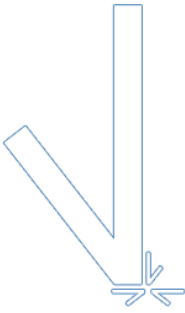


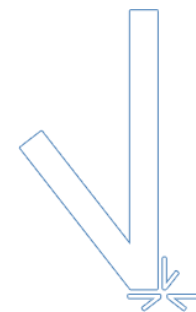
Systemes d'Exploitation



Objectif :

Comprendre comment un système
d'exploitation fonctionne et
comment l'utiliser

Systemes d'Exploitation



Shell et Utilitaires

(cours précédent : Systemes d'Exploitation – Permissions et Droits)

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires



I. Système d'Exploitation

1. Sommaire
2. Introduction
3. Noyau et Pilotes
4. Utilisateurs et Sessions
5. Système de Fichier
6. Permissions et Droits
7. **Shell et Utilitaires**
 1. Le Shell
 2. Utilitaires Système
8. Gestion de la Mémoire
9. Programmes et Processus
10. Variables d'Environnement
11. Scripts Shell
12. Gestion des Paquets

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell

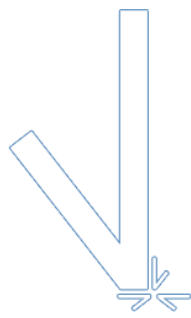


Le Shell

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell



- L'Interface de Ligne de Commande (CLI)
UNIX est appelée `shell`.
 - Le nom de la version du `shell` pour Linux est `BASH`.
- Le `shell` est un CLI qui agit comme une interface entre l'utilisateur et le système d'exploitation.
- L'accès au `shell` nécessite d'être **authentifié** et donc de démarrer une **nouvelle session utilisateur**.

```
pcnet32.c:v1.27a 10.02.2002 tsbogend@alpha.franken.de
pcnet32: PCnet/PCI II 79C970A at 0x10e0, 00 0c 29 43 b8 26 assigned IRQ 18.
eth0: registered as PCnet/PCI II 79C970A
pcnet32: PCnet/PCI II 79C970A at 0x1400, 00 0c 29 43 b8 30 assigned IRQ 19.
eth1: registered as PCnet/PCI II 79C970A
pcnet32: 2 cards found.
Starting interface eth0 [ OK ]
Starting interface eth1 [ OK ]
NET4: Ethernet Bridge 008 for NET4.0
Bridge firewalling registered
eth0: Promiscuous mode enabled.
device eth0 entered promiscuous mode
eth1: Promiscuous mode enabled.
device eth1 entered promiscuous mode
Starting interface br0br0: port 2(eth1) entering learning state
br0: port 1(eth0) entering learning state
Enable Routing [ OK ]
Disabling ECN [ OK ]
Starting WLAN Devices: [ OK ]
Starting Firewall [ OK ]
Starting system log daemon [ OK ]
Starting kernel log daemon [ OK ]
Starting hotplug modules
Setting GRSecurity parameters: [ OK ]

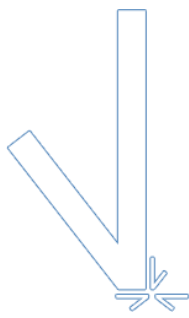
Welcome to Devil-Linux v1.1.1-2003-12-31-i586-SMP - Kernel 2.4.23-grsec (tty1).
Devil login: █
```

A shell login screen

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell



- Pour savoir qui est l'utilisateur actuel qui utilise le shell :

whoami

- Pour savoir qui sont les utilisateurs qui utilisent actuellement le shell :

who

- Pour exécuter plusieurs programmes sur la même ligne :

whoami ; who ;

(L'utilisation du ; permet l'exécution de plusieurs programmes sur la même ligne, indépendamment de leur échec ou de leur succès)

- Pour exécuter plusieurs programmes de manière dépendante :

whoami && who

(L'utilisation de l'opérateur && ne permet d'exécuter le second programme que si le premier réussit)

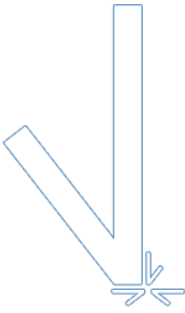
whoami || who

(L'utilisation de l'opérateur || permet d'exécuter le second programme uniquement si le premier échoue)

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell

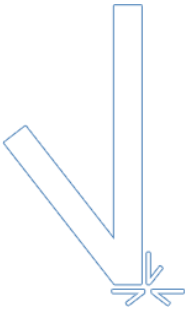


- Lorsque des programmes sont exécutés, ils effectuent des **appels système** au noyau.
- Ces programmes ont des valeurs en **entrée** (paramètres fournis au moment de l'exécution, par l'utilisateur par exemple) et lorsque leur exécution se termine, ils fournissent des valeurs en **sortie** au noyau.
- Les entrées peuvent être des :
 - Entrées standard (**Standard inputs**): `stdin`
- Les sorties peuvent être:
 - Sorties standard (**Standard outputs**): `stdout`
 - Sorties standard d'erreurs (**Standard Error outputs**): `stderr`

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell

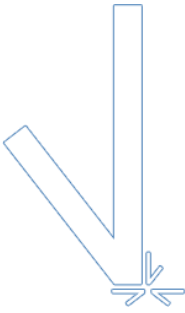


- Par défaut, le noyau **redirige** la sortie vers l'**affichage** et donc le `shell`.
- Cependant, lors de l'exécution d'un programme sous le `shell`, nous pouvons spécifier un comportement différent...

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell



- Pour **rediriger** la **sortie standard** vers un fichier :

who 1>fichier

- Pour **rediriger** la sortie d'**erreur** vers un fichier :

who 2>fichier

- Pour **fusionner** la sortie d'erreur dans l'**entrée standard** :

who 2>&1

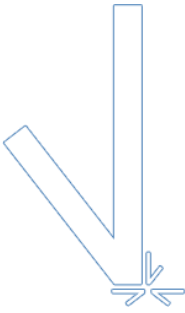
- Pour **fusionner** la sortie d'erreur dans l'entrée standard et **rediriger** la sortie standard vers un fichier :

who 1>fichier 2>&1

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.1 Le Shell



- Enfin, le noyau peut envoyer directement une sortie standard de processus (stdout) **dans** une autre entrée standard de processus (stdin).
- Ce comportement peut être déclenché au niveau du `shell` à l'aide du symbole de barre verticale (**pipe** `|`).
- Par exemple, pour que le noyau envoie la sortie standard de la commande `pwd` directement dans l'entrée standard de la commande `ls` :

`pwd | ls`



Utilitaires Système

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.2 Utilitaires Système

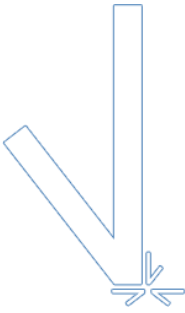


- Divers utilitaires (« commandes ») que nous avons vus précédemment sont des programmes situés sous le :
 - le dossier `/sbin` pour les binaires systèmes destinés à l'administrateur système;
 - le dossier `/bin` pour les autres binaires destinés à tous les utilisateurs

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.2 Utilitaires Système



- La commande la **plus importante** des Systèmes d'Exploitation UNIX est :

man

- Elle se trouve dans le dossier `/bin`. Elle fournit en sortie le manuel d'une autre commande.
- Par exemple:

man ls

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.2 Utilitaires Système

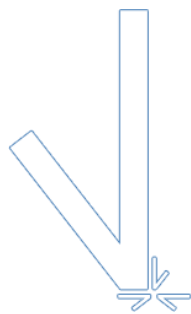


- Parmi les commandes les plus connues, on peut citer :
 - **cat** : Pour lire le contenu d'un fichier
 - **cut** : Pour filtrer le contenu d'un fichier
 - **sort** : Pour trier un fichier
 - **more** : Pour lire un fichier par page
 - **less** : Pour lire un fichier par page
 - **tail** : Pour lire les dernières lignes d'un fichier
 - **head** : Pour lire les premières lignes du fichier
 - **echo** : Pour générer un texte
 - ...

I. Systèmes d'Exploitation

I.7. Shell et Utilitaires

I.7.2 Utilitaires Système



- Apprenons à lire le manuel. Par exemple:

```
[root@VM_0_15_centos ~]# man cp
CP(1) User Commands CP(1)

NAME
    cp - copy files and directories

SYNOPSIS
    cp [OPTION]... [-I] SOURCE DEST
    cp [OPTION]... SOURCE... DIRECTORY
    cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
    Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --archive
        same as -dR --preserve=all

    --attributes-only
        don't copy the file data, just the attributes

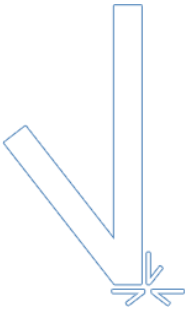
    --backup[=CONTROL]
        make a backup of each existing destination file

    -b
        like --backup but does not accept an argument

    --copy-contents
        copy contents of special files when recursive
```

https://blog.csdn.net/qq_40644291

L'entrée de manuel de la commande cp



Gestion de la Mémoire

(voir cours suivant : Systèmes d'Exploitation – Gestion de la Mémoire)