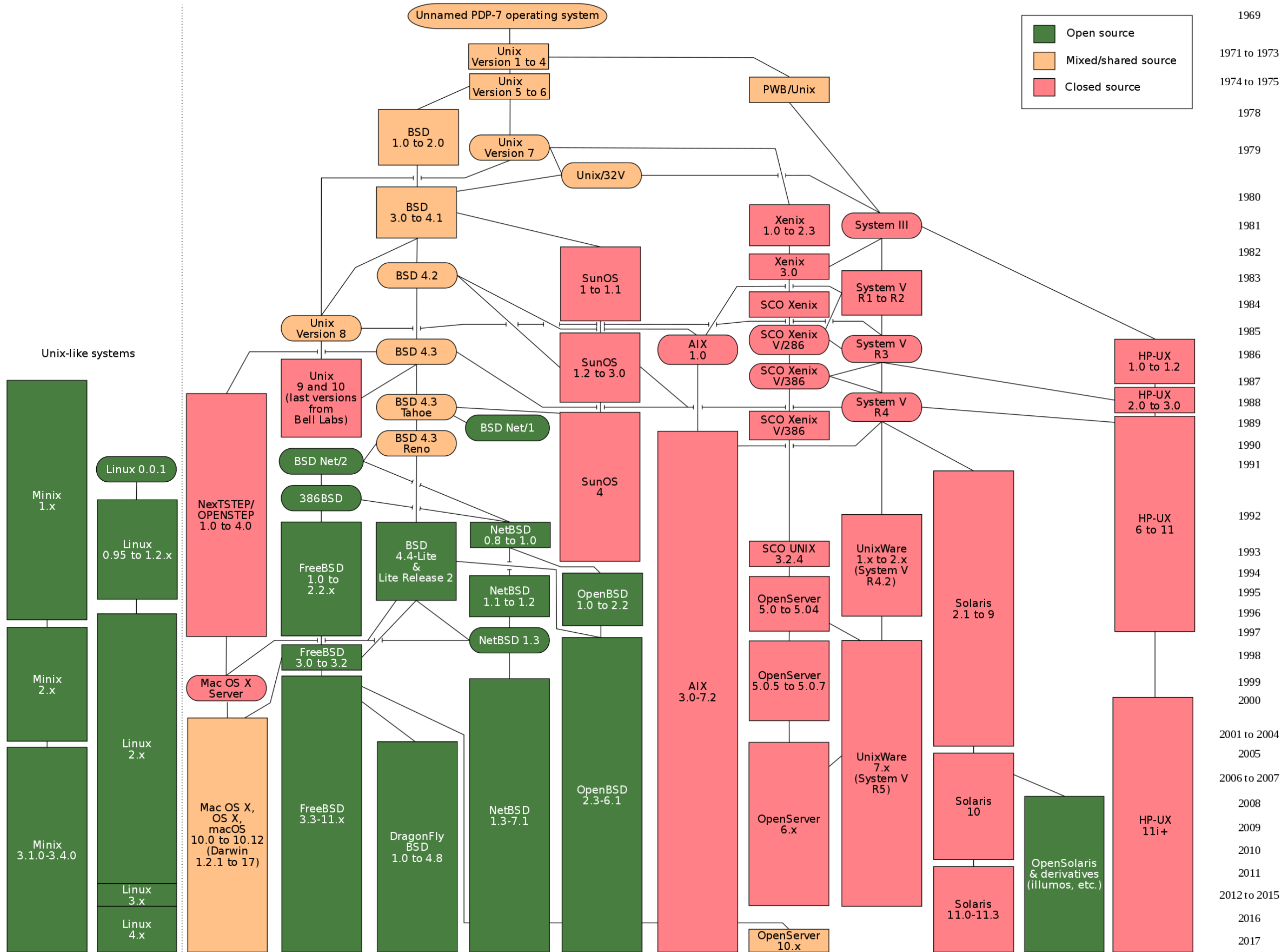


CH2 Administration UNIX

Introduction




Source: Wikipedia

Particularités

- **Ecrit en C, sources disponibles**
- **Fortement scriptable**
La plupart des outils de gestion de services et de démarrage sont des scripts.

Familles

- BSD: open source, fort actif, peu de support
- SysV: commercial, support, backups
- 3 standards
- Filesystem hierarchy standard 
- Portable Operating System Interfaces (POSIX)
- OpenGroup brands: Unix

Linux

- **Début des années 90**
Linus Torvalds: kernel
- **Unix le plus actif de nos jours**
En particulier sur Workstation
- **Supporte énormément de périphériques**

Unix

- **Kernel**
Cœur du système
- **Filesystem**
Structure du système de fichiers
- **Shell**
Interface en ligne de commandes
- **Interface GUI (optionnel !)**
Gnome, KDE, ...

Unix: kernel

- Chargé au démarrage par le bootloader
- Gestion des processus, de la mémoire et des périphériques
 - Scheduling (cf. Joël Goossens)
 - Gestion du système de fichiers
- Rôle: abstraction hardware

Pourquoi l'administration système sous Unix?

- Système le plus utilisé sur les serveurs Internet
- Souvent utilisé pour l'infrastructure DNS, DHCP, routage, firewall, ...

Utilisations

- Workstation
- En réseau
 - Serveur (accès via machine Win/Linux)
 - Client / Client léger (diskless)
 - Embarqué (routeurs)

Du command line au GUI

- Initialement: seul CLI (command line interface)
 - une commande à la fois
 - efficace et rapide
 - complexité de manipulation: que taper ?

Consoles multiples

- Utilisation multi-tâches de la CLI
- Disponible sur la plupart des systèmes Unix
- ALT-F1
- ALT-F7 retourne à la console graphique, si existante

Du command line au GUI

- XWindow, également appelé X11 ou X
 - Serveur graphique, sur lequel vient se greffer une interface utilisateur
 - Desktop environment: Gnome, KDE
 - Window manager: Metacity, Compiz, ...
 - Requiert des ressources
Donc peu utilisé en administration de systèmes...

Connexions à distance

- Command line: ssh ou telnet
- Graphique: avoir un serveur X sur la machine distante

Demo

- Notion de Window manager
- Retour à la console

Partie I : Utilisateurs

Différentes classes

- Root (utilisateur 0)
- Autres utilisateurs
 - Plus de finesse: groupes, ou attributs

Tous les Unix ne sont pas égaux...

- Utilisation du root fort critiquée, et théoriquement non nécessaire dans sa forme classique
- Solaris, par exemple, est très différent. Ici nous présenterons l'approche de Linux.

Groupes

- Regroupement d'utilisateurs
 - Soit pour leur donner des accès identiques
 - Soit pour la facilité de l'administrateur
Ex: étudiants en BA3 info

/etc/passwd et /etc/group

- Fichiers textes, éditables à la main
- /etc/shadow
Attributs étendus et mots de passe
encryptés

Commandes utiles

- adduser
- addgroup
- userdel
- groupdel
- id, who, whoami

Se faire passer pour un autre...

- pour...
- réaliser une opération spécifique
sudo reboot
- être sûr qu'un logiciel ne fait pas de dégâts
FTP tourne sous l'utilisateur ftp

Se faire passer pour un autre...

- `su bob`
- `sudo shutdown`

SU

- lancement d'un shell sous un autre nom d'utilisateur
- en tapant son mot de passe
- ou directement, si on est root

sudo

- **Exécution d'une commandes nécessitant des privilèges pour un utilisateur n'ayant pas les droits correspondants**
- **Logging de l'action effectuée**
Très utile pour l'administration système
- **Flexibilité**
/etc/sudoers, liste de commandes par utilisateur

MAKE ME A SANDWICH.



SUDO MAKE ME
A SANDWICH.



WHAT? MAKE
IT YOURSELF.



OKAY.



Importance de sudo

- **Un mot de passe par utilisateur**
Important quand les utilisateurs changent
- **Pas le choix! (ou presque)**
Ubuntu, Mac OS X n'ont pas de compte root par défaut

Importance de sudo

- Création d'utilisateurs ne correspondant pas à des humains, mais à des fonctions Administrateur, différentes classes d'utilisateurs
- sudo pour toutes les commandes privilégiée
- Attention: souvent l'utilisation des groupes suffit

Pas parfait...

- **sudo bash**
Pas de logging des commandes effectuées
- **Logging ne remplace pas la documentation**

Démo

- sudo
- contenu classique des fichiers
- liste des utilisateurs connectés
- info sur l'utilisateur courant

Partie 2: Système de fichiers

Concept de base

- Une seule hiérarchie
- Mount points
- Formats classiques: ext2, ext3, ext4
- Fonctionnement : table des inodes, voir cours de Fonctionnement de l'ordinateur

Version “classique” : FHS

- /
 - bin
 - boot
 - dev
 - etc
 - lib
 - sbin
 - usr
 - var
 - home or Users

Version “classique” : FHS

- /usr
 - local
 - bin
 - bin
 - sbin
 - ...

/bin /sbin

- Contiennent les programmes de base du système.
- sbin contient les commandes que seul l'administrateur doit utiliser

/usr

- Contient tous les programmes et leur documentation
- En particulier
 - /usr/local(/bin) : logiciels locaux (ex: scripts locaux, logiciels compilés)
 - /usr/bin : la plupart des logiciels

/var

- Utilisé pour des fichiers de données variables ou modifiés durant l'exécution du système
- En particulier: /var/log, /var/mail, /var/spool
- Attention: grandit!

/home

- Répertoire contenant les homes des utilisateurs normaux

/etc

- Contient toute la configuration du système
- Configuration par utilisateur contenue dans des fichiers cachés de la home des utilisateurs

Répertoires spécifiques

- /mnt ou /media ou /volumes
- /proc et /sys
- /dev

Automatisation

- `/etc/fstab`

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>    <type>    <options>            <dump>    <pass>
proc          /proc      proc      defaults              0          0
/dev/sda2     /          ext3      defaults,errors=remount-ro 0          1
/dev/sda3     /home      ext3      defaults              0          2
/dev/sda1     none       swap      sw                   0          0
/dev/scd0     /media/cdrom0 udf,iso9660 user,noauto          0          0
/dev/fd0      /media/floppy0 auto      rw,user,noauto        0          0
```

Commandes utiles

- `ls -la`
- `df`
- `du`
- `fuser`
- `lsof`

Types de fichiers

- Pas vraiment de type sous Unix
 - Réguliers
 - Exécutables
 - Spéciaux
- Notion flexible d'extension
- Programme “file”

Fichiers et répertoires particuliers

- . et .. représentent le répertoire courant et parent
- Tout fichier ou répertoire commençant par . est invisible (utiliser ls -la)

Liens

- Symbolic link et Hard link
- Refcount
- Utiles et dangereux
 - Récursivité
 - Espace disque

Permissions

- Droit d'accès à un fichier, pour le propriétaire, son groupe, et le reste du monde
- Trois types
 - read
 - write
 - execute

Manipulation des permissions

- Affichage : `ls -l`
- `chmod`
- `chown`

Permissions par défaut

- umask
 - peut être changé dans le profil
 - umask -S : plus lisible
 - danger!

Permissions particulières

- execute sur directory : liste des fichiers
- setuid bit
 - Exécuter un programme comme si il avait été lancé par un autre utilisateur
- sticky bit
 - Empêche la suppression de fichiers dans un répertoire

Permissions particulières

- Possibilité de forcer la création des fichiers dans un répertoire sous un groupe particulier, ...
- **ATTENTION:** ces subtilités varient beaucoup d'une plate-forme à l'autre

Filesystems classiques

- De UFS vers ext2, ext3, ext4
- NFS
- FAT32 / NTFS sous Windows
- HFS/HFS+ sous Mac

Filesystems classiques

- Tout n'est pas disponible sur les différents filesystems
 - Gestion des droits sous FAT32
 - Liens symboliques limités (FAT32, NTFS)
- Simulation au besoin des droits de base

Version étendue des permissions

- ACL: access control lists: beaucoup plus flexibles, utilisés par exemple sur HFS+
- Pas toujours clair
chmod a+rw ne permet pas toujours d'accéder à un fichier!

Démo

- Droits étendus: `ls -la@`

Partie 3: Services et daemons

Concept

- **Au sens strict: programmes tournant en arrière plan**
- **En pratique: tout ce qui est lancé au boot**

Notion de runlevel

- “Mode” du système d’exploitation
 - Sans réseau
 - Console
 - Graphique
 - Modes spéciaux: reboot
 - Plus réellement utilisé sous Linux

/etc/init.d

- Contient des scripts
- Liens par runlevel: /etc/rcX.d
 - La numérotation indique l'ordre
 - Pas de lien = pas lancé automatiquement

Utilisation directe

- `/etc/init.d/networking`
 - `start`
 - `stop`
 - `restart`

Partie 4: Shells

Shells

- Interprétation de commandes
- Nombreux shells différents, le plus utilisé est bash, Bourne Again Shell (standard sur de nombreux Linux et Mac OS X)
- sh : Bourne shell, base de la plupart des shells modernes. Simpliste.
- csh, C shell: pour la syntaxe plus “C”
- Korn shell, TC shell, ... tous ont leurs spécificités

Fonctionnalités

- Historique
- Auto-complétion
- Gestion des scripts
- Langage de programmation interprété de base
- Gestion des variables d'environnement

Variables d'environnement

- Variables au sens “programmation”, locales ou globales
- Dépendantes du shell
- Accessibles par les programmes
- PATH, USER, HOME, PWD, HOSTNAME,...

Variables locales/ globales

- `export MAVARIABLE`
- `set` retourne toutes les variables connues dans le shell courant

Fichiers lus au lancement

- `/etc/profile` et `/home/seb/.profile`
- `/etc/bashrc` et `/home/seb/.bashrc`

Commandes internes

- **Alias**

`alias ll='ls -al'`

- **cd**

- **echo**

- **... (voir documentation de bash)**

Partie 5: Processus

Processus

- Tout programme exécuté est découpé en une série de processus
- ps liste les processus en mémoire
- top fait la même chose de façon interactive

Signaux

- Un signal est un message envoyé à un processus
- Message : un chiffre
 - 1 HUP (hang up)
 - 2 INT (interrupt)
 - 3 QUIT (quit)
 - 6 ABRT (abort)
 - 9 KILL (non-catchable, non-ignorable kill)
 - 14 ALRM (alarm clock)
 - 15 TERM (software termination signal)

kill et killall

- Envoyent un signal au processus pour l'arrêter
- kill a besoin d'un numéro de processus
- killall a besoin d'un label

Détournement des signaux

- Signaux parfois utilisés pour “réveiller” un programme
- HUP
- Masque

Priorités

- Chaque processus s'exécute avec un niveau de priorité
 - Possibilité de réaliser des logiciels d'arrière plan
 - Certains logiciels sont critiques
- nice et renice
 - -20 à +20
 - Inversé!

Contrôle du processus via le shell

- CTRL-C, CTRL-Z
- bg
- fg
- jobs

Démo

- ps
- top
- kill
- killall
- CTRL-C -Z, bg, fg