

INTRODUCTION AU SYSTEME D'EXPLOITATION UNIX

Michel R. Dagenais
département de génie électrique
et de génie informatique
Ecole Polytechnique de Montréal

HISTORIQUE

MOTIVATIONS

- Un système d'exploitation différent pour chaque architecture
- Systèmes de traitement par lot
- Une tâche interactive à la fois
- Chaque périphérique est accédé différemment

ORIGINES DU SYSTEME UNIX

- Chercheurs de GE, du MIT et de Bell Labs collaborent à la réalisation de MULTICS. (1965-1969)
- Chercheurs de Bell Labs implantent un petit système d'exploitation pour leurs besoins personnels (AT&T ne peut vendre de matériel informatique). (Ken Thompson, Dennis Ritchie et al. 1969-1973)
- Utilisation du langage C (dérivé de B et BCPL), une sorte d'assembleur de haut niveau transportable.
- Système multi-usager, multi-tâche, développé sur PDP-11 en C.
- Des copies sont distribuées à coût nominal aux universitaires intéressés. (125 en 1977)
- AT&T System V en 1983, BSD 4.2 peu après. En 1984 on comptait 100 000 systèmes UNIX.

VERSIONS EN UTILISATION

- Berkeley travaille sur UNIX et sort une version grandement améliorée (BSD 4.0, 4.1, 4.2, 4.3).
- Plusieurs universités se convertissent à UNIX BSD
- AT&T intensifie le développement de UNIX pour rattrapper BSD (mémoire virtuelle) avec UNIX System V release 3
- Plusieurs compagnies utilisent BSD 4.2: SUN, DEC...
- ou System V release 3.2: SCO, NCR, IBM
- ou les deux.

VERSIONS EN DEVELOPPEMENT

- SunOS et AT&T convergent vers System V release 4
- IBM, HP, DEC et plusieurs autres développent OSF/1
- Carnegie-Mellon travaille sur MACH
- Berkeley termine avec 4.4BSD
- Le projet GNU
- Dérivés gratuits de Berkeley: FreeBSD, BSD386
- Linux
- POSIX

SYSTEM V RELEASE 4

- Supporte les facilités de Berkeley et System V release 3
- Quelques ajustements dans les zones de recouvrement
- Extensions prévues pour les multi-processeurs et la haute sécurité
- Création de Unix Software Laboratories
- Interface graphique X windows avec OPEN LOOK, éventuellement battue et remplacée par MOTIF.
- Devait réconcilier tout le monde

OPEN SOFTWARE FOUNDATION

- Système OSF/1 basé sur MACH et UNIX
- Devrait être utilisé par DEC, HP et IBM
- Mode de compatibilité BSD et System V
- Facilités multi-processeurs et temps réel
- Première version en retard et performance moyenne
- Interface graphique X windows avec MOTIF

LE SYSTEME MACH DE CARNEGIE-MELLON

- Technologie à petit noyau qui gère les messages
- Services élaborés inspirés de BSD 4.3 fournis par tâches externes:
 - système de fichier
 - gestion de priorité
 - accès aux périphériques...
- Prévu pour les multi-processeurs
- Besoins réduits en mémoire pour les applications spécialisées
- Bonne performance temps réel

LE PROJET GNU

- GNU's not UNIX
- Version améliorée de UNIX et indépendante de AT&T

- Noyau MACH, système de fenêtre X windows
- Services style BSD 4.3 en développement (the HURD). Version 0.1 disponible en septembre 1996.
- Applications presque terminées: Emacs, GCC, GDB, Groff, Ghostscript, xInfo, tar, cpio et bien d'autres
- Linux fut développé par Linus Torvalds en Finlande et utilise les applications de GNU. Linux fut prêt bien avant HURD et est très utilisé.

LE STANDARD POSIX

- Effort de l'IEEE endossé par les compagnies et gouvernements
- Description fonctionnelle de UNIX
- Ensemble de fonctions de système qui doivent exister
- Applications et formats de données qui doivent exister
- VMS/POSIX, MVS-POSIX...

QUOI ATTENDRE ?

- Technologie à petit noyau
- Support multi-processeur et temps réel
- Facilités pour sécurité et internationalisation
- Services de type BSD et System V supportés
- X windows
- Versions gratuites disponibles
- Attention aux extensions incompatibles...
- Applications de gestion (bases de données, administration de système) non standardisées

CONCEPTS DE BASE

L'USAGER

- Numéro d'utilisateur (UID)
- Mot de passe
- Maison
- Interpréteur de commande
- Groupe par défaut
- Informations complémentaires

LES GROUPES

- Nom
- Numéro de groupe
- Liste de membres
- Un usager peut appartenir à plusieurs groupes

- Un usager appartient à un groupe par défaut qui peut être changé dynamiquement

LES PROCESSUS

- Numéro de processus
- Usager ayant créé le processus
- Numéro d'utilisateur effectif associé au processus
- Groupe associé au processus
- Parent du processus
- Variables d'environnement
- Création de processus par dédoublement (fork)
- Changement de module exécutable associé au processus (exec)
- Code de retour
- Statistiques d'exécution

LES SIGNAUX

- Les processus d'un même usager peuvent s'interrompre via des signaux
- Suspension de l'exécution
- Destruction du processus
- Copie du processus sur disque pour analyse

L'USAGER ROOT

- Peut envoyer des signaux à tous les processus
- Peut devenir n'importe quel autre usager
- Possède tous les privilèges
- Ne peut cependant trouver le mot de passe des usagers mais il peut le changer

LES PERIPHERIQUES

- Chaque classe de périphérique a un module de commande associé (device driver) avec un numéro qui lui est propre (major number)
- Dans chaque classe il peut y avoir plusieurs périphériques connectés, avec un numéro associé à chacun (minor number)
- Chaque module de commande doit accepter un certain nombre de commandes de base (open, read, write, rewind, close...)
- Certaines catégories ont des commandes supplémentaires
- Les commandes très spécifiques passent par la fonction ioctl().

LE SYSTEME DE FICHER

- Un système de fichier par partition de disque
- Têtes de fichiers (inodes) pour date (création, modification et accès), propriétaire (usager et groupe), liste de blocs...

- Liste de blocs pour fichiers ordinaires et répertoires
- Numéro de périphérique pour les *devices*
- Chaîne de caractère pour les liens symboliques
- Code spécial pour les *named pipes* et *sockets*
- Les répertoires contiennent les noms et les *inodes* associés
- Possibilités de quotas par système de fichier

CONNEXIONS ENTRE ARBRES DE FICHIERS

- Une partition est définie comme racine au départ
- Une table de connexions (*mount table*) est construite qui connecte les systèmes de fichiers d'autres partitions sur des répertoires.
- Avec NFS il est possible de connecter des portions de l'arbre de fichier d'un autre ordinateur, relié en réseau, sur un répertoire.
- Le contenu des répertoires sur lesquels se fait la connexion n'est plus accessible
- Avec ces connexions, le système fait la traduction *nom de fichier* à *device et inode*
- On peut avoir plusieurs disques, chacun avec plusieurs partitions. La répartition des partitions se fait au moment du formatage. Certains systèmes limitent les partitions à 1GB.

PERMISSIONS D'ACCES SUR LES FICHIERS

- Chaque fichier a un propriétaire et un groupe
- Il y a trois ensembles de permissions: user, group, other
- Pour chacun on a: read, write, execute, setuid, sticky
- Pour les répertoires *execute* veut dire traverser
- Il se peut qu'on ne puisse écrire un fichier mais qu'on puisse le remplacer.
- Root peut changer le propriétaire et le groupe des fichiers

CARTE DE L'ARBRE DE FICHIERS

- /vmunix: système d'exploitation
- /dev: fichiers pour accéder les périphériques
- /etc: fichiers de configuration du système
- /sys: fichiers de configuration du système d'exploitation
- /bin: exécutables proches du système
- /tmp: espace disque temporaire
- /usr/users: comptes des usagers

CARTE SUITE

- /usr/bin: exécutables
- /usr/lib: bibliothèques
- /usr/man: documentation
- /usr/include: fichiers de définitions .h pour programmation en C
- /usr/src: source de UNIX

- /var/adm: fichiers d'administration de système
- /var/spool: queues
- /usr/local: emplacement pour logiciels ajoutés avec sous-répertoires bin, lib, man, include, src...

EMPLACEMENT LOGIQUE VS PHYSIQUE

- Chaque partition est montée sur /mnt/machine/no-de-partition.
- Sur chaque partition on retrouve users, local/src, local/man, local/include, local/bin.sun3, local/bin.sun4, local/lib.sun3, local/lib.sun4
- Des liens symboliques relient les usagers. Ainsi /usr/users/schtroumpf pointe à /mnt/erda/0/users/schtroumpf
- Des liens symboliques relient les répertoires locaux. Ainsi /usr/local/bin pointe à walhalla:/mnt/walhalla/0/local/bin.sun4
- On peut utiliser les connexions (mount ou automount) à la place des liens symboliques.

FICHIERS IMPORTANTS

- /etc/passwd: Liste des usagers
- /etc/group: Liste des groupes
- /etc/rc: Commandes exécutées au départ du système
- /etc/ttytab: Liste des terminaux actifs
- /etc/gettytab: Tâche initiale pour chaque type de terminal
- /etc/termcap: Caractéristiques de chaque sorte de terminal
- /etc/printcap: définitions d'imprimantes
- /etc/fstab: Liste des connexions (systèmes de fichiers montés)
- /etc/hosts: Liste des autres systèmes connus sur le réseau

AUTRES FICHIERS

- /var/spool/mail: Boîtes postales des usagers
- /var/yp/: Base de données réseau pour le système
- /var/adm: Statistiques d'utilisation du système
- /var/spool/lpd: Queues d'impression
- /usr/lib/sendmail.cf: Configuration du courrier électronique

TERMINAUX

- Terminal fixe: sorte et vitesse, programme getty attend une requête puis exécute login le moment venu et finalement donne la propriété du port à l'utilisateur qui se connecte.

- Connexion par modem: vitesse et sorte variables, programme getty doit être interrompu si le modem est utilisé en sortie.
- Terminal virtuel réseau: pas de vitesse, sorte variable. Pas de getty.
- Serveur X-windows: chaque fenêtre terminal (xterm) est un terminal virtuel. L'utilisateur connecté à la fenêtre initiale contrôle le serveur.

SESSION A UN TERMINAL

- Le programme login demande le nom et le mot de passe, encrypte le mot de passe puis le compare à la version encryptée qui est dans le fichier /etc/passwd.
- Finalement login démarre l'interpréteur de commande (shell).
- Le processus Init qui était le parent de getty, l'est aussi de login et du shell.
- Un processus démarré via le shell a son entrée et sa sortie connectées au terminal du shell et a le shell pour parent.
- Un processus qui exécute en arrière plan perd le contrôle du terminal et tombe en attente s'il veut lire du terminal.
- Lorsque l'utilisateur s'en va, les processus d'arrière plan sont adoptés par Init.
- Lors de perte de connexion (modem) le shell reçoit un signal de terminaison.

ACCES AUX PERIPHERIQUES

- Les périphériques sont accédés comme des fichiers (permission, open, read, write, seek...)
- Le terminal ou la mémoire video, la souris et le clavier appartiennent à l'utilisateur courant
- Accès avec queue d'attente par lpr (imprimante, traceuse, ruban...)
- Accès partagé par verrouillage (ruban)
- Même la mémoire physique et la mémoire virtuelle sont accessibles

ENVIRONNEMENTS D'INTERACTION ET COMMANDES DE BASE

LES INTERPRETEURS DE COMMANDES

- Le vieux Bourne shell (sh) à utiliser pour les scripts puisque c'est le plus portable
- Le C shell (csh) de Berkeley qui offre des facilités interactives plus avancées
- Le Korn shell de AT&T qui remplace le Bourne shell. Facilités supplémentaires dont le mode d'édition des commandes précédentes
- BASH de GNU, un autre excellent produit FSF disponible gratuitement sur la plupart des stations UNIX.
- Le langage pour les scripts plus complexes: Perl, mieux adapté pour la programmation

LE ROLE DE L'INTERPRETEUR

- Recevoir les commandes de l'utilisateur et faire les appels système en conséquence
- Commandes intrinsèques exécutées directement par le shell
- Les autres commandes activent le programme associé à la commande
- Le shell interprète la commande, trouve le module exécutable, construit la liste d'arguments et exécute le tout

SYNTAXE DES COMMANDES

- Le \$ introduit une variable d'environnement, de shell ou d'argument
- Les expressions régulières construisent des listes de fichiers en argument
- Les ' et " empêchent le blanc d'agir comme séparateur et cachent les expressions régulières, le ' cache les \$
- Le permet de placer un caractère pour lui-même
- Le ; sépare des commandes à exécuter en séquence
- Le | sépare des commandes en pipeline
- Le < et > redirigent l'entrée et la sortie
- Le & sépare des commandes asynchrones
- Les () permettent de grouper les commandes (sub shell)
- Le ` instancie la sortie d'une commande

EXPRESSIONS REGULIERES POUR FICHIERS

- ., ..: répertoire courant, parent
- , user: résidence de l'utilisateur
- ?: n'importe quel caractère
- [a-h]: un caractère compris entre a et h
- *: une chaîne de caractère quelconque
- Dans egrep, a | b: veut dire expression a ou expression b

LES VARIABLES D'ENVIRONNEMENT

- CWD: répertoire courant
- HOME: répertoire de résidence de l'utilisateur
- PATH: répertoires à examiner pour trouver les commandes
- TERM: sorte de terminal
- DISPLAY: où afficher les applications X windows
- MAIL: quel fichier surveiller pour l'arrivée de nouveau courrier
- Environnement passé de processus parent à enfant

SYNTAXE DES COMMANDES

- Nom de la commande (peut correspondre à un alias du shell) à exécuter qui doit se trouver sur le chemin d'exécution ou commencer par la racine (/)

- Liste de paramètres (switches) introduits par un - et dont certains sont accompagnés de valeurs
- Liste de fichiers ou de répertoires
- [*optionnel*] ... répété
- exemple: cp [-r -p] filename ... directory

FICHIERS D'INITIALISATION

- `${HOME}/.login`: interprété par le shell lors de la connexion
- `${HOME}/.profile`: interprété lorsque sh démarre
- `${HOME}/.cshrc`: interprété lorsque csh démarre
- `${HOME}/.plan`, `.project`: information sur l'utilisateur
- `${HOME}/.mailrc`, `.exrc`, `.xstartup`, `.dbxinit`: fichiers d'initialisation pour les différentes applications comme mail, vi, X windows, dbx...

COMMANDES DE BASE

- `cd`, `pushd`: changer de répertoire
- `popd`: revenir de `pushd`, `pwd`: imprimer le répertoire courant
- `ls`: lister le contenu du répertoire
- `cp`: copier des fichiers
- `mv`: déplacer ou changer de nom un fichier
- `rm`: effacer des fichiers ou répertoires
- `mkdir` et `rmdir`: créer et effacer un répertoire
- `^C`, `^Z`: tuer ou interrompre un processus
- `setenv`: manipuler les variables d'environnement

COMMANDES ESSENTIELLES

- `man` commande, `man -k` mot-clé: lister le manuel associé à une commande ou les commandes associées à un mot clé
- `cat`, `more`, `less`: lister le contenu d'un fichier
- `ps`: lister les processus en exécution
- `kill`: envoyer un signal (tuer ou interrompre) à un processus
- `bg`: repartir un processus en arrière plan
- `fg`: repartir un processus interactivement
- `tset`, `stty`: configurer un terminal
- `chmod`: changer les permissions d'accès
- `passwd`: changer le mot de passe

COMMANDES UTILES

- `find`: trouver un fichier par son nom
- `grep`: trouver un fichier par son contenu
- `diff`: trouver les différences entre deux fichiers
- `wc`: compter les lignes et mots

- tar: regrouper un arbre de fichiers en un seul (pour ruban)
- compress: réduire l'espace occupé par un fichier
- df: statistiques sur les systèmes de fichiers
- du: espace disque utilisé
- file: deviner le type d'un fichier
- split: diviser un fichier

COMMANDES POUR COMMUNIQUER

- w: qui est sur le système
- mail: envoyer un message électronique
- write: écrire un message à l'écran
- talk: conversation interactive entre usagers
- finger: trouver l'information sur un usager

COMMANDES D'IMPRESSION

- lpr -Pprinter file: imprimer un fichier
- lpq -Pprinter: afficher la queue d'impression
- lprm -Pprinter: enlever les tâches en queue d'impression
- pr: programme de mise en page simple

PROGRAMMATION DE SHELL

- #! /bin/sh
- for var in list do ... done
- if ... then ... else ... fi
- echo: afficher les arguments
- test: vérifier certaines conditions
- expr: calculer des expressions plus complexes

L'EDITEUR VI

- Mieux que EDLIN, assez puissant même, mais peu convivial
- 3 modes: commande vi (mode initial), commande ex (obtenu avec :), insertion
- En insertion: chaque caractère est inséré, ESC retourne en mode vi et ^ V introduit un caractère special comme ESC.
- En mode vi: i insert, a append, o open line, /string/RET recherche, x efface caractere, dd efface ligne, h j k l gauche, monte, descend, droite, : mode ex
- En mode ex: wq sauve et sort, s/A/B/gc remplacement de A par B global avec confirmation...
- Offre toutes les facilités voulues, incluant les macros mais souffre de son âge, de l'audience visée et de la compatibilité

L'EDITEUR EMACS

- Langage de programmation LISP complet avec extensions pour l'édition de textes
- Peut à peu près tout faire, même être convivial, il suffit de programmer le tout
- Sert pour éditer, formater, vérifier l'orthographe, compiler, envoyer le courrier, lire les nouvelles, transférer les fichiers
- Les experts emacs ne se sont pas souciés des débutants emacs et la convivialité vient encore en option
- Peut attacher n'importe quelle commande à n'importe quelle clé ou séquence de clés
- Fonctionne sur à peu près toutes les sortes de terminaux, les définitions par défaut s'inspirent des machines LISP
- Les extensions graphiques (fontes proportionnelles, menus...) s'en viennent

COMMANDES EMACS DE BASE

- ^ X^ F ouvrir fichier, ^ X^ S sauvegarder, ^ X^ W écrire ailleurs, ^ X^ C sortir, ^ Z suspendre et fg reprendre
- ^ X2 deux fenêtres, ^ X1 une fenêtre, ^ Xo autre fenêtre
- ^ S rechercher, ESC R remplacer, ^ SPACE définir point, ^ W effacer à partir du point, ESC W mémoriser à partir du point, ^ Y copier ce qui a été effacé ou mémorisé
- ^ B ^ F ^ P ^ N arrière, avant, ligne précédente et suivante, ^ K détruire le reste de la ligne, ^ D effacer caractère, ^ Xu undo, ^ H help
- ESC X met en mode commande: apropos keyword, switch-to-buffer, describe-mode (aide contextuelle), info, rmail, gnus (nouvelles), gdb (debugger)
- Chaque mode ajouté par des fichiers emacs-lisp ajoute de nouvelles commandes qui sont parfois décrites avec ?

LE TRAITEMENT DE TEXTE SOUS UNIX

- nroff: dérivé de runoff et script. Macros simples avec noms limités à deux caractères. Sortie sur écran ou papier
- troff: comme nroff mais sortie de bonne qualité sur équipement de typographie et maintenant imprimante laser
- eqn: préprocesseur qui traite les équations mathématiques
- tbl: préprocesseur qui compose les tableaux
- refer: système qui gère les listes de références
- spell: dictionnaire anglais américain et britannique
- pic: préprocesseur pour les diagrammes

DOCUMENTATION ELECTRONIQUE

LA DOCUMENTATION AVEC MAN

- Utilise nroff avec les macros spéciales man
- Décomposé en plusieurs sections: 1 commandes, 2 appels système, 3 fonctions de librairie, 4 format de périphériques, 5 formats de fichiers, 6 jeux, 7 macros de formattage, 8 administration
- Un fichier de description pour chaque commande: /usr/man/manX/commande.X, où X est le numéro de section
- Chaque fichier contient une description abrégée, d'une ligne, utilisée pour les mots clés

AUTRES TRAITEMENT DE TEXTE

- Info: documentation hypertexte, basé sur SGML et qui vient avec emacs
- LaTeX: système de compilation de texte beaucoup plus avancé que troff, excellent pour les équations
- Interleaf et FrameMaker: systèmes commerciaux de typographie
- HTML: format du World Wide Web. Plusieurs editeurs WYSIWYG existent maintenant pour ce format.

LE SYSTEME DE FENETRAGE

LE SYSTEME X WINDOWS

- Protocole pour définir des fenêtres de pixels en mémoire ou sur écran et y afficher des primitives graphiques
- Les commandes sont envoyées des clients (applications) vers le serveur (serveur X pour l'écran) par messages qui peuvent passer par le réseau
- Le serveur X communique aux clients tous les événements qui les concernent et les intéressent: clé pressée, déplacement de la souris, zone de la fenêtre recouverte ou découverte
- Un gestionnaire de fenêtre est une application qui prend le contrôle de la fenêtre de fond et qui coordonne l'interaction de l'utilisateur avec les applications: twm, mwm, gwm, olwm, olvwm...
- Des bibliothèques permettent de créer des menus, des barres de défilement, des entêtes, des boutons de commande... dont l'apparence varie d'une à l'autre
- Le style Open Look de SUN et AT&T, et Motif (basé sur présentation manager) de OSF

APPLICATIONS COURANTES SUR X

- xterm: émulateur de terminal vt100 et tektronix
- tn3270: émulateur 3270
- Editeurs de diagrammes idraw, xfig...
- Interpréteur PostScript ghostscript, xps

- Calculateur, horloge, surveillance de la performance
- Documentation, librairies graphiques GKS et PHIGS, jeux
- Plusieurs applications commerciales de traitement de texte, CAO et autres

EVOLUTION DE X

- Version X11R6
- Plus de contrôle sur qui affiche quoi
- Fontes par contour plutôt que par pixels
- Performance raisonnable
- Graphisme 3D avec PHIGS
- Plusieurs applications commencent à le supporter
- Modèle de bas niveau basé sur le pixel, librairies de plus haut niveau et gestionnaires de fenêtres mal standardisés

ACCES AUX RESSOURCES, INCLUANT LE RESEAU

ORGANISATION DU RESEAU

- Chaque ordinateur a un nom et une adresse unique
- L'adresse (pollux.vlsi.polymtl.ca ou 132.207.8.31) se décompose en réseau (vlsi.polymtl.ca ou 132.207.8) et machine (pollux ou 31)
- La correspondance entre l'adresse numérique et symbolique peut provenir d'un fichier (/etc/hosts) ou être obtenue dynamiquement via le serveur de nom
- Certains ordinateurs sont connectés à plusieurs réseaux et font le pont
- Le matériel réseau sous-jacent utilise sa propre numérotation (adresse Ethernet 48 bits)

LES FICHIERS DE SYSTEME EN RESEAU

- Plusieurs fichiers de système sont communs à un réseau
- Les fichiers passwd, group, hosts... sont placés dans une base de donnée avec serveur répliqué (Network Information Services, anciennement Yellow Pages)
- D'autres fichiers doivent être redistribués autrement (mot du jour...)
- Problèmes de sécurité avec la redistribution de fichiers de système

LE COURRIER ELECTRONIQUE

- Le serveur de nom s'occupe spécialement des adresses de courrier électroniques
- Des alias peuvent facilement être définis
- Possibilité de faire suivre le courrier
- Le courrier peut arriver sur une machine particulière ou sur un serveur accessible de tout le réseau

PARTAGE DE DISQUES ET PERIPHERIQUES

- On peut facilement accéder les imprimantes ou rubans connectés en réseau
- Avec le Network File System (NFS) on peut monter une portion de système de fichier d'un serveur sur n'importe quel client
- Un ordinateur peut être serveur et client et communiquer avec d'autres ordinateurs proches ou loin, par Ethernet ou même par modem
- Il est aussi possible pour un ordinateur de n'avoir aucun disque local
- Système de permission basé sur l'équivalence au niveau de l'utilisateur ou de l'ordinateur

IMPLICATIONS AU NIVEAU DE LA SECURITE

- Equivalence d'un utilisateur veut dire faire confiance à l'utilisateur et aux ordinateurs et réseaux qu'il utilise
- Equivalence d'un ordinateur veut dire faire confiance à cet ordinateur et au réseau
- Limiter les permissions réseau ou assurer la sécurité physique des serveurs ainsi que du câblage
- L'utilisation de l'encryption peut réduire la vulnérabilité liée au câblage
- Le réseau rend les trous de sécurité accessibles à la planète

LES SERVICES RESEAU

- Telnet et rlogin: connexion à distance avec émulateur de terminal
- Rsh: exécution à distance d'une commande
- Ftp: transferts de fichiers, un nombre impressionnant de serveurs ftp disponibles
- Archie: base de données des fichiers disponibles sur des serveurs ftp à travers le monde
- Usenet News: babillard électronique mondial, principalement pour les universitaires dans le domaine scientifique
- De plus en plus d'informaticiens refusent un travail s'ils n'ont pas accès au *RESEAU* (Internet)

ENVIRONNEMENT DE DEVELOPPEMENT

- Librairie C assez bien standardisée (POSIX) (libc)
- Compilateur C optimiseur avec éditeur de lien (gcc)
- Utilitaires pour manipuler les bibliothèques (ar)
- Environnements de mise au point plein écran (gdb)
- Gestion automatique des dépendances (Make)
- Etude de performance (gprof)
- Mesures de couverture des énoncés par les tests (tcov)
- Systèmes de contrôle des versions (rcs, cvs)

SOURCES D'INFORMATION SUPPLEMENTAIRES

LIVRES ET REVUES

- Les revues BYTE et UNIX WORLD
- The UNIX programming environment, Kernighan et Pike
- The design of the UNIX operating system, Bach
- Internetworking with TCP/IP, Comer
- The C programming language, Kernighan et Ritchie
- L'éditeur O'Reilly & Associates de Boston

ASSOCIATIONS

- USENIX
- UNIFORM
- IEEE/Computer
- Association for Computing Machinery
- League for programming freedom

BABILLARDS ELECTRONIQUES

- Comp.lang: langages informatiques
- Comp.unix.programmers, comp.unix.admin
comp.unix.wizard: Unix
- Comp.sources.unix: sources de programmes utiles
- Comp.sys.sun, comp.sys.sgi...: marques spécifiques d'ordinateurs
- Comp.database, comp.graphics, comp.arch, comp.newprod...

LES PRINCIPAUX LOGICIELS DISPONIBLES

- X windows V11R6: système de fenêtrage
- Emacs: éditeur puissant
- Gcc, g++, libg++, gdb: environnement de développement
- TeX et LaTeX: traitement de texte
- Gnuplot, Interviews: logiciels graphiques
- Ghostscript: interpréteur PostScript
- Khoros: visualisation graphique
- Postgres95: base de donnée orientée objet
- Modula-3, Oberon, Sather, Hermes, Smalltalk, Lisp, SML: compilateurs ou interpréteurs pour langages de programmation

CONCLUSION

- Versions gratuites puissantes et très répandues
- Multi-processeurs
- Logiciels prêt à porter multi vendeurs
- Applications distribuées et orientées objet
- Sécurité accrue