemets à l'envers, »comme ça« , pour figurer une citation dans la citation. On tolère des

[1] Les lecteurs fidèles du *Hublot* le savent désormais par cœur ; toutefois, rappelons encore que les guillemets français sont des guillemets « carrés ». Les guillemets “anglais” ou "américains" n'ont pas à figurer dans un document francophone.

guillemets anglais, “comme ceci”, dans ce cas précis. Il semble que l'usage est de mettre des guillemets normaux : « Il dit « J'arrive » et ouvrit la porte ». Si les deux niveaux de guillemets se referment à la fin de la citation, on n'en referme qu'un seul : « En arrivant, il déclara : « Je suis là ».

Questions-Réponses

Comment fait-on pour rediriger son courrier électronique vers une autre adresse ? Cette opération s'appelle « forwarder » son courrier. Pour cela, il suffit de créer chez soi un fichier appelé `.forward`[2] qui contient la nouvelle adresse. Par exemple :

```
corvette-~ $ cat .forward
toto_g@mit.edu
```

C'est tout ! Avec ça, le courrier sera automatiquement renvoyé à l'adresse `toto_g@mit.edu`.

Commande du mois

La commande `finger` vous donne des informations sur les utilisateurs du réseau. Elle peut aussi s'utiliser avec un nom d'ordinateur, mais on n'en parlera pas ici.

Suivi d'un nom de login, `finger` vous donne des informations sur la personne correspondante :

```
corvette-~ $ finger tuteurs
Login name: tuteurs        In real life: Comptes tuteurs
Directory: /users/staffs/tuteurs
Last login Thu Feb 24 15:46 on pts/12 from jonque
New mail received Mon Mar 6 17:14:25 2000;
unread since Sat Mar 4 11:31:38 2000
No Plan.
```

Vous avez donc le nom de login, le nom complet, le répertoire de cette personne (ce qui vous donne sa promotion et sa branche), et des renseignements techniques (dernier login sur la machine où vous vous trouvez, dernière fois que le courrier a été lu).

Attention avec la question du courrier : s'il est redirigé ailleurs, la personne peut avoir lu son courrier très récemment sans que cela apparaisse... Ainsi, un biologiste dont le courrier est redirigé sur `horus` peut donner l'impression de n'avoir pas lu son courrier depuis six mois ; c'est vrai *sur clipper*, la machine que vous interrogez, mais pas sur `horus`.

D'autre part, certains comptes sont un peu spéciaux, parce qu'une copie du courrier reçu est envoyé à un certain nombre de personnes. C'est le cas pour les tuteurs ; aucun tuteur ne se logue `tuteurs` pour lire le courrier, car il le reçoit directement. Le courrier est donc lu, mais ça n'apparaît pas sur le `finger`.

Vous voyez enfin l'indication `No Plan`. Cela signifie que cet utilisateur n'a pas de fichier `.plan` sur son compte ; ce fichier sert à personnaliser les renseignements donnés par `finger` (coordonnées, etc). Attention, ne mettez pas d'accent dans ce fichier. On peut aussi créer un fichier `.project`. Celui-ci ne contient qu'une seule ligne, par exemple pour une devise. Là encore, il ne faut pas mettre d'accents.

`finger` s'utilise aussi avec des noms ou des prénoms. Pour savoir le nom de famille de toutes les Nathalie qui ont un compte, tapez `finger Nathalie`.

Unix pas à pas

Le cours Unix se divise en trois parties : « Concept », « Commande », « Usage ». Ce mois-ci, on va expliquer ce que signifie « multitâches ». La partie « Commande » explique comment on obtient des renseignements sur les processus ; la partie « Usage » indique comment gérer, en pratique, les processus que l'on lance.

Concept : « multitâche »

Deux personnes qui se promènent en bavardant sont en train de faire plusieurs choses à la fois : leur attention est tournée principalement vers la conversation, mais aussi sur les expressions de l'interlocuteur, sur l'heure qui tourne, etc. De façon presque automatique, elles évitent les obstacles courants (marche, racine d'arbre...). Quant à la marche elle-même (lever le pied, lancer la jambe, déporter le poids d'une jambe à l'autre...), c'est un geste purement mécanique.

Un ordinateur est conçu globalement sur le même principe : plusieurs programmes s'effectuent simultanément, à des niveaux différents : sur votre ordinateur, vous pouvez lancer à la fois Netscape, un logiciel de courrier, un éditeur de textes, une compilation... Tout cela est analogue au niveau « conscient » de nos deux personnages.

Mais il se passe autre chose, sur le modèle automatique ou végétatif d'un être humain : afficher des fenêtres, réagir quand on déplace la souris, maintenir l'horloge à l'heure, vérifier si un nouveau courrier est arrivé, répondre si quelqu'un veut se connecter...

Chacune de ces opérations s'appelle un processus : un processus est une tâche donnée, individualisée, identifiée, qui appartient à un utilisateur bien précis et qui doit accéder à des données bien précises. Le `pine` que vous avez lancé n'interfère pas avec le rapport que, d'autre part, vous êtes en train d'écrire. Ou encore, s'il

[2] Ce n'est pas la première fois qu'on vous parle de fichiers dont le nom commence par un point. Ce sont des fichiers « cachés », qui n'apparaissent pas naturellement quand on tape `ls`. Pour les voir, il faut taper `ls -a` (*all*). Ce sont des fichiers de configuration, qui personnalisent un certain nombre de programmes. Ils se trouvent dans ~ (votre répertoire d'accueil). Surtout ne les effacez pas !

```
 PID USERNAME PRI NICE  SIZE   RES STATE  TIME  WCPU    CPU  COMMAND
13102 robin    27    0 1376K 1304K cpu    0:00 1.15%  0.92%  top.5.5.sun4u
11894 robin    33    0 8120K 6160K sleep  0:35 0.63%  0.66%  Xsun
12491 tuteurs  33    0 1592K 1360K sleep  0:04 0.64%  0.50%  vim-5.4
11895 robin    33    0 2288K 1720K sleep  0:01 0.06%  0.12%  fvwm2
(...)
```

Dans cet exemple, c'est la commande `top` elle-même, de numéro 13102, qui consomme le plus de ressources (0,92%). L'état (`STATE`) indique qu'elle est en cours d'exécution.

Viennent ensuite le serveur X (pour l'interface graphique), l'éditeur `vim` avec lequel je tape le *Hublot* au nom de `tuteurs`, et le gestionnaire de fenêtres `fvwm2`. Leur état dit qu'ils « dorment », c'est-à-dire qu'ils sont en attente.

FIG. 1: La commande `top`

arrive que l'horloge ne se lance pas, cela n'abîme pas votre boîte aux lettres.

Unix est un système multitâche et multi-utilisateurs[3], ce qui signifie que plusieurs tâches différentes et appartenant à des personnes différentes, peuvent cohabiter. C'est possible parce que le noyau* veille : si une tâche tente de manipuler les données d'une autre tâche (ce que Windows appelle « effectuer une opération non conforme »), la contrevenante se fait promptement éliminer. Sous Unix, cela ne plante pas la machine.

Commandes : `ps`, `top` et priorités

— `ps` — La commande `ps` affiche la liste des processus lancés à votre nom sur votre machine, sous la forme d'un tableau.

```
corvette-~ $ ps
  PID TT       S  TIME COMMAND
29503 pts/6    S  0:01 /usr/local/util/bin/zsh
29523 pts/6    S  0:14 xdvi hublot5.dvi
29601 pts/6    S  0:04 vim5.new hublot5.tex
29752 pts/7    S  0:01 /usr/local/util/bin/zsh
```

Décrivons les principaux champs de ce tableau : le `PID` est le numéro du processus (*process identifier*). La colonne `TT` (*teletype*) indique à quel terminal est rattaché le procesus. La colonne `S` indique l'état du processus : ici c'est `S` (*sleeping*), car les processus attendent que quelque chose se passe. La dernière colonne indique la commande qui a lancé le processus. Ici, il y a un éditeur de textes (`vim`), un DVI, et deux shells*.

— `top` — `top` affiche un petit tableau remis à jour régulièrement, qui montre les processus qui consomme le plus de ressources ; le plus gourmand figure en haut du tableau (voir l'exemple de la figure 1). Sont indiqués entre autres le numéro du processus (`PID`), le login de son propriétaire (`username`), les ressources consommées (`CPU`), et enfin le nom de la commande qui a lancé le processus.

— `nice` — Deux colonnes sont intéressantes dans ce tableau : `PRI` (priorité) et `NICE`. La priorité est l'importance donnée au processus dans la répartition du temps du processeur[4].

C'est un aspect essentiel du partage des ressources des ordinateurs lorsque l'on est dans un environnement multi-utilisateurs : un calcul lancé en tâche de fond sur un ordinateur ne doit pas monopoliser le processeur au point d'empêcher l'utilisation de la machine par quelqu'un d'autre. Quand on lance une tâche de longue haleine, comme un calcul, on dispose de la commande `nice`, pour fixer la priorité du processus. Plus le *nice* est élevé, moins le processus sera prioritaire.

— `kill` — La commande `kill` sert à envoyer un signal à un processus. Un de ses usages consiste à ordonner au processus de prendre fin. Mais ce serait une très mauvaise idée d'utiliser systématiquement `kill` pour quitter un programme ! Cette commande sert quand plus rien d'autre n'est possible.

Usage : gérer les tâches

Un programme lancé dans un terminal peut être en mode texte, comme `pine`, auquel cas il s'affiche dans la fenêtre où il a été lancé. Il peut aussi être en mode graphique, comme un jeu, Netscape, un DVI, et se lancer dans une nouvelle fenêtre. Dans ce cas, il se passe quelque chose comme ceci quand vous essayez de taper des commandes :

```
corvette-~ $ netscape www.bnf.fr
ls
toto
```

Votre Netscape s'est lancé, et il immobilise la fenêtre dans laquelle vous l'avez lancé ; vous ne pouvez rien taper.

Dans ce cas, tapez `^Z`. Cette commande suspend le processus qui est en cours d'exécution (ici, Netscape). Vous récupérez la main, c'est-à-dire que vous pouvez de nouveau taper des commandes. En l'occurence, tapez immédiatement `bg` (*background*), pour mettre votre

[3] Voir le numéro 1 du *Hublot*.

[4] Pour pinailler : le `nice` est statique, fixé par le propriétaire. La priorité est dynamique et calculée par le noyau en fonction des processus lancés et de leurs nices respectifs.

Netscape en arrière-plan (car pour le moment, il est toujours suspendu, et ne fonctionne pas). Ça donne, au total :

```
corvette-~ $ netscape www.bnf.fr
^Z
zsh: suspended  netscape www.bnf.fr
corvette-~ $ bg
[1]  + continued  netscape www.bnf.fr
```

Le plus simple est encore d'utiliser l'esperluette (&), qui met le programme lancé en arrière-plan :

```
corvette-~ $ netscape www.bnf.fr &
[1] 11321
```

Le premier numéro indiqué est le *job ID* (numéro de tâche), le second est le PID, c'est-à-dire le numéro du processus. Vous pouvez ainsi lancer plusieurs processus qui tourneront en tâche de fond. La commande `jobs` vous permet de savoir ce qui tourne en tâche de fond dans un terminal. Pour mettre au premier plan un processus donné, vous avez la commande `fg` (*foreground*), en indiquant, le cas échéant, le numéro de tâche, précédé d'un pour-cent :

```
corvette-~ $ jobs
[1]  - running    xdvi hublot5.dvi
[2]  + running    netscape www.bnf.fr
corvette-~ $ fg %2
[2]  - running    netscape www.bnf.fr
```

Internet

Dans cette section, vous trouverez la suite du cours de Thomas sur les réseaux; ce mois-ci, il explique comme est organisé Internet. Dans la section Netscape, on va parler de la navigation sur le Web.

Cours : qu'est-ce qu'un réseau ?

Dans les numéros précédents, nous avons vu comment deux ordinateurs peuvent s'échanger des données, puis les principes de l'organisation des couches basses d'un réseau plus étendu : couche physique, transport de paquets ou par circuit (« couches basses » : voir niveau*). Dans ce numéro, nous abordons le routage : suivre de bout en bout un paquet de données de la machine origine à la machine destination.

5- Internet dans tout ça

On a vu, dans la section précédente, comment relier ensemble deux stations pour qu'elles puissent s'échanger des données, pas forcément de façon fiable, et sans garantie de débit. Ces méthodes sont locales, et il convenait de passer à un réseau global. Ceci a été effectué grâce au réseau Internet, dérivé de l'Arpanet au cours des années 1970 (Arpanet était le réseau des militaires américains).

Le principe est le suivant : quand une station veut envoyer un message à une consœur, elle commence par examiner ses branchements, pour voir si la destinatrice ne serait pas, par hasard, accessible directement. Dans ce cas, elle lui envoie le message par le moyen physique présent. Dans le cas contraire, elle envoie le message à une station dont elle sait qu'elle est plus qualifiée qu'elle pour résoudre ce problème. La station qualifiée est nommée *routeur*, ou aussi *passerelle.*

Pour savoir qui est contactable et comment, chaque station est munie d'une adresse de 4 octets (chaque octet contenant un nombre entre 0 et 255). Ainsi, la station `galion`, en salle S, est dotée de l'adresse 129.199.129.10. Ces adresses sont mondiales, et toutes les adresses commençant par 129.199 sont réservées à l'ENS.

— **Un exemple à l'École** — Prenons l'exemple de `galion`, tentant d'envoyer un message (sous la forme d'un paquet IP, comme Internet Protocol) à `aviso`, en salle Infirmatique.

`galion`, d'adresse 129.199.129.10, sait qu'elle est reliée directement à toutes les stations dont l'adresse commence par 129.199.129. Or, `aviso` a l'adresse 129.199.128.1 ; `galion`, constatant cela, décide de transmettre le paquet à sa passerelle, à savoir `clipper` (129.199.129.1, contactable par un lien ethernet direct depuis `galion`). Puis `galion` se lave les mains de ce qui se passe ensuite, ce n'est plus son affaire.

`clipper` ne peut pas non plus contacter `aviso` directement, mais il peut parler sur un deuxième lien depuis son deuxième visage, `clipper-gw` (129.199.1.22). Sur ce lien, il peut contacter `finn` (129.199.1.128), qui est responsable des adresses en 129.199.128. `clipper` transmet donc le paquet à `finn`, et se désintéresse lui aussi de la question.

`finn` possède aussi deux visages, le second étant `finn128` (129.199.128.254), relié directement aux stations de l'Infirmatique. `finn` peut donc communiquer directement avec `aviso`, et lui envoie le paquet.

— **Routeurs et réseau** — Donc, pour que tout se passe bien, il suffit que chaque station sache reconnaître les adresses contactables directement, et une passerelle pour les autres cas. Les routeurs, eux, doivent avoir une notion locale de la hiérarchie (`clipper` doit connaître `finn`, mais ce que `finn` doit faire pour contacter `aviso` ne le regarde pas).

Il est même possible de reconstruire ces informations à la volée : `clipper` peut tout envoyer sur la machine par défaut (`renater`), qui lui signalera à chaque fois qu'il existe une route plus directe ne passant pas par lui ; `clipper` s'en rappellera pendant quelques minutes. Ce mécanisme, dit de routage dynamique, est un peu délicat à mettre en place, aussi on s'en sert avec parcimonie (il est aisé d'obtenir, à la suite d'un malentendu, une partie de ping-pong, où deux stations considèrent, pour un paquet donné, que l'autre station est la passerelle à utiliser).

Et voilà, ceci est Internet : des stations qui s'échangent des paquets (d'une taille maximale de 65 536 octets, mais souvent plus petits, de l'ordre de 1 500 octets). Normalement, un paquet n'a pas à effectuer plus de 30 sauts pour faire le trajet d'une station à une autre. Les paquets peuvent être fractionnés et recombinés au gré des routeurs, afin de s'adapter aux spécificités locales de la liaison.

Le chemin entre deux stations n'est pas forcément unique ; ceci permet une tolérance aux pannes ou une adaptation aux embouteillages. Notamment, les communications à grande échelle sont alors résistantes aux attaques nucléaires (c'est ce qui plaisait aux militaires américains). Une conséquence de ce fait est que deux paquets successifs ne suivent pas forcément le même chemin ; ils peuvent notamment arriver dans le désordre, et certains peuvent être dupliqués (quand une passerelle cherche à savoir, via un protocole approprié, si un paquet est arrivé, et, ne voyant rien venir, en émet un autre, alors que le premier était simplement parti par un chemin détourné).

▷ Thomas Pornin

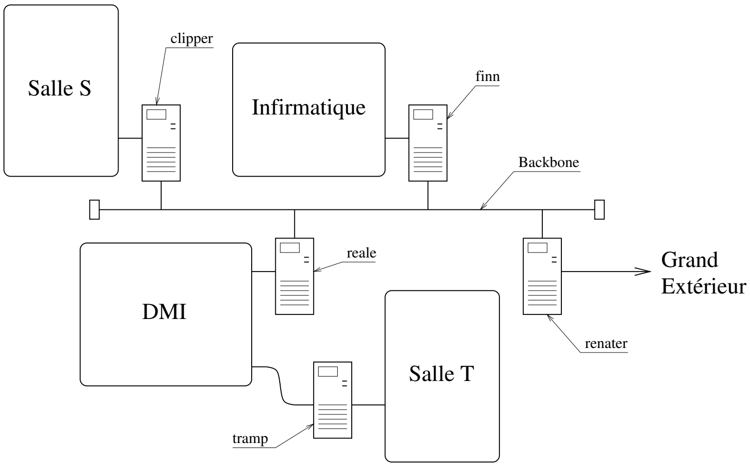


Fig. 2: Une partie du routage à l'ENS

Utiliser Netscape

Le mois dernier, on a vu comment lancer et quitter proprement Netscape. Ce mois-ci, on va envisager la navigation sur le Web. Comme d'habitude, j'utilise la version 3 de Netscape.

— **Le Web et l'hypertexte** — Vous savez que le principe du Web est de mettre en relation des informations dispersées à travers le monde : par exemple, on crée un lien entre une page Web située en France vers une page située au Brésil, et suivre le lien vous emmène automatiquement sur la page brésilienne. C'est ce que l'on appelle l'hypertexte, et c'est ce que veut dire le `http` placé au début des adresses Web : *HyperText Transfer Protocol*.

— **Principe du Web** — Sur le Web, une adresse s'appelle une URL (*Uniform Resource Locator*). Prenons pour exemple l'adresse du serveur des élèves de l'ENS : `http://www.eleves.ens.fr:8080/`. Le `http` au début indique le protocole à utiliser, et signale qu'il s'agit d'une page Web (cependant, toutes les pages Web ne sont pas servies par HTTP). Le `.fr` est un nom de domaine, qui indique ici que la page se trouve en droit en France (mais il se pourrait que le serveur, en tant que machine physique, matérielle, soit aux États-Unis). `www.eleves.ens` désigne en réalité une machine qui héberge le site des élèves de l'institution ENS. Le `8080` désigne un numéro de port.

Enfin, le `/` (*slash*) qui termine l'adresse indique qu'on se rend au point de départ de l'arborescence du site, en pratique la page d'accueil.

Tout ceci indique la machine à contacter pour rejoindre la page Web. Ensuite, il peut y avoir d'autres informations quelle page précisément on demande à voir. Par exemple, `http://www.eleves.ens.fr:8080/COF/CineClub/` correspond la page d'accueil du Ciné-Club de l'ENS, elle-même un sous-ensemble logique de `COF/`.

Quand vous sélectionnez un lien, votre ordinateur envoie une requête sur le serveur désigné par l'URL. Ce serveur renvoie la page demandée par votre navigateur (« client » du serveur).

L'ensemble de ces données circulent sur des liens physiques, les câbles, qui ont un débit maximal. La réponse d'un serveur physiquement très éloigné mettra un peu plus de temps à arriver ; une grosse quantité de données mettra plus de temps à arriver qu'une petite quantité ; enfin, à certaines heures, le réseau est encombré par l'ensemble des données transférées par tous les utilisateurs connectés à ce moment-là. C'est ce qui explique que parfois, une page soit longue à charger. Vous pouvez toujours interrompre le transfert en cliquant sur le bouton « `Stop` ».

— **Noms de domaines** — Il y a un certain nombre de noms de domaine. Une partie indique un pays, dont dépend en droit le serveur : `.fr` pour les serveurs français, `.es` pour l'Espagne, `.ru` pour la Russie, `.jp` pour le Japon, etc.

Il en existe d'autres, comme `.edu` pour les universités américaines, `.org` pour les institutions à but non lucratif, `.gov` pour le gouvernement américain. `.int` est utilisé en particulier par les organisations établies par traités internationaux, par exemple l'Union européenne.

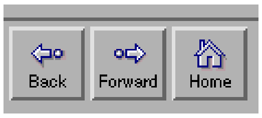
De nouveaux noms de domaines sont apparus avec l'engouement pour le Web. Par exemple, `.com` désignait à l'origine des sites d'entreprises, et `.net` des fournisseurs d'accès au réseau. Mais AOL, fournisseur d'accès, a comme adresse `aol.com` (pour des raisons historiques, certes). D'autre part, des entreprises peuvent avoir un site avec un suffixe `.net`, à cause de leur fournisseur d'accès.

Un autre exemple : le suffixe `.tv` est théoriquement réservé aux îles Tuvalu ; mais le Web est quelque chose de rentable, et elles ont vendu quelques noms de domaines à des chaînes de télévision qui voulaient une adresse en `.tv`...

— **Aller quelque part** — Que faire quand vous voulez aller à une adresse précise, par exemple `http://www.unesco.org/` ? Cliquez dans le cadre rose « `Location:` », effacez l'adresse de la page sur laquelle vous vous trouvez, et écrivez à la place l'adresse souhaitée.

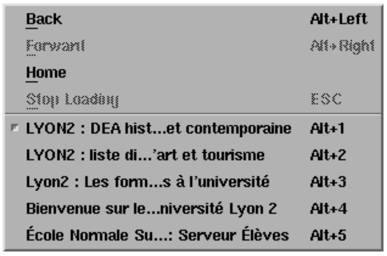
Vous pouvez aussi sélectionner `Open Location` dans le menu `File` (ou taper `Alt-L`), et taper votre adresse dans le petit cadre qui s'affiche.

— **Pages déjà visitées** — Netscape offre deux possibilités pour revenir sur des pages déjà visitées. Il s'agit d'abord des trois boutons `Back`, `Forward` et `Home`.



`Back` vous ramène en arrière, vers la dernière page que vous avez visitée ; `Forward` vous ramène vers les pages visitées en dernier. Enfin, `Home` vous replace sur votre page d'acceuil par défaut, qui est la page du serveur des élèves de l'ENS.

Vous disposez aussi du menu `Go`. Celui-ci reprend les fonctions `Back`, `Forward` et `Home`, en indiquant les raccourcis claviers correspondants.



Il propose ensuite la liste des pages visitées ; les pages vues récemment sont en haut de la liste, tandis que la toute première page visitée se trouve tout en bas. Un petit carré indique sur laquelle de ces pages on se trouve. Il suffit de cliquer sur le nom d'une page pour y revenir immédiatemment.

Les CD-ROM du Centre d'Études Anciennes

— **Disques disponibles** — Le Centre d'Études Anciennes dispose de trois CD-ROM.

Le **TLG** (*Thesaurus Linguae Graecae*) comporte les textes de la littérature grecque depuis Homère jusqu'en 600 ap. J.-C., plus des textes littéraires et techniques, ainsi que des recueils de scholies de l'an 600 à 1453, soit 3 366 auteurs et 10 823 œuvres.

Le **PHI n° 5.3**, du Packard Humanities Institute, embrasse la littérature latine depuis l'époque archaïque jusqu'aux environs de 200 ap. J.-C. Il comporte 362 références d'auteurs, dont les grands textes classiques, mais présente des lacunes pour la période postclassique. Quiconque travaille sur la poésie latine aura intérêt à consulter plutôt le CD-ROM **Poesis** (de l'éditeur italien Zanichelli) à la section latine de l'Institut de Recherche et d'Histoire des Textes. Outre les textes littéraires latins, Le PHI comprend la *Bible juive* (en hébreu), la *Septante* et le *Nouveau Testament* grec (en caractères grecs), la *Vulgate* (en latin), la Bible anglaise dans deux traductions, celle du roi Jacques et la version standard révisée, et le *Nouveau Testament* copte (en caractères coptes).

Le **PHI n° 7** renferme divers *corpus* documentaires en caractères grecs et latins. Il s'agit des inscriptions de l'université Cornell, ainsi que des *papyri*, *ostraca* et tablettes du fonds documentaire Duke (186 références). Les inscriptions couvrent la période grecque, la période romaine, l'Antiquité tardive jusqu'au VIIIe siècle et la période byzantine jusqu'en 1453. On y trouve de plus, en caractères coptes, la *Bible sahidique* et le fonds Nag Hammadi (64 références d'apocryphes bibliques).

— **Utilisation** — Ces disques sont consultables avec deux logiciels d'interrogation : **Pandora** (une pile HyperCard conçue aux États-Unis) et **SNS Greek & Latin** (une application programmée à la Scuola Normale Superiore de Pise), qui tournent sur le PowerMac 8500 de la cellule informatique du CEA. Les néophytes préfèrent Pandora pour la simplicité de son interface. Mais SNS est incontournable quand il faut définir des requêtes très précises, car la syntaxe de son langage d'interrogation est héritée du `grep` en usage dans le monde Unix, et quand il faut extraire des œuvres complètes (au format RTF) pour les récupérer dans un traitement de texte, par exemple si l'on travaille à une édition critique.

Chaque année, en novembre, des stages d'initiation aux CD-ROM sont proposés aux élèves antiquisants par D. Béguin, l'un des deux caïmans d'informatique littéraire. On y montre que toute problématique peut se reformuler sous forme d'une recherche de mots ou de parties de mot dans un corpus d'œuvres librement défini par l'utilisateur. Les réponses sont fournies sous forme de listes d'occurrences où les mots recherchés sont replacés dans un contexte dont la longueur est, elle aussi, définie par l'utilisateur. Les CD-ROM font office d'index et de concordances, avec une rapidité et une exhaustivité que ne sauraient égaler leurs prédécesseurs sur support papier.

`http://www.tlg.uci.edu/` : la page officielle du TLG à l'Université de Californie.

▷ Daniel Béguin

Lexique

— **Niveau** — Est bas niveau ce qui est proche du matériel. Est haut niveau ce qui est proche de l'utilisateur.

— **Noyau** — Le noyau (*kernel*) est le cœur du système. C'est lui qui gère le matériel pour les besoins des applications. Il est la partie principale du système d'exploitation.

— **Port** — Première définition, pour expliquer à quoi ça sert : un port est une sous-adresse, comme une boîte aux lettres dans un immeuble.

Seconde définition, technique : un port est un numéro qui définit, à l'intérieur d'une machine donnée, un point sur lequel un programme peut attendre des connexions de l'extérieur ou depuis lequel il peut en établir. La donnée du numéro IP de la machine et du port TCP constitue l'analogue d'un numéro de téléphone vers lequel ou depuis lequel on peut établir des connexions.

— **Shell** — Le shell est le programme qui interprète les commandes tapées au clavier. Selon le cas, il peut exécuter le programme demandé (`pine`, `ls`, `date`...), demander confirmation (avant d'effacer un fichier par exemple), corriger la syntaxe (`netscape` et pas `nestcape`), afficher un message d'erreur comme « `command not found` », etc. On peut aussi écrire des programmes (appelés scripts), dans le langage du shell, pour créer de nouvelles commandes, automatiser des tâches, ou autres.

Un shell est lancé dans chacun des terminaux que vous ouvrez. Pour quitter le shell, il suffit de taper `^D`, ce qui aboutit en pratique à fermer la fenêtre, car plus aucun programme ne tourne dedans.

Des pingouins partout

Alors, ça vole ou ça vole pas ?

Parmi les choses qui m'énervent, et il y en a beaucoup bien que ça ne se voit pas sur ma figure, car je suis d'un calme proverbial, c'est la confusion entre le pingouin et le manchot. Le manchot, c'est cette espèce d'oiseau aux ailes atrophiées et aux pattes encore plus petites, et qui vit là où il fait vraiment froid, c'est-à-dire en Antarctique (le pôle Sud, pour les ignares). Le pingouin ressemble un peu au manchot, mais il est plus petit, possède des ailes bien développées qui lui permettent de voler aussi bien qu'une mouette, il vit sur les côtes boréales, à des latitudes élevées (en Islande, par exemple), et, pour tout dire, il n'a avec le manchot qu'un lointain cousinage, autant qu'avec les canards, pour situer. Autrement dit, pingouin et manchot n'ont rien de commun, et sont séparés de pas loin de 20 000 kilomètres.

Mais quand on montre un manchot à un quidam quelconque, il l'identifie quasiment à coup sûr comme étant un pingouin, et il en est très content. C'est terrible comme l'inculture est satisfaisante. La confusion est certainement d'origine anglo-saxonne, d'abord parce que les Anglais ont toujours tout fait pour nous pomper l'air, à nous autres Français, et ensuite parce que chez eux, manchot et pingouin se traduisent tous deux par *penguin*.

J'en vois déjà qui se demandent ce que ces précisions ornithologiques et linguistiques viennent faire là, il y en a même qui se sentent lésés, après tout si on lit le *Hublot* c'est pour s'entendre dire qu'Unix c'est Bien et que le piratage c'est Mal, pas pour disserter sur les qualités aérodynamiques de volatiles même pas comestibles. Certes, je vous comprends, mais rassurez-vous, il y a un rapport avec l'informatique en général et Unix en particulier, j'y arrive de suite. Ce préambule était nécessaire pour bien vous faire ressentir l'effort surhumain que je vais produire en parlant de pingouins dans tout le reste de cet article, alors qu'il s'agit de manchots. Et puis, vous aurez appris quelque chose, c'est toujours ça de gagné.

Donc, la translation manchot/pingouin ayant été établie, j'aborde le vif du sujet : il y a des pingouins partout. Vous ne l'aviez peut-être pas remarqué, ce serait même tout à votre honneur, ça voudrait dire que dans les kiosques à journaux, vous consultez des opuscules normaux tels que *Paris-Match* ou *Le Monde*. Mais pour les dépravés qui se repaissent de ce qu'on nomme vulgairement la « presse informatique », c'est indubitable. Presque chaque mois apparaît un nouveau périodique dont le nom est une variation quelconque autour du terme « Linux », et décoré avec un goût très sûr de diverses effigies d'un pingouin rigolard, qui me filerait des cauchemars si j'avais encore cinq ans (ah, vieillesse ennemie, etc).

L'esprit inquisiteur, et je ne doute pas que vous soyez tous avides de connaissances, aura certainement compulsé quelques unes de ces parutions, en tout bien tout honneur et dans un but de recherche uniquement, j'en suis persuadé. À l'intérieur, on trouve des concepts étranges, parlant de logiciel qu'on donne mais qui rapporte des sous, de businessmen qui filent des ronds alors qu'il n'y sont en rien obligés, d'un gourou chevelu qui joue du pipeau, d'un esprit démoniaque américain qui serait autant mauvais programmeur qu'il est bon vendeur, et autres articles constellés d'acronymes au goût étrange venu d'ailleurs : Linux, BSD, Hurd, FUD, KDE, Tux, X11, etc...

Qu'est-ce donc que tout cela ? Et bien, synthétiquement, c'est la nouvelle mode : le logiciel libre. Je vais ici tenter de détailler un peu ce qu'il en est, car si l'honnête homme peut parfaitement se passer de ce genre de détails, ça peut toujours servir pour briller en société, ou réhausser un peu le niveau d'une fin de banquet.

Le commerce du vent

Le logiciel, en bon anglois *software*, c'est ce qui transforme un ordinateur complètement crétin en station de travail parfaitement crétine. L'ordinateur, à la base, c'est un amas savamment agencé de morceaux de silicium, aluminium, cuivre, plastique, et autres composants. Ça sait exécuter des ordres ; des ordres simples, vraiment très simples, exprimés dans un langage barbare qui n'a rien à voir avec nos complexes moyens d'expression humains. Ces ordres, ont les appelle le logiciel.

Le logiciel, c'est immatériel. Ça coûte très cher à produire, mais, *in fine*, ce n'est jamais qu'une suite de 0 et de 1, comme tout ce qui rentre dans un ordinateur, et on peut le reproduire autant de fois qu'on veut. Le logiciel se vend bien, et il y en a partout où il y a l'ombre d'un semblant d'ordinateur, donc dans les voitures, les fours à micro-ondes, les cartes bleues, les lecteurs de CD, et bien d'autres objets, dont on n'aurait pas imaginé tant de malice.

La forme de logiciel compréhensible par les ordinateurs est, disons-le tout net, immondissime. Alors les développeurs utilisent des outils informatiques qui traduisent automatiquement en un logiciel utilisable sur un ordinateur, des descriptions de ce logiciel en un autre langage, plus facilement appréhensible (par un informaticien, hein, ce n'est quand même pas du français). La forme compréhensible, c'est ce qu'on appelle le code source, ou tout simplement le source (oui, au masculin – même si certains déviants l'utilisent au féminin). La traduction, on appelle ça la compilation, et le logiciel tel qu'il est avalé par la machine, c'est le binaire.

Au passage, évoquons le système d'exploitation : c'est ce logiciel particulier qui sait parler au matériel ; les autres logiciels (on dit « applications », voire « progiciels » quand on veut faire pro), quand ils veulent lire un fichier ou afficher un texte sur l'écran, se contentent de faire une requête au système d'exploitation. L'équivalent sur une voiture, c'est le tableau de bord : on tourne le volant quand on veut, en fait, tourner les roues. Et si on change les pneus, qu'importe, c'est toujours le même volant et ça se manipule pareil. Le conducteur n'a pas à connaître tous les types de pneus, il peut se contenter d'identifier le truc rond, au milieu du tableau de bord. Microsoft Windows est un système d'exploitation, Microsoft Word est une application.

Il y a 20 ans, quand une boîte quelconque fournissait un logiciel à une autre boîte ou un particulier, elle vendait le binaire, et le binaire seulement, pour plein de brouzoufs. Elle amortissait le coût de développement sur les différentes ventes, et si le logiciel se vendait bien, jackpot, puisque la reproduction ne coûte rien. D'ailleurs, ce coût nul (en fait presque, un CD-ROM ça coûte 7 FF) de la copie a entraîné beaucoup de copies faites par des tiers (tout bénef, pour eux, puisqu'il ont des coûts de développement nuls), d'où une législation adaptée à base de licences d'utilisation.

Depuis, un illuminé ricain, issu d'une université spécialisée dans ce genre de cas, a eu une idée qui peut sembler idiote, mais qui doit contenir de vrais morceaux d'intelligence dedans puisqu'elle se vend bien : le logiciel, on va le fournir avec les sources. Oui, on file nos petits secrets de fabrication avec. Et puis, tant qu'à faire, on autorise celui qui a récupéré le logiciel d'en faire autant de copies qu'il veut. Il peut les donner, ces copies, ou les vendre au prix qu'il veut à qui il veut. Le susdit amerloque a fait des émules, fondé une religion (lui, il dit « fondation » mais c'est clairement une secte), promulgué un dogme, et bien évidemment il y a eu des schismes et des guerres. J'y reviendrai plus tard.

Qu'est-ce que c'est que cette utopie, vous demandez-vous. On dirait un délire communautaire issu tout droit des années 1970. Un poison crypto-communiste, une pourriture bolchévique qui tente d'empêcher de braves gens de devenir honnêtement richissimes. En effet, qu'est-ce qui empêche une société d'acheter un exemplaire du logiciel libre d'une autre société, et de le revendre tel quel mais moins cher ? Plus de coût de développement, juste de la distribution, et le marché est acquis. On dirait un encouragement au piratage, une théorisation du masochisme. Ça ne peut pas tourner, une fantasmagorie pareille.

Et pourtant, elle tourne. Comment ? Pourquoi ? Que va faire le pingouin ? Vous le saurez au prochain épisode...

▷ Thomas Pornin