

# R1.04 – Cours 1

Introduction, OS, Interfaces graphiques, Shell

Département Informatique

IUT2, UGA

2023/2024

# Plan du cours

- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2
- 3 OS et types d'OS
- 4 Interfaces graphiques
- 5 Langage de commande (*shell*)
- 6 Résumé

# Plan du cours

- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2
- 3 OS et types d'OS
- 4 Interfaces graphiques
- 5 Langage de commande (*shell*)
- 6 Résumé

# Objectifs de la ressource R1.04 et de la SAÉ S1.03

Comprendre le rôle, les composants et le fonctionnement d'un système d'exploitation

- Caractéristiques, fonctionnalités et types d'un système d'exploitation
- Manipuler efficacement une interface graphique
- Utiliser un langage de commande
- Connaître les commandes les plus utiles
- Processus, fichiers, utilisateurs
- Réseau de l'IUT2 : architecture, protocoles utilisés
- Applications réseau indispensables
  - transferts et partages de fichiers
  - utilisation de machines à distance, ...
  - ...
- Installer un système sur une machine et le personnaliser

Mise en œuvre avec Linux

# Modalités

- 6 semaines pour la ressource R1.04
  - cours : promo entière
  - TP : par 1/2 groupes, 1 étudiant par machine
- 1 semaine bloquée pour la SAÉ S1.03
  - cours
  - ateliers d'installation de Linux et d'outils de développement

# Informations diverses

- Documents fournis
  - Transparents de cours
  - Sujets de TP
  - Disponibles en ligne sur l'intranet
- Bibliographie
  - Cf. intranet
  - Documentation intégrée au système, intranet, Web
- Évaluation
  - R1.04 : Mini QCM au début de chaque TP, contrôle sur machine
  - S1.03 : rendus en ligne (Chamilo)
- Enseignants

# Plan du cours

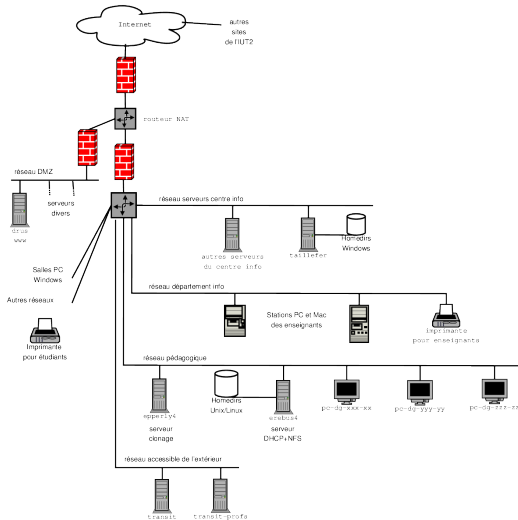
- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2**
- 3 OS et types d'OS
- 4 Interfaces graphiques
- 5 Langage de commande (*shell*)
- 6 Résumé

# Machines accessibles aux étudiants

- Stations de travail sous Linux
  - ~ 90 machines, 6 salles
  - réservées au département info
- PC sous Windows
  - ~ 135 machines, 9 salles
  - pour tous les départements
- Ressources accessibles via le réseau
  - imprimantes
  - serveurs (fichiers, Web, intranets, SSH, SGBD, ...)
- La plupart des enseignements sont faits sur les stations Linux (programmation, bases de données, Web, ...)



# Le réseau de l'IUT2



# Comptes informatiques

- Compte = nom de login + mot de passe
- Respect de la **charte**
- Compte nécessaire pour utiliser toutes les ressources informatiques de l'UGA et de l'IUT2
  - postes de travail
  - messagerie
  - intranets
  - autres serveurs
  - ...

# Espaces de stockage

Pour chaque étudiant

- répertoire personnel commun à toutes les stations Linux,  
**limité à 1.5 Go**
- attention : toutes vos **données** seront **effacées** en fin d'année  
(date exacte sur l'intranet, à noter sur vos agendas)
- répertoire personnel commun à tous les PC sous Windows,  
limité en taille
- une boîte de messagerie, limitée en taille
- —→ **ménage régulier nécessaire**

# Accès depuis l'extérieur de l'IUT2

- Services disponibles depuis tout ordinateur connecté à Internet
  - accès aux stations Linux
  - transferts de fichiers
  - messagerie
  - intranets
- Accès sécurisés par
  - protocoles SSH, SSL / TLS
  - login/mot de passe
- Méthodes et outils d'accès depuis l'extérieur présentés lors d'un prochain cours

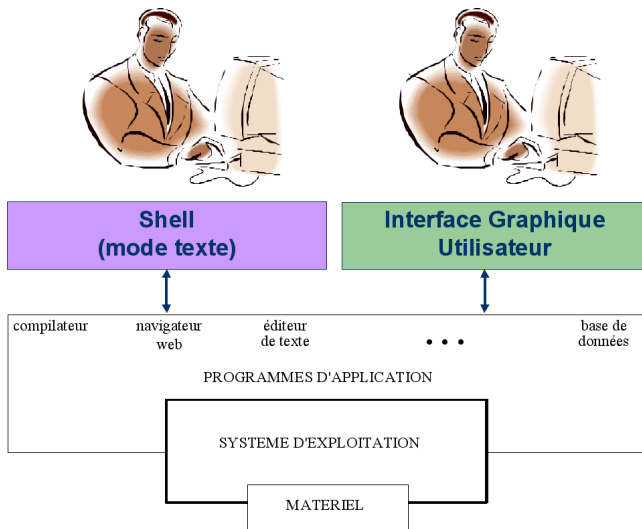
# Plan du cours

- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2
- 3 OS et types d'OS**
- 4 Interfaces graphiques
- 5 Langage de commande (*shell*)
- 6 Résumé

# Rôle du système d'exploitation

- Abréviations
  - Système d'exploitation : SE
  - *Operating System* : OS
- Le matériel fournit différentes ressources
  - calcul : CPU (processeur), GPU (processeur graphique)
  - mémoire : RAM (mémoire vive)
  - périphériques de stockage :  
SSD (mémoire flash), HDD (disque dur)
  - autres périphériques : clavier, souris, audio, réseau, ...
- Les programmes d'application (logiciels) utilisent ces ressources à travers le SE
- Le SE contrôle et partage l'emploi des ressources parmi les programmes d'application et les différents utilisateurs

# Système d'exploitation



# Types d'OS

- Par type de machine
  - Ordinateur (*desktop*, *laptop*) : Linux, MacOS, Windows, ...
  - Mobile (téléphone, tablette) : Android, iOS, ...
  - autres :
    - très grosses machines (super-calculateurs)
    - très petites machines (informatique embarquée, IoT)
- Par usage
  - Poste de travail :  
Linux avec interface graphique, Windows (10, 11), MacOS
  - Serveur :  
Linux sans interface graphique, Windows Server (2016, 2022), Darwin, ...
- Par licence d'utilisation
  - OS libres : Linux, (Android), ...
  - OS propriétaires : Windows, MacOS/iOS, ...
- OS de niche
  - OS temps réel
  - OS distribués
  - OS pour équipements réseau
  - ...



# Plan du cours

- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2
- 3 OS et types d'OS
- 4 Interfaces graphiques**
- 5 Langage de commande (*shell*)
- 6 Résumé

# Interface graphique utilisateur

- On peut travailler sur un système Linux
  - une console en mode texte + *shell*
  - une console en mode graphique + une interface graphique
- On parle aussi de «bureau» (*desktop*)
- Interface Homme Machine (IHM)  
*Graphical User Interface (GUI)*

# Composants d'un bureau

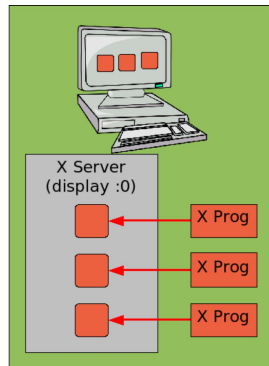
- Un bureau est composé de bibliothèques et d'applications
- Les bibliothèques graphiques définissent
  - un ensemble d'objets graphiques (*widgets*) :  
boutons, menus, barres de défilement, listes déroulantes, icônes, ...
  - l'aspect et le comportement de ces objets (*look & feel*)
- Applications dans un bureau complet
  - gestionnaire de fenêtres ou «compositeur»
  - gestionnaire du fond du bureau (fond d'écran, icônes, ...)
  - gestionnaire de tableaux de bord
  - gestionnaire de fichiers
  - ...
- Repose sur un système graphique
  - X Window System (ou X11 ou Xorg) : système historique
  - Wayland : nouveau système en cours de diffusion

# Exemples d'interfaces graphiques

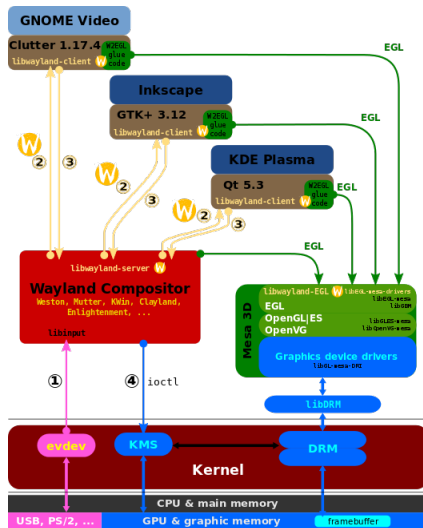
- KDE/Plasma 5
  - utilisé lors de l'atelier Linux
  - simple au premier abord, très riche quand on en a besoin
- GNOME 3
  - minimal, peu configurable
- MATE, Cinnamon
  - les héritiers de GNOME 2
  - pour ceux qui trouvent GNOME 3 trop minimal
- XFce, LXDE, LXQt
  - moins gourmands en ressources
  - adaptés aux mini-machines
- Autres
  - Budgie, Deepin, ...
  - simples gestionnaires de fenêtres

# Le modèle client/serveur de X11

- Les applications graphiques sont clientes d'un logiciel serveur qui tourne sur une console graphique (le serveur X11/Xorg)
- Elles envoient au serveur des ordres graphiques que le serveur traduit en commandes pour le GPU



# Architecture de Wayland



Source : Wikipedia

# Le gestionnaire de fenêtres ou compositeur

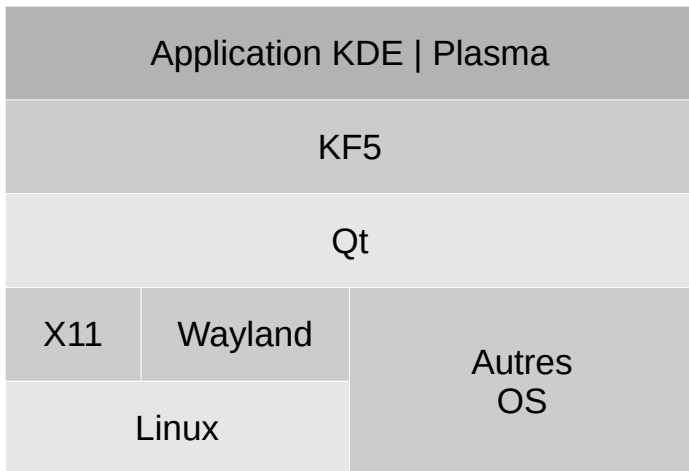
- C'est un composant essentiel d'un bureau
- Gestionnaire de fenêtres (*Window Manager*)
  - fenêtres rectangulaires, pas d'effets graphiques
  - sollicite peu le GPU (pourrait aussi tourner sur CPU)
- Compositeur
  - effets graphiques complexes
  - utilisation d'un GPU obligatoire
- Son rôle
  - gérer les fenêtres sur l'écran
  - permettre les manipulations par l'utilisateur :  
position, taille, avant-plan, arrière-plan, réduction, ...
  - placer autour d'une fenêtre un habillage spécifique (*look*)  
qui permet de manipuler la fenêtre
  - définir le comportement des fenêtres (*feel*)  
boutons de manipulation, dynamique de déplacement, ...
- La plupart des gestionnaires de fenêtres sont paramétrables
  - thèmes graphiques (*skins*)
  - comportement

# Quelques gestionnaires de fenêtres

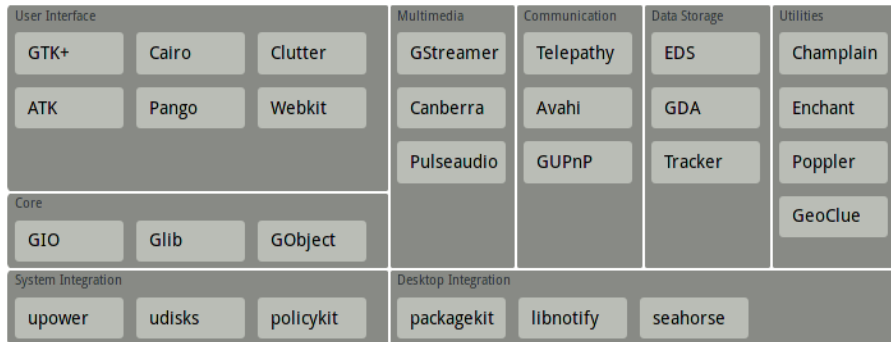
- Kwin
  - gestionnaire de fenêtres et compositeur de KDE/Plasma
  - effets graphiques intégrés
- Metacity
  - gestionnaire de fenêtres par défaut de GNOME 2
  - utilisé par GNOME 3 en mode « flashback »
- Xfwm
  - gestionnaire de fenêtres de Xfce
- Compiz
  - 1er compositeur
  - peut être utilisé avec MATE
- OpenBox, WindowMaker, IceWM, AfterStep, Ion, Blackbox, Enlightenment, Sawfish, ...
  - gestionnaires de fenêtres indépendants d'un bureau
  - peu gourmands en ressources



# Architecture logicielle simplifiée de KDE



# Architecture logicielle de GNOME 3



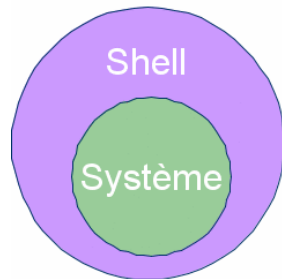
Source : projet GNOME

# Plan du cours

- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2
- 3 OS et types d'OS
- 4 Interfaces graphiques
- 5 Langage de commande (*shell*)**
- 6 Résumé

# Shell : introduction

- *shell* = interpréteur de commandes
- Intermédiaire (*shell* = coquille)  
entre l'utilisateur et le système d'exploitation
- Commandes permettant de piloter tous les aspects du système :
  - fichiers
  - processus
  - réseau
  - périphériques
  - ...
- Exemples de commandes :
  - `ls`
  - `ps`
  - `who`



# Shell : utilité immédiate

Un *shell* permet de :

- Gagner du temps par rapport à l'utilisation d'une interface graphique
  - manipuler d'un grand nombre de fichiers
  - naviguer dans une arborescence complexe et profonde
  - ...
- Accéder à tous les logiciels du système  
< 10% accessibles par les menus
- Écrire des scripts
- ...

# Shell : utilité plus générale

- Le *shell* est un outils très puissant
  - combinaison de commandes simples  
→ tâches complexes
  - automatisation de tâches répétitives
  - ...
- Un *shell* est présent dans tous les systèmes
  - Unix/Linux : nombreux *shells* disponibles
  - MacOS : un système Unix en fait, même *shell* que Linux
  - Windows : `cmd.exe`, PowerShell, *shells* Linux
- Aller au-delà de la simple utilisation d'un système
  - administration système (parcours B)
  - résolution de problèmes
  - ...

# Shell : 2 types de commandes

## Commandes externes (au *shell*)

- tous les logiciels disponibles sur le système
  - logiciels système Linux
  - compilateurs
  - éditeurs, IDE
  - applications diverses
- le *shell* sert de lanceur
- commandes indépendantes du type de *shell* utilisé

## Commandes internes

- commandes propres au *shell*, exécutées par le *shell* lui-même
- leur exécution ne nécessite pas le lancement d'un autre logiciel que le *shell*
- certaines peuvent varier d'un *shell* à l'autre
- exemples : `cd`, `alias`, ...

# Shell : 2 modes d'utilisation

## Mode interactif

- taper une commande au clavier
- valider avec la touche "Entrée"

→ le *shell* interprète la commande immédiatement

## Mode script

- écrire une suite de commandes dans un fichier texte
- lancer l'exécution du fichier

→ le *shell* exécute automatiquement le contenu du fichier ligne par ligne (comme un programme Java)



# Shell : logiciels

- Plusieurs *shells* existent

- Bourne Shell (`sh`) : *shell* historique (Steve Bourne, 1978)
- Korn Shell (`ksh`) : sur certains systèmes Unix propriétaires
- C-Shell (`csh`) : syntaxe proche du langage C
- Bourne Again Shell (`bash`) : version améliorée de `sh`
- Debian Almquist Shell (`dash`) : version rapide et légère de `sh`, *shell* utilisé par défaut pour `/bin/sh` (*shell* utilisé par les scripts du système)
- TENEX Shell (`tcsh`) : version améliorée de `csh`
- Z Shell (`zsh`) : très riche
- ...

- Différences

- fonctionnalités
- syntaxe de leur langage interne
- convivialité de leur "interface utilisateur"  
(facilité avec laquelle l'utilisateur peut taper/éditer une commande en mode interactif)

- En TP, nous utilisons le *shell* `bash`  
(*shell* par défaut dans la plupart des systèmes Linux)

# Syntaxe d'une ligne de commande

- Syntaxe identique en mode interactif ou dans un fichier de commandes (*script*)
- Composition :
  - un logiciel
  - un espace
  - ses éventuels arguments (paramètres), séparés par des espaces
  - un caractère de fin de ligne
- Exemple de commande

`ls`                      `-l`                      `-a`                      ↵  
 logiciel    argument 1    argument 2    Entrée/fin de ligne

→ Résultat

- `ls` : liste les fichiers
- `-l` : en format long
- `-a` : y compris les fichiers cachés
- pas d'autre argument : liste le répertoire courant

# Autres exemples de commandes

- `date`  
Affiche la date et l'heure
- `man LOGICIEL`  
Affiche des informations (le manuel) sur le logiciel dont le nom est passé en paramètre
- `cat FICHER`  
Affiche le contenu d'un fichier (convient pour du texte)
- ...

Énormément de logiciels disponibles en ligne de commande

- ~ 5000 sur les stations Linux de l'IUT2
- > 10000 sur un système Linux très complet

# Interface de la ligne de commande du *shell*

On peut :

- **éditer la ligne** en cours de saisie :  
→ flèches gauche et droite, BackSpace, Suppr
- **relancer une commande** précédente sans avoir à la retaper :  
→ flèches bas et haut, commande `history`
- **copier du texte** n'importe où :  
→ bouton de gauche de la souris (1, 2 ou 3 clics)
- **coller du texte** dans la ligne de commande,  
à l'emplacement du curseur  
(et pas ailleurs !, la position du pointeur importe peu) :  
→ bouton du milieu de la souris
- **compléter automatiquement** un nom de commande  
ou un nom de fichier :  
→ touche tabulation
- **interrompre une commande** :  
→ touches `Ctrl-C`
- **effacer l'écran** :  
→ touches `Ctrl-L`
- ...

# Sensibilité à la casse

- Les systèmes Unix (et donc Linux) sont sensibles à la casse
- Ils font la différence entre minuscules et majuscules pour
  - les logiciels et commandes
  - les noms de fichiers
  - tout ce qui porte un nom

→ Il faut donc respecter la casse dans le *shell*!

- Les logiciels et commandes usuelles sont en minuscules

# Plan du cours

- 1 Présentation de la ressource
- 2 Moyens informatiques de l'IUT2
- 3 OS et types d'OS
- 4 Interfaces graphiques
- 5 Langage de commande (*shell*)
- 6 **Résumé**

# Résumé

- Ressource importante comme **prérequis pour les autres ressources**
- Nombreux moyens informatiques  
reste à savoir les **utiliser efficacement** !
- **Respecter** : la charte, les quotas, ...
- Avec Linux on a la **liberté**
- Sous Linux on a le **choix**
- Apprendre à se servir d'un *shell* est **indispensable**