Présentation du système Linux

Système d'exploitation

- Un système d'exploitation (en anglais operating system, souvent abrégé en OS) assure des tâches de liaison entre le matériel, l'utilisateur et les applications (traitement de texte, jeu,..).
- C'est l'interface entre l'utilisateur et le matériel.
- Ses fonctions principales sont:
  - Contrôle des ressources
  - Contrôle des processus
  - Contrôle des périphériques
  - .....
- Il contient des outils de gestion utilisables par les applications, tels que la manipulation de fichiers, gestion d'impressions, date...

Historique d'UNIX

- Unix est né aux laboratoires Bell (Filiale d'ATT) développé en assembleur à partir de 1969 par Ken Thompson et Dennis Ritchie.
- En 1973, Unix est réécrit à 90% en langage C (Créé pour l'occasion par Brian kernighan)
- Actuellement, Unix est un système d'exploitation des stations de travail et des serveurs de base de données (utilisé sur de plus en plus de plateformes grâce au développement des Unix « Libres »
- Fournisseurs : Digital Equipement, Hewlett Packard, IBM, Silicon Graphics, Sun Microsystems + tous les Unix « Libres » (Linux, OpenBSD, FreeBSD,…)
- Concurrents : Windows (Microsoft), Mac OS

Distributions Linux

Archlinux

Knoppix


Ubuntu


Mandriva

Zenwalk


Gentoo


Fedora

Frugalware

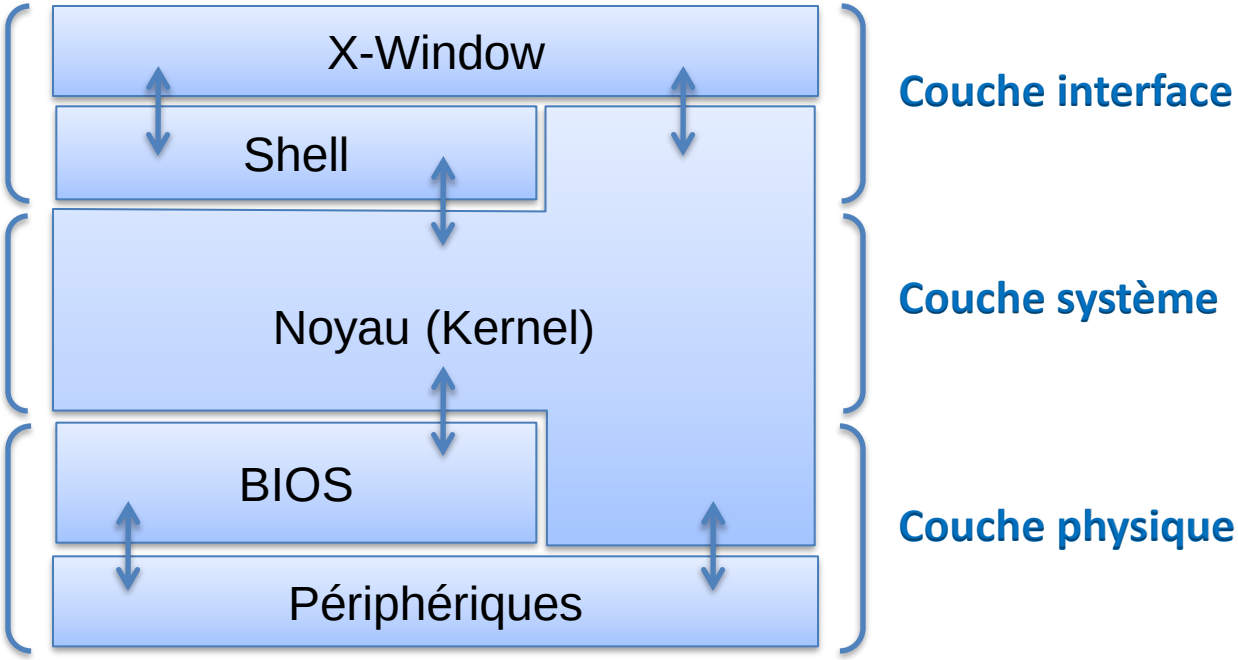
Caractéristiques d'UNIX

UNIX est un Système d'exploitation :

- multi-utilisateurs
- multi-tâches
- multi-plate-formes (c'est-à-dire portable) : IBM, SUN, HP, IRIX (Silicon Graphics)… et maintenant sur PC avec Linux, mais aussi caché dans MacOS.
- qui gère la répartition des ressources (mémoire et espace disque)
- orienté réseau (e.g. partage de fichiers sur une machine distante : NFS…)
- …
- très utilisé en développement et en recherche (le développement d'applications y est SIMPLE)
- très stable
- Devient d'utilisation simple pour tous
- .....

Architecture Linux

- Divisée en 3 couches distinctes
  - ✓ La couche physique : Périphériques et BIOS
  - ✓ La couche système : Gérée par le noyau
  - ✓ La couche interface : le Shell et/ou le système X-Window



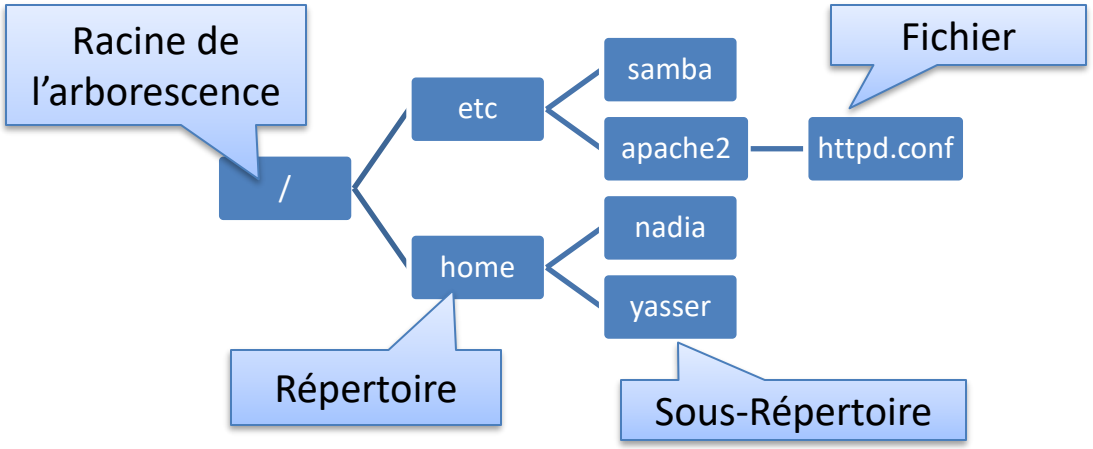
Système de fichier

- En UNIX, tout est fichier :
  - les fichiers dits normaux
    - ✓ fichiers textes (suite de caractères ASCII = caractères lisibles)
    - ✓ fichiers exécutables (suite de caractères binaires, compréhensibles uniquement par l'ordinateur, en général il s'agit des programmes)
  - les répertoires (peuvent contenir d'autres fichiers ou d'autres répertoires)
  - les fichiers de périphériques
- Chaque fichier possède un "i-node" : moyen pour Unix de stocker les caractéristiques du fichier

  (emplacement, nom du propriétaire, droits, taille, date de création et de dernière modification)

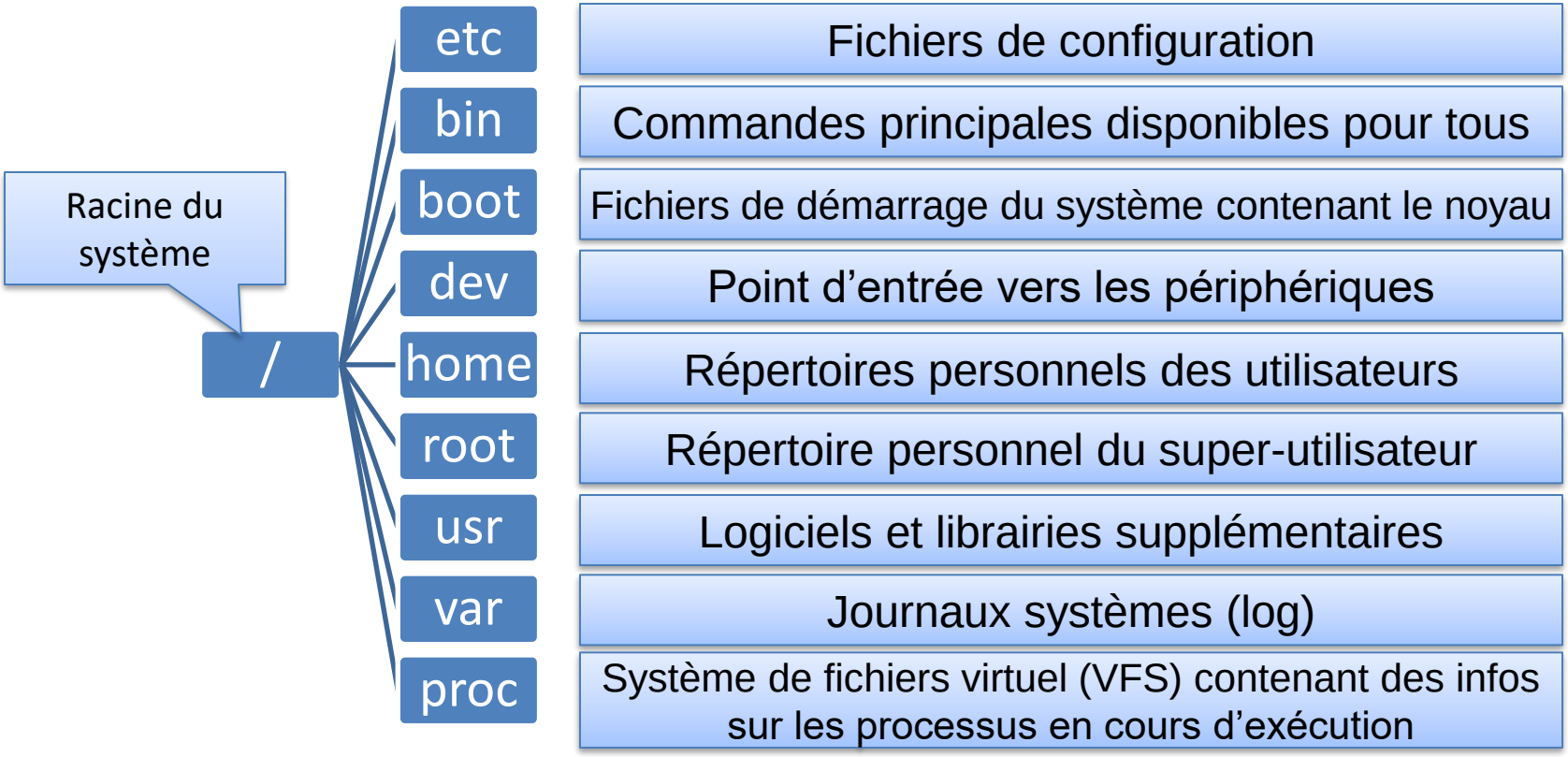
L'arborescence Linux

- Qu'est-ce qu'un système de fichier ?
  - Organisation physique des données sur un support
    - ✓ Sur un disque dur, une clé USB, un DVD, …
- Qu'est-ce qu'une arborescence ?
  - Organisation logique des fichiers sur un ou plusieurs systèmes de fichiers
  - Il s'agit d'une structure de données hiérarchique de type arbre



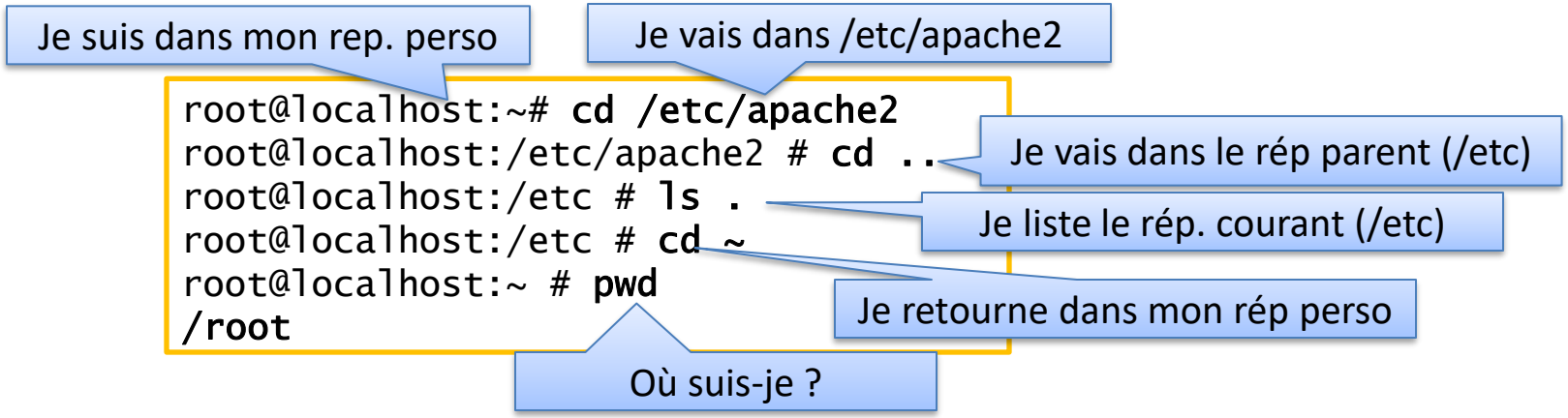
L'arborescence Linux

- Voici l'arborescence typique d'un système Linux :



Les symboles associés à l'arborescence

- Différents symboles sont utilisés pour désigner des répertoires
  - Le « . » : Répertoire courant
  - Le « .. » : Répertoire parent
  - Le « ~ » : Répertoire personnel de l'utilisateur courant
- La commande « cd » permet de changer de répertoire
- La commande « ls » permet de lister un répertoire
- La commande « pwd » permet de connaître le rép. courant
- Exemples :

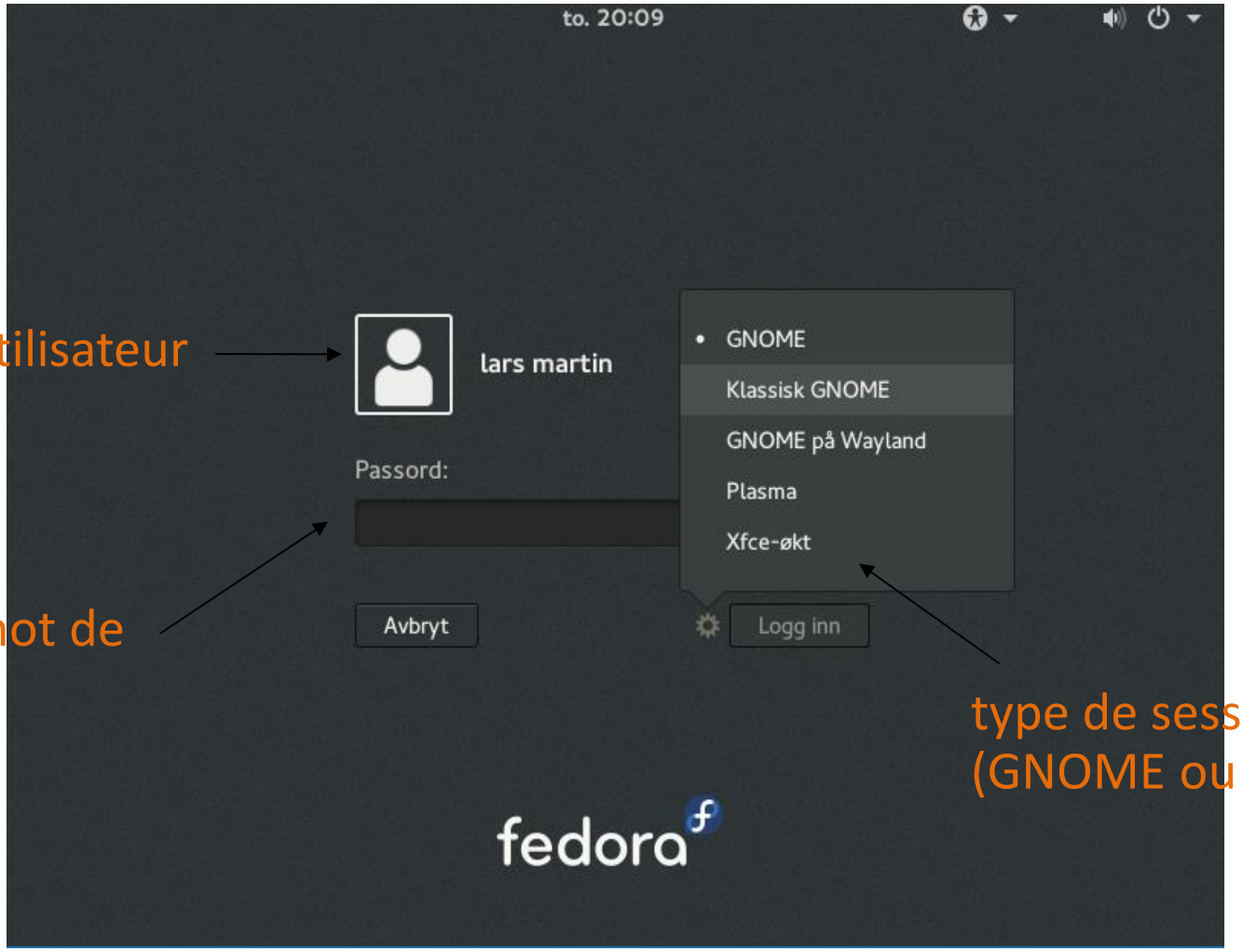


Les Utilisateurs

- Unix est multi-utilisateurs
  - nécessité d'avoir un nom particulier aux yeux du système ("login") ainsi qu'un mot de passe pour la sécurité
- 2 types d'utilisateurs
  - l'utilisateur "root" : super utilisateur -> possède tous les droits sur la machine
  - les autres utilisateurs (possèdent des droits restreints)
- Chaque utilisateur est caractérisé par :
  - un nom ("login") et un numéro d'utilisateur ("UID")
  - un groupe et un numéro de groupe ("GID")
  - un mot de passe ("passwd")
  - un type de shell (= intérpréteur de commandes)
  - un répertoire utilisateur qui lui appartient ("home directory")

(Rq : toutes ces informations sont stockées dans le fichier /etc/passwd)

Première Connexion



La session d'utilisateur

Saisie du mot de passe

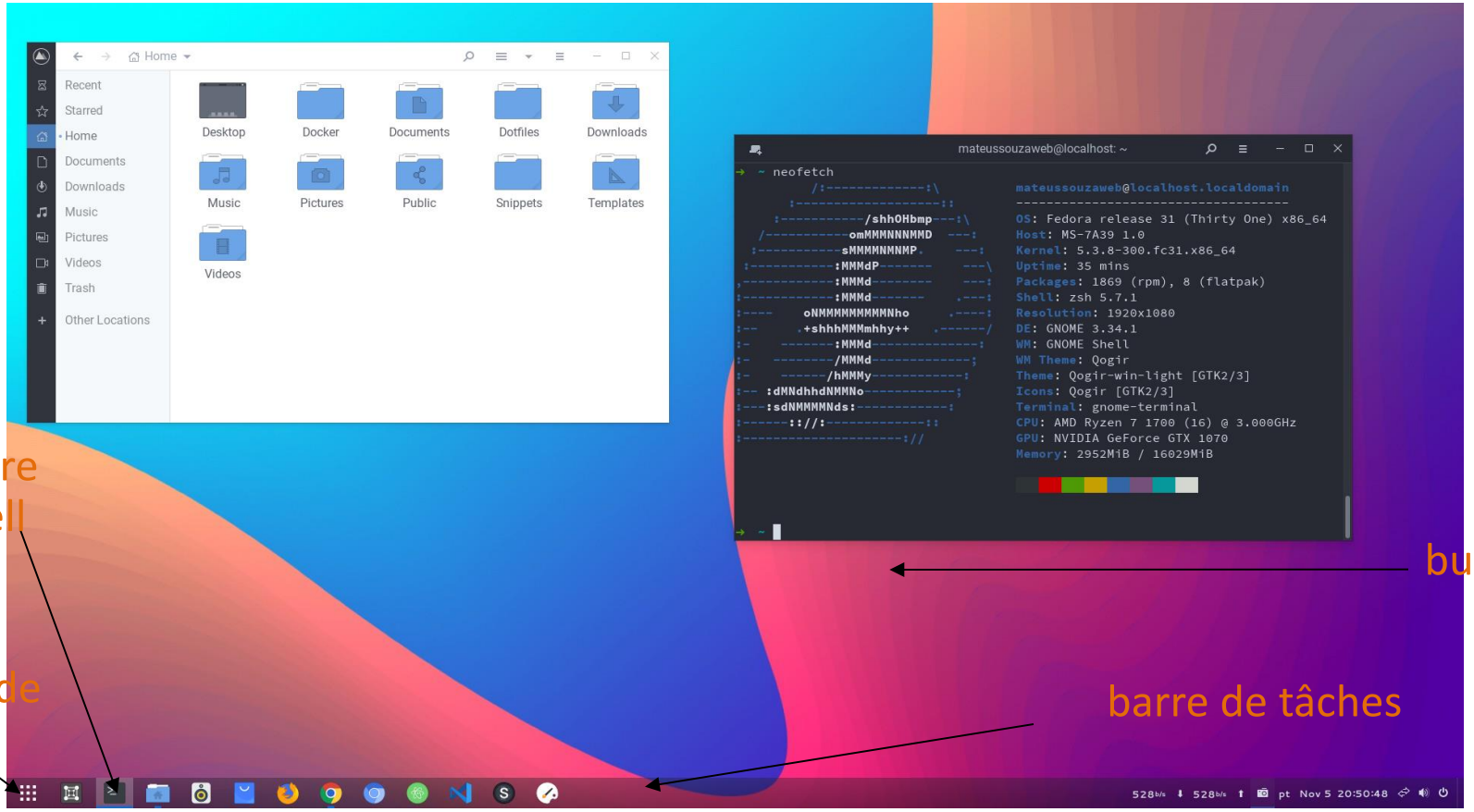
type de session X (GNOME ou KDE)

Mot de passe

- le mot de passe est personnel :
  - ne pas le divulguer
  - éviter les mots de passe triviaux (e.g. votre nom…)
  - utiliser des lettres (majuscules et/ou minuscules) et des chiffres (éventuellement des caractères spéciaux)
  - doit faire 8 lettres au minimum

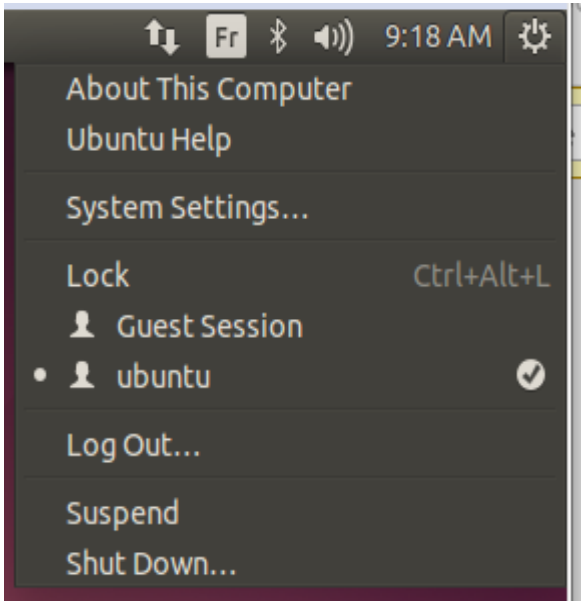
L'environnement X-Windows

- On arrive sous l'environnement X-Windows (KDE ou gnome = interface graphique rendant l'interaction avec le système plus conviviale) :



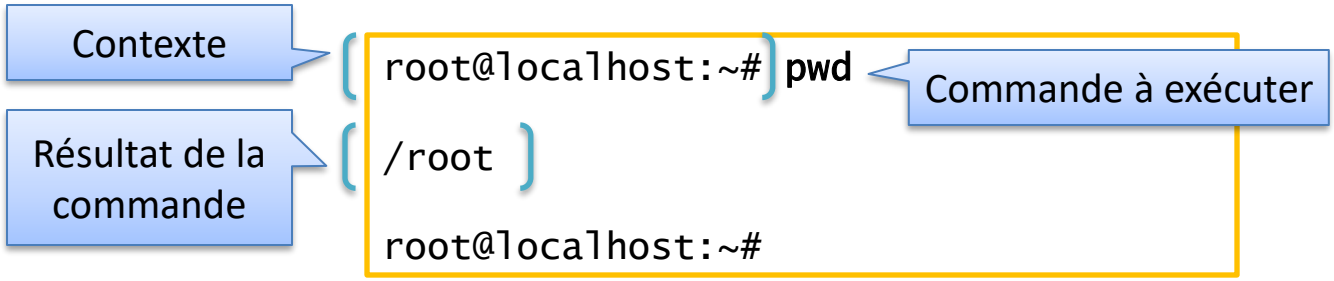
Déconnexion de la session X-Windows

- Nécessité de se déconnecter à la fin d'une session (pour des raisons de sécurité)



Notion de shell

- Le shell est un interpréteur de commandes
  - Permet à l'utilisateur d'interagir avec le système
  - Il lit et exécute les commandes de l'utilisateur
- C'est aussi un véritable langage de programmation
  - Il sera possible d'écrire des scripts exécutant des commandes répétitives
- Il en existe plusieurs
  - Le shell « bash » est le plus courant
  - Mais aussi les shells « csh », « ksh », « tcsh »



Notion de shell (2)

- Ouverture du shell (sous X-Window) :
  - cliquer sur l'icône représentant le shell, ou sélectionner «ouvrir un terminal» dans le menu droit de la souris
- A ce point le shell peut recevoir des commandes :
  - exemples :
    - ✓ date : affiche la date
    - ✓ ls : liste les fichiers du répertoire courant
- Fermeture du shell :
  - commande exit
  - commande logout
  - Ctrl-D

Syntaxe générale des commandes UNIX

- syntaxe générale :

  $ commande options arguments

Exemple : ls -al

les options (souvent très nombreuses) permettent de modifier le comportement de la commande; en général elles sont précédées du signe '-' (e.g. ls -l)

Certaines commandes utilisent des arguments (e.g. nom de fichier) il y a un manuel en ligne: **man ls**

- IMPORTANT :
  - Unix est sensible à la casse **(a != A): ls != LS** ou de **Ls**
  - Unix utilise l'espace comme séparateur de commandes (e.g. utiliser man date et non mandate)

Quelques commandes de base

- **more fichier :** affiche le contenu de fichier page par page
  - utiliser la touche espace pour passer à la page suivante
  - utiliser la touche b pour revenir à la page précédente
  - utiliser la touche q pour quitter
- **ls :** affiche la liste des fichiers
- **cd :** change de répertoire
- **rm :** supprime un fichier
- **mv :** Change le nom d'un fichier, déplace un fichier
- **vi :** édite un fichier
- **man commande** : affiche les pages de manuel de commande (utilise les mêmes touches que more pour se déplacer le long des pages)
- **date :** affiche la date

Quelques commandes de base (2)

pwd : Affiche le répertoire courant
mkdir : Crée un répertoire
rmdir : Supprime un répertoire
cp : Copie des fichiers dans un répertoire
du : Affiche la taille d'une arborescence
find : Recherche de fichier dans une arborescence

La commande ls

- Syntaxe

  $ ls [Option…] [(Chemin | Fichier)…]

- Principales options
  - ls -m : Affiche les fichiers en les séparant par une virgule au lieu de les présenter en colonnes.
  - ls -t : Affiche les fichiers par date, c'est-à-dire en les classant du récent au plus ancien.
  - ls -lu : Affiche les fichiers par date de dernier accès et indique cette date.
  - ls -F : Affiche les fichiers par type. Ainsi un fichier suivi d'un slash (/) est un répertoire, un fichier suivi d'une étoile est un fichier exécutable et un fichier suivi d'un "@" est un lien (nous reviendrons sur les liens dans la section consacrée à ln).
  - ls -S : Affiche les fichiers triés par ordre de taille décroissante.
  - ls -X : Affiche les fichiers par type d'extension.
  - ls -r : Affiche les fichier en ordre alphabétique inverse.

La commande cp

- Syntaxe

  $ cp [Option…] Fichier1 Rep_Dest

- Principales options
  - cp -i: Avertit de l'existence d'un fichier du même nom et demande s'il peut le remplacer ou non.
  - cp -l: Permet de faire un lien en "dur" entre le fichier source et sa copie
  - cp -s: Permet de faire un lien "symbolique" entre le fichier source et sa copie
  - cp -p: Permet lors de la copie de préserver toutes les informations concernant le fichier.
  - cp -r: Permet de copier de manière récursive l'ensemble d'un répertoire et de ses sous répertoires

La commande mv

- Syntaxe

$ mv [Option…] Fichier1 Répertoire

  - mv -b: Va effectuer une sauvegarde des fichiers avant de les déplacer
  - mv -i: Demande pour chaque fichier et chaque répertoire s'il peut ou non le déplacer
  - mv -u: Demande a "mv" de ne pas supprimer le fichier si la date de modification est la même ou plus récente que son remplaçant.

La commande rm

- Syntaxe

  $ rm [Option…] Fichier1 Fichier2

- Principales options
  - rm -i : demander à l'utilisateur la confirmation avant la suppression des fichiers.
  - rm -r: Permet de supprimer un répertoire et ses sous répertoires.
  - rm -f: Permet de supprimer les fichiers protégés en écriture et répertoires sans confirmation.

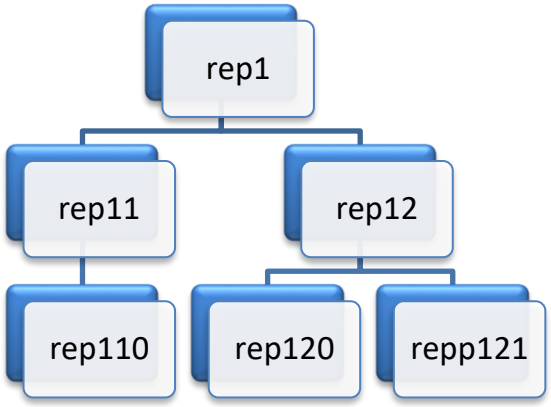
La commande mkdir

- Syntaxe

  $ mkdir répertoire ….

  La commande mkdir crée les répertoires qui sont fournis en argument. Les sous répertoires qui peuvent figurer dans le chemin d'accès à un répertoire à créer doivent exister, sinon la commande échoue.

  $ mkdir –p répertoire/sous-répertoire …..



La commande rmdir

- Syntaxe

  $ rmdir [-p] répertoire ….

  La commande rmdir supprime les répertoires qui sont fournis en arguments , à condition qu'ils soient vides. Avec l'option –p, les sous répertoires qui peuvent figurer dans le chemin d'accès à un répertoire sont eux aussi détruits s'ils sont vides, après que le répertoire terminal ait été détruit.

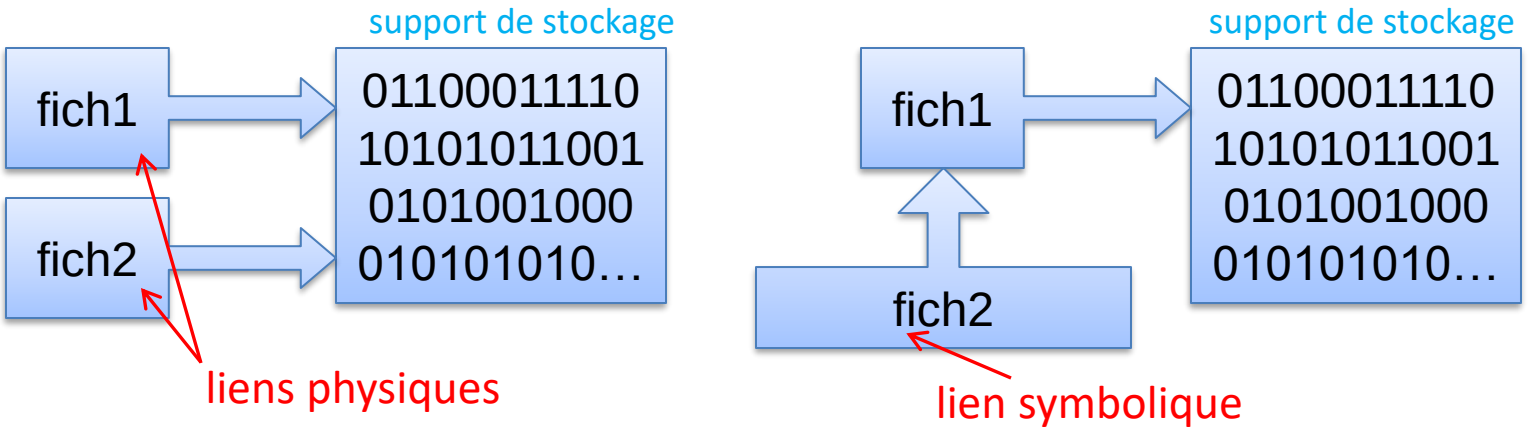
La commande ln

- Syntaxe

  $ ln [-s] fichier 1 fichier2

  La commande ln permet de créer des entrées multiples dans l'arborescence d'un système de fichiers pour un même fichier physique. Ce qui revient à dire que si l'on modifier, ses liens le sont aussi.

- L'option –s permet de faire des lien symbolique
- Un petit dessin :



Caractères spéciaux

- Certains caractères ont une signification particulière
  - Interprétés par le shell
- Astérisque ou étoile : *
  - Interprété comme toute suite de caractères alphanumérique
  - Exemple : Effacer tous les fichiers commençant par « rapport »

```
rm rapport*
```

- Point d'interrogation : ?
  - Interprété comme un seul caractère alphanumérique
  - Exemple : Effacer certains fichiers commençant par « rapport?.doc »

```
rm rapport?.doc
```

  - « rapport1.doc » sera effacé mais pas « rapport12.doc »
- Point virgule : ;
  - Séparateur de commandes

```
cp bilan.txt bilan2007.txt ; rm bilan.txt
```

Caractères spéciaux (2)

- Les crochets : []
  - Remplace un caractère choisi parmi ceux énumérés entre les crochets
  - Exemple : Effacer les fichiers dont la 1ère lettre est « a » ou « b » et se terminant par « .txt »

```
rm [ab]*.txt
```

    - « args1.txt » et « bilan.txt » seront effacés mais pas « comment.txt »
  - Exemple : Effacer les fichiers numérotés de 10 à 29
    - « rapport12.txt » mais pas « rapport3.txt »

```
rm rapport[12][0-9].txt
```

- L'espace
  - Utilisé comme séparateur de paramètres pour une commande
  - Exemple : Effacement de 2 fichiers passés en paramètres

```
rm rapport.doc rapport2008.txt
```