

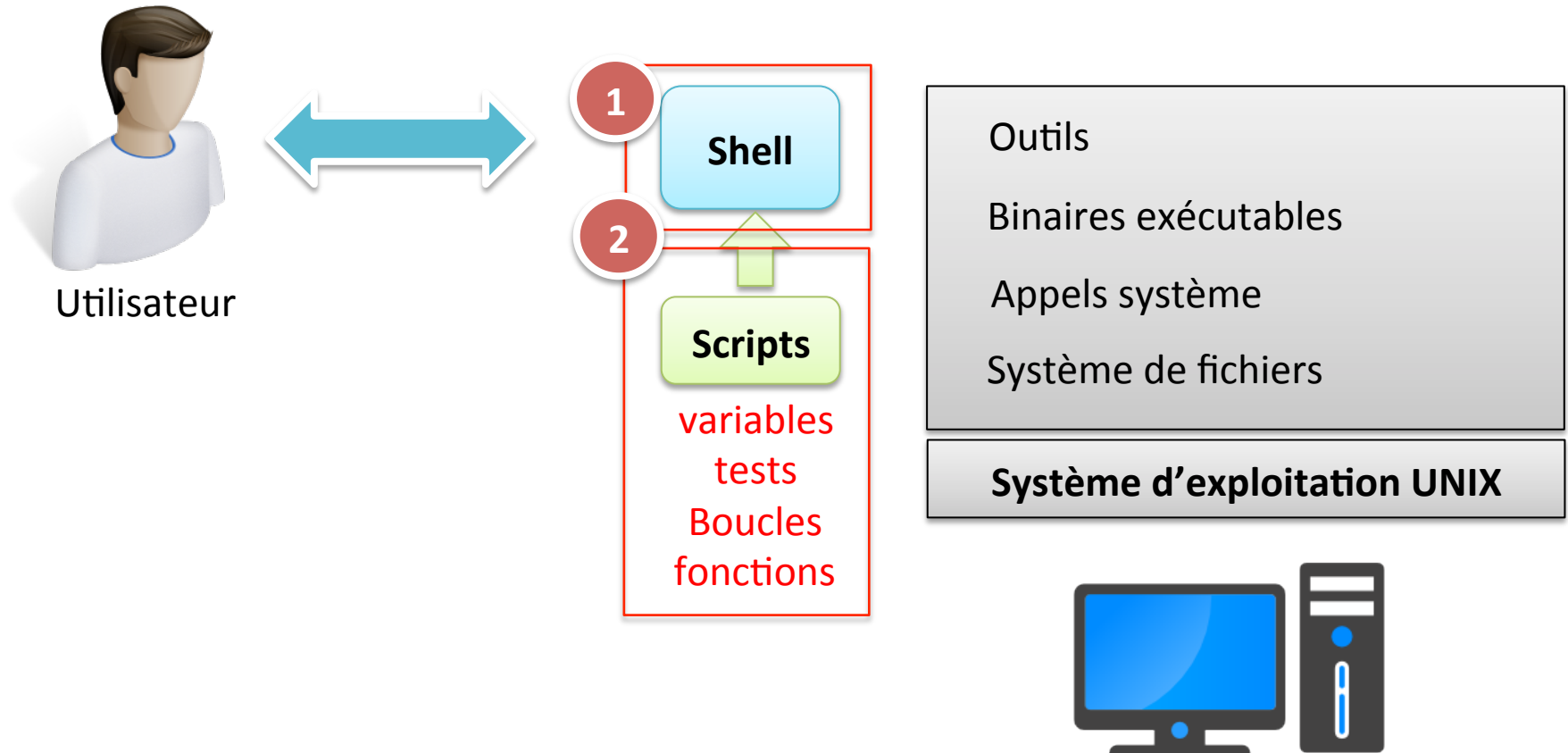


# CS230 - Programmation shell sous Unix

## Cours 2 – Quelques commandes de base

Dr. Bassem DEBBABI

# Objectif du cours 2



# Plan du cours 2

---

- ▶ Manipulation de fichiers
- ▶ Les filtres
- ▶ Éditeur de texte en mode texte
- ▶ Protections sur les fichiers

# Plan du cours 2

---

- ▶ Manipulation de fichiers
- ▶ Les filtres
- ▶ Éditeur de texte en mode texte
- ▶ Protections sur les fichiers

# Manipulation de fichiers

## ► Commande « ls »

Syntaxe \$ `ls [-liadR] [fichier...]`

Liste le contenu d'un répertoire.

- La commande "ls" sans argument, liste les noms de fichiers (ou de répertoires) présents dans le répertoire courant.
- Cette commande, utilisée avec un nom de fichier comme argument, permettra de vérifier l'existence de celui-ci.
- Si l'argument utilisé est un nom de répertoire, "ls" en listera le contenu.
- Il existe de très nombreuses options pour la commande "ls".

-l	affiche le type de fichier, les protections, le nombre de liens avec le fichier, le propriétaire, le groupe, la taille en octets, la date de dernière modification et le nom du fichier.
-i	affiche le i-nœud des fichiers.
-a	liste tous les fichiers y compris les fichiers cachés.
-d	ne descend pas dans un répertoire si le paramètre est un nom de répertoire.
-R	liste les fichiers et les répertoires de façon récursive.

# Manipulation de fichiers

## ► Commandes « pwd » et « cd »

Syntaxe \$ pwd

Affiche le chemin d'accès absolu du répertoire courant.

Syntaxe \$ cd repertoire

Change le répertoire courant.

- pwd = Print Working Directory
- cd = Change Directory

# Manipulation de fichiers

## ► Commandes « mkdir » et « rmdir »

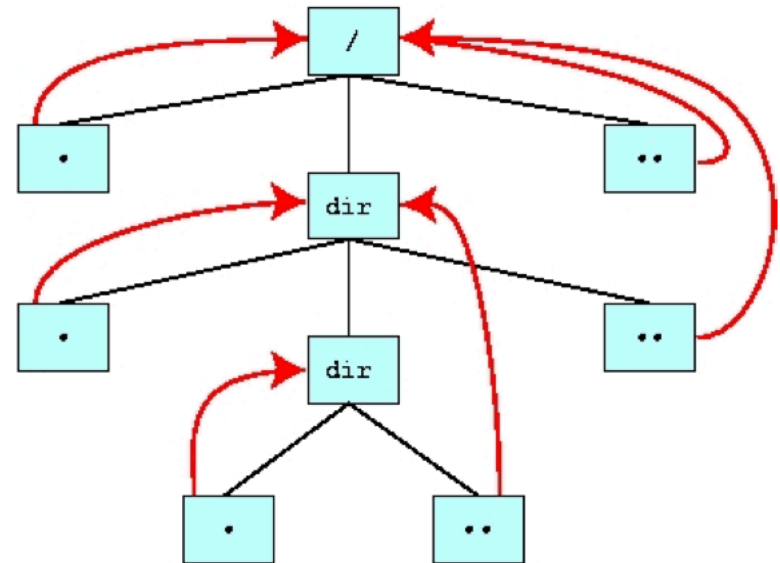
Syntaxe \$ `mkdir [-p] repertoire...`

Créer des répertoires.

Syntaxe \$ `rmdir repertoire...`

Supprimer des répertoire.

- Quand un répertoire est créé, le système génère automatiquement deux sous-répertoires représentant des liens vers le répertoire créé et le répertoire père :
  - "." = répertoire courant,
  - ".." = répertoire père.
- Les répertoires à supprimer doivent impérativement être vides
- L'option `-p` permet de créer des répertoires intermédiaires.



# Manipulation de fichiers

## ► Commandes « touch », « cat » et « more »

Syntaxe \$ touch fichier...

Permet de créer un fichier vide

- Elle modifie la date d'accès et la date de modification de chaque *fichier* indiqué
- Si fichier inexistant, elle permet de créer un fichier vide

Syntaxe \$ cat fichier...

concatène le contenu des fichiers en arguments et affiche le contenu

Syntaxe \$ more fichier...

visualise le contenu des fichiers page écran par page écran.



# Manipulation de fichiers

## ► Commandes « cp », « mv » et « ln »

Syntaxe \$ `cp [-i] fichier... dest`

Copie les fichiers.

### ► `cp fichier1 fichier2`

- effectuer une copie du fichier1 vers fichier2.
- Si fichier2 existait déjà, il est supprimé pour être remplacé par la nouvelle copie.

### ► `cp fichier1 fichier2 repertoire`

- Duplique ces fichiers dans le répertoire spécifié.

Syntaxe \$ `mv [-i] fichier... dest`

renomme les fichiers et répertoires et/ou déplace les fichiers.

- Les fichiers source de la commande “mv” peuvent être de n’importe quel type (fichier, répertoire, etc.).

- Option “-i” permet de demander à l’utilisateur son avis sur le renommage d’un fichier.

Syntaxe \$ `ln [-s] fichier... lien`

crée un lien sur un fichier ou un répertoire.

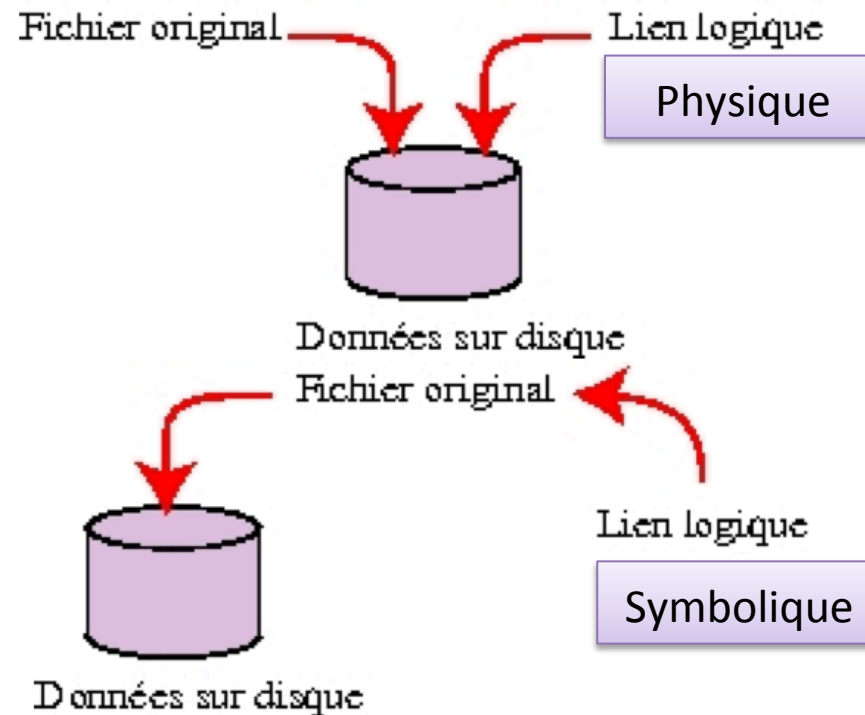
### ► `ln fichier1 fichier2`

- fichier1 qui existe déjà, va pouvoir être accédé via le nouveau nom fichier2.

- Option “-s” permet de créer un lien symbolique.

# Manipulation de fichiers

## ► Lien physique et lien symbolique



# Manipulation de fichiers

## ► Commande « rm »

Syntaxe \$ `rm [-irf] fichier...`

Effacer des fichiers

- Si plusieurs arguments sont fournis à la commande, tous les fichiers spécifiés seront effacés.

-r	(récuratif) indique la récursivité et permet d'effacer un répertoire et tout son contenu.
-i	(interactive) demande une confirmation (y ou n) sur chaque fichier à effacer.
-f	(force) ne fait plus tenir compte à "rm" des protections du fichier, mais uniquement du propriétaire. Vous pouvez donc effacer vos fichiers, même s'ils sont protégés.

# Manipulation de fichiers

## ► Commande « find »

Syntaxe \$ `find repertoire criteres`

Retrouver des fichiers à partir de certains critères

-name	recherche sur le nom du fichier
-perm	recherche sur les droits d'accès du fichier
-user	recherche sur le propriétaire du fichier
-type	recherche sur le type (d=répertoire, c=caractère, f=fichier normal)
-size	recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets)
-atime	recherche par date de dernier accès en lecture du fichier
-mtime	recherche par date de dernière modification du fichier
-ctime	recherche par date de création du fichier

- On peut combiner les critères avec des opérateurs logiques :
  - **critère1 critère2**  
ou **critère1 -a critère2**  
correspond au **et** logique,
  - **! critère** non logique,
  - **\ (critère1 -o critère2\)** ou logique.
- La commande **find** est récursive
- `-exec cmd {} \;`
- Tous les critères définies par un nombre, on peut rajouter les sign +/-
  - `-size +2` # plus que 2
  - `-size -2` # moins que 2

# Manipulation de fichiers

## ► Commande « tar »

Syntaxe \$ `tar [-mode -opts] fichier...`

Creation et manipulation des fichiers d'archives

### modes

-c	Créer une nouvelle archive
-r	Rajouter des fichiers (l'option -f est nécessaire)
-t	Lister le contenu de l'archive
-u	Rajouter des fichiers seulement s'ils étaient changés
-x	Extraire le contenu de l'archive

### options

-v	Afficher les détails de traitement sur la sortie standard
-z	(avec le mode -c seulement) compressé l'archive résultante en utilisant <i>gzip</i>
-f <i>file</i>	Lire ou écrire l'archive sur le fichier spécifié

```
$ tar -czf file.tar.gz source.c source.h
# créer une nouvelle archive nommée « file.tar.gz » contenant
# deux fichiers « source.c » et « source.h »
$ tar -tvf file.tar.gz
# visualiser le contenu de l'archive
$ tar -xvf file.tar.gz
# extraire le contenu de l'archive
```

# Plan du cours 2

---

- ▶ Manipulation de fichiers
- ▶ **Les filtres**
- ▶ Éditeur de texte en mode texte
- ▶ Protections sur les fichiers

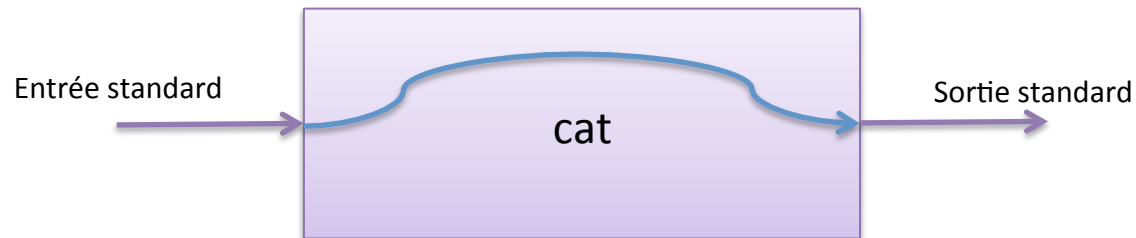


# Les filtres

## ► Commande « cat »

Syntaxe \$ `cat fichier...`

concatène le contenu des fichiers en arguments et affiche le contenu



# Les filtres

## ► Commande « sort »

Syntaxe \$ `sort [-nrtd] fichier...`

permet de trier le contenu d'un ou plusieurs fichiers

-n	le tri se fait numériquement.
-r	inverser le résultat de tri.
-d	ne considérer que les blancs et les caractères alphanumériques.
-t	redéfinir le séparateur

- Les champs sont délimités par défaut par une tabulation ou de façon explicite par le caractère spécifié avec l'option `-t <caractère>`.



# Les filtres

## ► Commande « wc »

Syntaxe \$ `wc [-lwc] [fichier{...}]`

Compte le nombre de lignes, de mots ou de caractères

-l	compte le nombre de lignes
-w	compte le nombre de mots
-c	compte le nombre de caractères

# Les filtres

## ► Commandes « head » et « tail »

Syntaxe \$ `head [-n] fichier...`

permet de lister les n premières lignes

Syntaxe \$ `tail [-n] fichier...`

permet de lister les n dernières lignes

<code>-n num</code>	Par ligne
---------------------	-----------

# Les filtres

## ► Commande « cut »

Syntaxe \$ `cut -cliste [fichier...]`

Extraire des caractères à partir de l'entrée standard

Syntaxe \$ `cut -fliste [-dcaractère] [-s] [fichier...]`

Extraire des champs

- Dans les deux modes, "liste" est une séquence de numéros pour indiquer à cut quels sont les champs ou les colonnes à retenir.
- Il y a plusieurs formats possibles pour cette liste :

"A-B"	champs ou colonnes A à B inclus
"A-"	du champ ou colonne A jusqu'à la fin de la ligne
"A,B"	champ ou colonnes A et B
"-B"	du début jusqu'au champ ou colonne B

► Lors d'un découpage par champs

- -d spécifie le caractère séparateur de champs
  - Par défaut le caractère séparateur est la tabulation « TAB ».
- -s indique à cut d'écarter toutes les lignes qui ne contiennent pas le séparateur

# Manipulation de fichiers

## ► Commande « grep »

Syntaxe \$ `grep [-invE] expression [fichier...]`

Recherche sur le contenu d'un fichier

-i	indique à "grep" qu'il ne faut pas tenir compte des majuscules/minuscules
-n	permet de voir afficher les numéros de lignes courantes
-v	indique à "grep" qu'il faut afficher les lignes ne contenant pas l'expression précisée en argument
-E	forcer grep à utiliser des expressions régulières étendues.

# Manipulation de fichiers

---

## ► Commande « sed »

Syntaxe \$ `sed [-e] 's/ancienMotif/nouveauMotif/g' fichierSource`

Permet, entre autres, de réaliser des substitutions

# Plan du cours 2

---

- ▶ Manipulation de fichiers
- ▶ Les filtres
- ▶ Éditeur de texte en mode texte
- ▶ Protections sur les fichiers

# Éditeur de texte « vi »

---

# Plan du cours 2

---

- ▶ Manipulation de fichiers
- ▶ Les filtres
- ▶ Éditeur de texte en mode texte
- ▶ Protections sur les fichiers



# Protections sur les fichiers

## ► Permissions

Accès	Fichier	Répertoire
<b>Lecture</b>	Le contenu du fichier est visible.	Le contenu du répertoire est visible.
<b>Écriture</b>	Le contenu du fichier peut être modifié.	Le contenu du répertoire peut être modifié.
<b>Exécution</b>	Le fichier peut être utilisé en commande.	Le répertoire peut devenir le répertoire courant.

Niveau	explication
<b>User</b>	Le niveau utilisateur.
<b>Group</b>	Le niveau groupe.
<b>Others</b>	Le niveau autre.

# Protections sur les fichiers

## ► Commande « chmod »

Syntaxe \$ `chmod mode fichier...`

Changement de protection d'un ou plusieurs fichiers.

### ► avec :

- ❑ mode=masque de protections (en octal)
- ❑ ou bien mode=<u|g|o><+|-><r|w|x>

### ► Pour affecter les droits d'accès à un fichier ou à un répertoire :

- ❑ on additionne les autorisations d'accès,
- ❑ on effectue cette opération pour chaque niveau d'accès

Droits d'accès	lecture 2 <sup>2</sup> 4	écriture 2 <sup>1</sup> 2	exécution 2 <sup>0</sup> 1
Abréviation utilisée	r	w	x

Utilisateur			Groupe			Autre		
r	w	x	r	-	x	-	-	-
1	1	1	1	0	1	0	0	0
2 <sup>2</sup> x 1	2 <sup>1</sup> x 1	2 <sup>0</sup> x 1	2 <sup>2</sup> x 1	2 <sup>1</sup> x 0	2 <sup>0</sup> x 1	2 <sup>2</sup> x 0	2 <sup>1</sup> x 0	2 <sup>0</sup> x 0
	7			5			0	