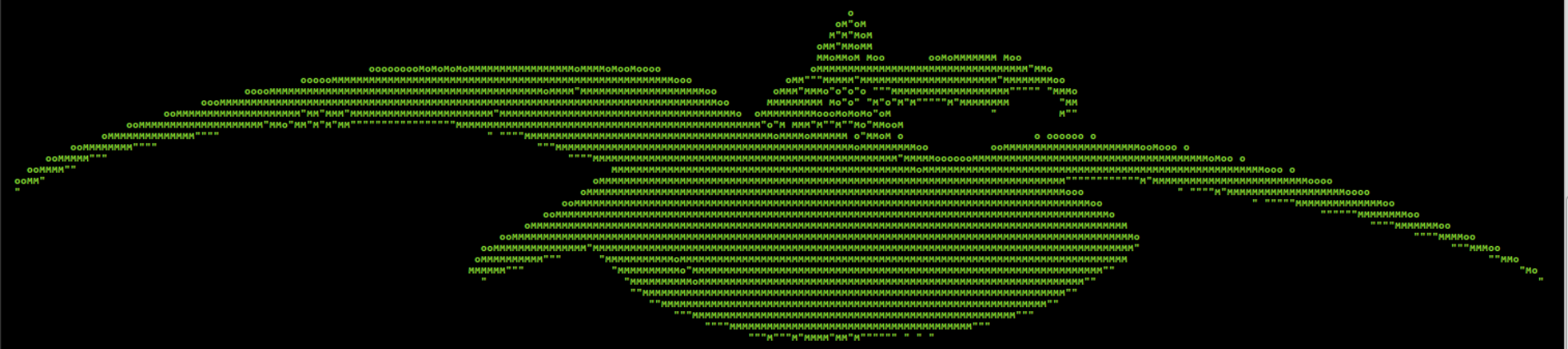
©3 arzack

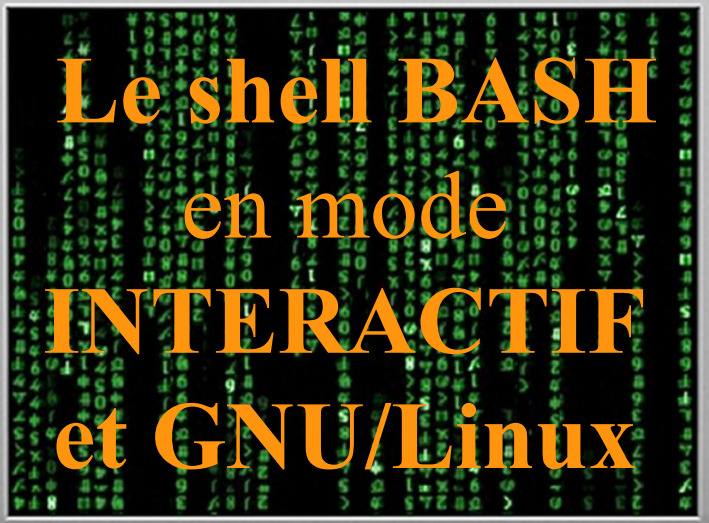
Fichier Édition Affichage Rechercher Terminal Aide

# Les Universités du Tetal@b



/media/user/UBACK/arzack>

# GNU/Linux avec : la coquille du terminal



# Le Shell c'est quoi, à quoi ça sert ?

- C'est un logiciel qui est un interpréteur de commandes pour le système d'exploitation de l'ordinateur.
- C'est aussi un langage de programmation en mode interprété.
- Il démarre et contrôle d'autre logiciels.
- En mode interactif, il est lancé dans un terminal.
- l'ensemble permet de communiquer avec le système.
- L'utilisateur peut alors activer un logiciel (ou commande) en spécifiant des options et ou des paramètres.
- Quand la commande est terminée, le shell envoie les résultats vers le terminal, sauf si l'utilisateur choisit une autre destination parmi :

  une variable, un fichier, un périphérique ou une autre commande.

# Le Shell c'est C.U.S.

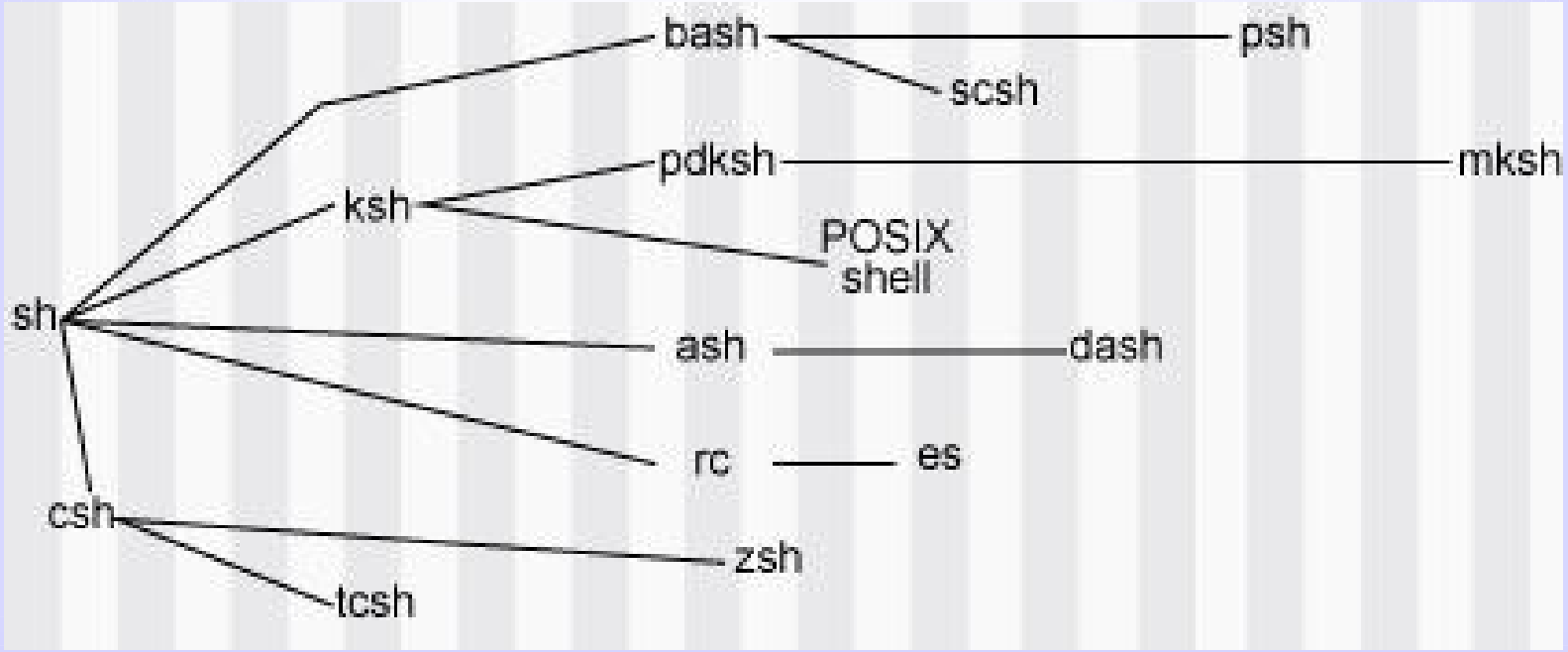
- Complet
  - Toute les commandes et logiciels du système sont accessibles (pour l'administrateur)
- Uniforme
  - Quelle que soit la distribution GNU/Linux utilisée, et pour un shell donné, l'utilisation est identique
- Secourable
  - Si le serveur graphique est hors service, le plus souvent une console « TTY(i) » fonctionne
  - Si le démarrage normal ne fonctionne plus, il est en général possible de démarrer sur une console de secours

# Shells utilisés avec GNU/Linux

- ‘sh’ l’ancêtre, souvent un lien vers dash ou bash
- ‘csh’ shell avec une syntaxe proche du C
- ‘ksh’ Korn shell
- ‘bash’ Bourne shell
- ‘zsh’ descendant de csh
- ‘ash’ Almquist SHell
- ‘dash’ descendant de ash

# Filiation des shells, Unix, Linux

- 1971 V6 shell par Ken Thompson (Bell Labs)
- 1975 BASH de Steve Bourne (Bell Labs)
  Hérite d'Algol68,  grammaire, macros



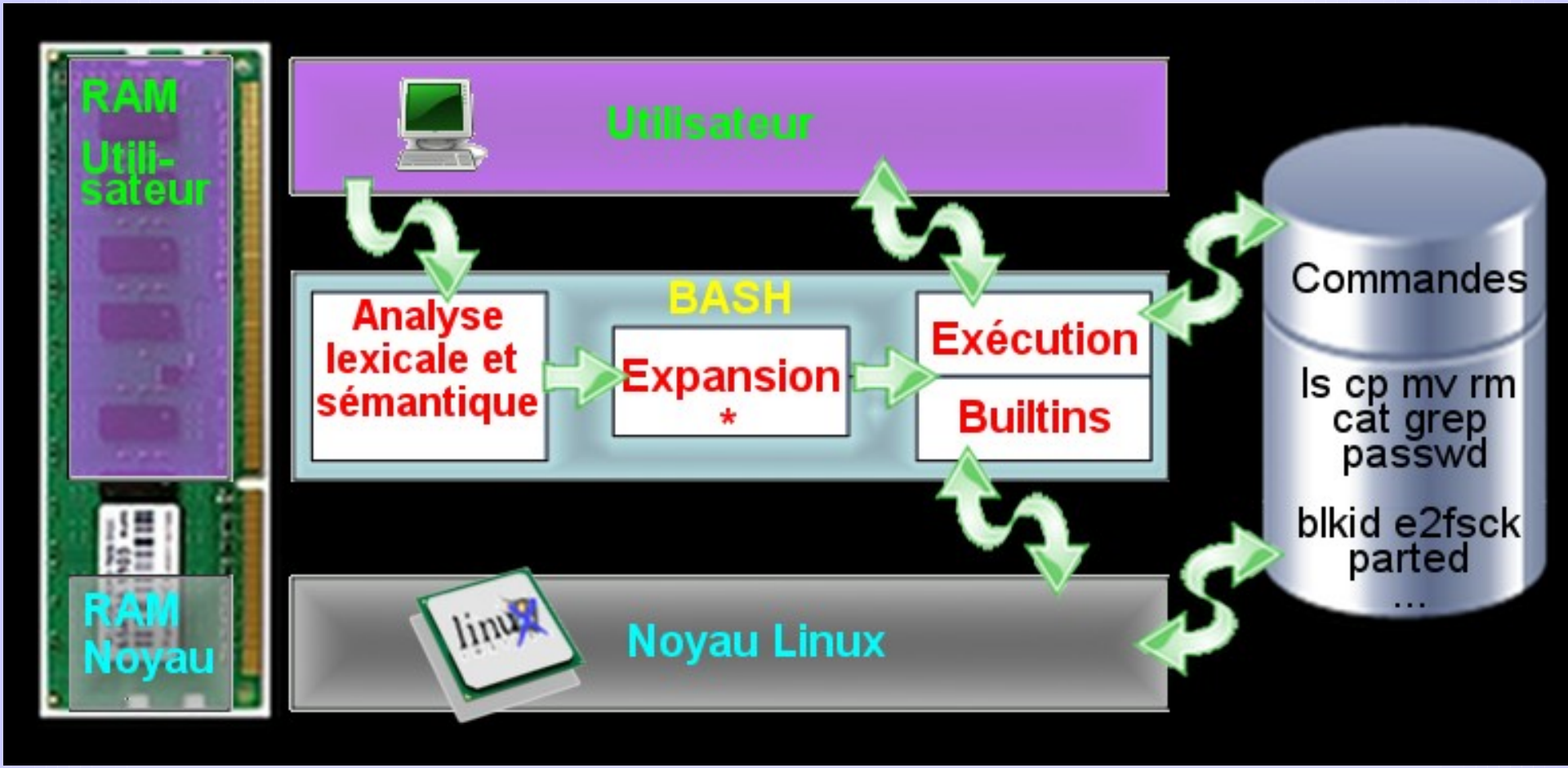
# Choix du shell

- À la création d'un utilisateur l'administrateur spécifie quel sera le shell à utiliser en mode console / terminal.
- Sous debian et dérivés la commande « chsh » permet de changer de shell.
- En général on trouve bash et dash par défaut.

# Présentation du shell BASH

- Shell GNU le plus utilisé, puissant et complet
  - Vaste communauté d'utilisateurs
  - Syntaxe pas toujours évidente
  - Ses auteurs le jugent trop gros et lent
- En mode interactif, lit et écrit dans un terminal :
  - console physique en mode caractères : TTYi
    - accessible avec « Ctrl Alt F[1-6] »
  - pseudo terminal en mode graphique : PTSi
    - xterm, gnome-terminal, lxterminal, ...

# L’environnement du shell

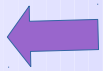
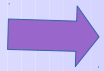


# Usage du BASH

- Depuis la ligne de commande on peut exécuter :
  - Des commandes BASH internes, nommées ‘built-in’
  - Des scripts, et des fonctions associées à la session
  - Des exécutables :
    - fichiers binaires ELF ou fichiers interprétables
    - Si leur chemin est présent dans $PATH (/bin, /sbin, /usr/bin, ... )
    - Sinon en donnant le chemin complet

ELF : Executable & Linking Format
$PATH : variable qui contient le chemin des répertoires où sont des exécutables

# Facilités du BASH

- Il facilite l'usage et l'administration du système
  - Complétion commande et nom de fichier : Tab
  - Historique, rappel de commande :  
  - Édition d'une commande historisée :  
  - Fichier script d'init personnalisable : .bashrc
  - Aide avec :
    - ses built-in: 'help « nom »', 'help help'
    - son manuel d'utilisation : man bash

# Algorithme simplifié du BASH

```
Au lancement: Interpréter le fichier .bashrc qui initialise des
  variables, dont le prompt, et les fonctions de l'utilisateur
Tant que le processus bash s'exécute faire
  Écrire le prompt et attendre une commande puis « Entrée »
  Si c'est une commande interne au bash
     Vérifier la syntaxe de la commande
     Si c'est OK alors
         Exécuter la commande interne
     Sinon signaler l'erreur
     Fin si
  Sinon rechercher un exécutable dans le PATH
    Si le PATH permet d'en trouver un alors
      Donner le contrôle à l'exécutable
      Attendre la fin de l'exécutable
    Sinon afficher "La commande est introuvable" ;
    Fin si
  Fin si
Fin Tant que
```

# Depuis BASH, l’aide GNU/Linux

- Pour l'utilisation d'une commande :
  - ‘man la_commande’
    (commencer par « man intro »)
  - ‘info la_commande’ (pas toujours installé)
- Et très souvent :
  - ‘la_commande - - help’
  - ‘la_commande - h’

# BASH, documentation sur le net

- Bash Guide for Beginners de Garrels sur :
  www.tldp.org
  https://guidespratiques.traduc.org/guides/vf/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf

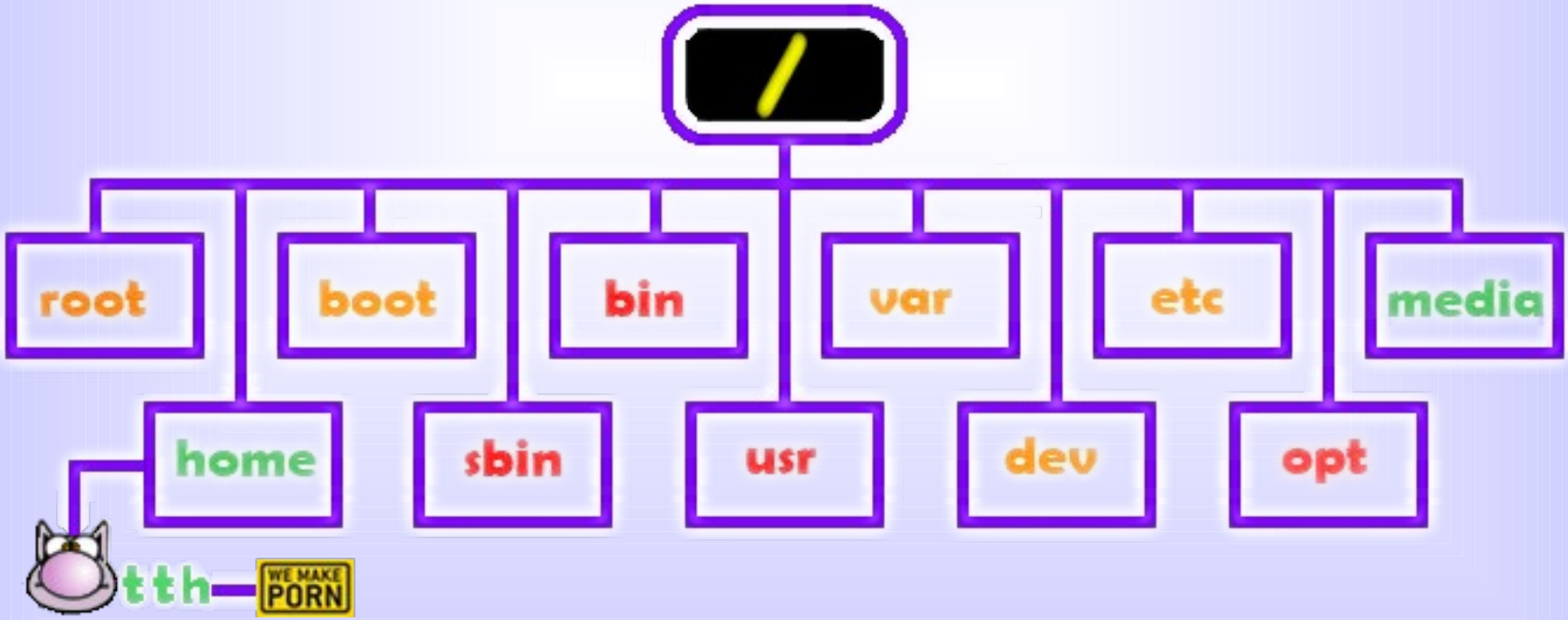
- https://debian-facile.org/doc:programmation:shells:debuter-avec-les-scripts-shell-bash

# Le contexte GNU/Linux

- Les répertoires du système ont une hiérarchie standard
  - Avec gestion de droits / répertoires et fichiers
- Tout exécutable s'inclut dans une hiérarchie de processus
  - Le processus n°1 (init ou systemd) initialise le système
- Tout périphérique est vu comme un fichiers spécial
  - /dev/sd[a-z][1-128] pour les media de stockage
  - /dev/tty[0-63] pour les terminaux
  - /dev/null => trou noir, /dev/zero envoie des '0'
  - /dev/random envoie n'importe quoi (aléatoire)

# Organisation des répertoires

- Voir « man hier » ou « man file-hierarchy »
- Chemin (path), exemple : « /home/tth/wmp »
  - Racourcis : « . » courant, « .. » contenant, « ~ » /home/tth



# Gestion des répertoires & fichiers

- Voir les fichiers : « ls chemin »
- Voir le répertoire courant : « pwd »
- Changer de répertoire : « cd chemin »
- Créer un répertoire : « mkdir nom »
- Créer un fichier : « touch nom »
- Copier un fichier : « cp nom nom2 »
- Supprimer : « rm nom2 » => nom2 irrécupérable

- Renommer, déplacer : « mv source cible »
- Modifier des droits « chmod ‘droits’ nom »

# BASH dans un terminal

- Il utilise 3 périphériques, re-dirigeables, vers ou depuis un fichier, variable, ou vers un autre processus :
  - /dev/stdin : entrée standard, alias 0 (défaut le clavier)
  - /dev/stdout : sortie standard, alias 1 (défaut l'écran)
  - /dev/stderr : sortie erreur standard, alias 2 (défaut l'écran)
- Exemples :
  - « ls >liste » ou « ls 1>liste » écrit dans le fichier liste
  - « ls inexistant 2 > ls.log » écrit l'erreur dans le fichier ls.log

# Les périphériques

- Matérialisés par les fichiers du « /dev »
- Il existe le type BLOCK :
  - Disques, CD, clé USB, …
  - Requièrent un formatage :
    - formater une clé USB en FAT32 :
      - « mkfs.fat  -I /dev/sdX »
- Et le type CHARACTER :
  - terminaux, son, clavier, ...

# Afficher les droits

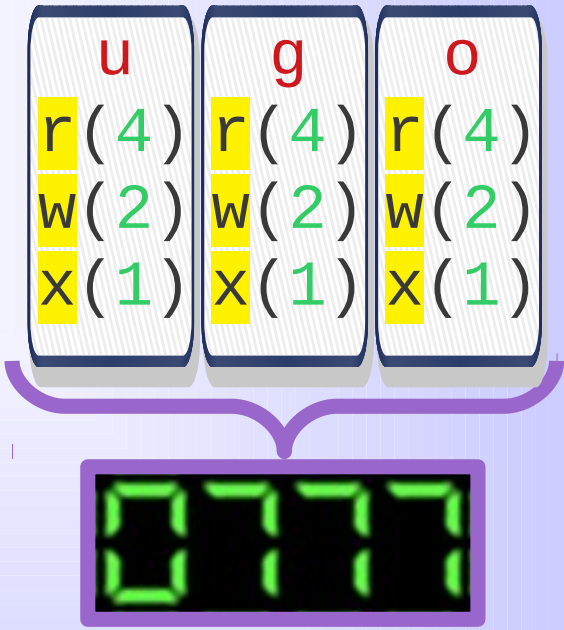
- Avec « ls -dg nom » ou « stat -c %A nom »

```
tth@DS1000 ~ $ stat -c %A wmp
-rw-r--r--
tth@DS1000 ~ $ stat -c %A dossier
drwxrwxr-x
```

- La 1ère lettre décrit le type, avec : '-' fichier normal, 'd' dossier (directory), 'l' lien, 'b' block, 'c' character
- Puis les droits (mode) 'rwxrwxr-x' read, write, exec' dans l'ordre : du propriétaire rwx, du groupe rwx , et r-x pour le reste des utilisateurs.
- Pour un dossier, 'x' autorise son parcours.

# Gestion des droits (modes)

- chmod « mode » « chemin » et mode sous forme :
  - Symbolique (Qui Droit) : '[ugoa]*([-+=]([rwx]*))
    - Qui : « user group others all »
    - Droit : « read write execute ». On peux les fixer : '=', ajouter : '+', soustraire : '-'
  - Octale : {+,-,=}{0..7}{0..7}{0..7}
    - Voir le schéma pour passer en octal.
    - Qui = position (colonne) du digit.
    - Droit = ∑colonne = valeur du digit.
    - Cette valeur qui est la somme du poids de chaque droit (en vert).
    - Le 4ème digit, à gauche est d'usage spécial. Les curieux iront voir « man chmod » pour les explications.



# Les droits, exemples

- Écriture symbolique :
  - « chmod a=rw lire_écrire_pour_tous»
  - « chmod a-rwx aucun_droit_pour_tous »
  - « chmod u=rwx,g-rwx,o-rwx moi_seul »
- Même résultat en écriture octale :
  - « chmod 666 lire_écrire_pour_tous»
  - « chmod 0 aucun_droit_pour_tous »
  - « chmod 700 moi_seul »

# Les logiciels en cours d'exécution

- Vu du système, c'est des processus connus par leurs PID
  - PID : Process IDentification, c'est un n° d'ordre
- Le 1er, le PID n°1 active des processus fils, qui lancent d'autres processus fils, l'ensemble est une hiérarchie :
  - « ps -u tth » affiche ceux du légendaire utilisateur tth
  - « pstree » affiche toute la hiérarchie
    avec « pstree -p » ajoute les PID)
  - « killall 'nom'» ou «kill -9 'pid'» anéantit le processus
- Exécuté depuis BASH, un processus
  est un fils de BASH et s'appelle un job

# Les jobs du BASH

- Une commande externe lancé depuis BASH est un « job » :
  - Qui peut être soit au premier plan : fg (foreground)
  - Qui peut être soit à l'arrière plan : bg (background)
  - Chaque job est identifié par son n° d'ordre
- Ces commandes internes de BASH gérent les jobs :
  - Lister : « jobs »
  - Lancer en bg : « cmde& », lancer en fg : « cmde »
  - Passer de fg à bg « Ctrl Z », puis de bg à fg : «fg %n°» ,
  - Terminer : «kill %n°», «kill %cmde»
- Pour les détails, commande : « help jobs »

# Jokers & substitutions du BASH

- ‘*?[]{}’ sont substitués (expansion) par BASH
  - ‘*’ : filtre 0 ou n caractères, ‘?’ : filtre 1 caractère
  - [xy] : x ou y, [a-z] : tout de ‘a’ à ‘z’, [^Aa] : ni ‘A’ ni ‘a’
  - {mot1,mot2} : filtre ces mots, {1..5} : épand la séquence et filtre
  - Pour échapper à la substitution :
    - Tout caractère spécial précédé d’un ‘\’
    - Toute suite de caractères entre ‘"’ ou ‘'’
- $ en début de mot est une variable a substituer par sa valeur, sauf si elle est placée entre 2 simples quotes ‘'’, elle sera substituée si elle est entre 2 doubles quotes ‘"’

# Essais de Jokers & Substitutions

- Si on crée le fichier *etoile il faut échapper l'*
  - On crée «touch *etoile» en échappant * avec \
  - Essayer : « ls * » , « ls ** »
  - Essayer : «?[e]* »
  - avec « rm ** » on détruit les fichiers débutant par *
  - Mais « rm * » supprime tout les fichiers du répertoire

- Créons le fichier etoile et la variable doom :
  - «touch etoile ;  doom=toile »
    - « echo *$doom »  => etoile
    - « echo "*$doom" » => *toile  ( ou « echo *$doom » )
    - « echo '*$doom' »  => *doom

# Re-directions « < > » et Tuyaux « | »

- Créer un fichier contenant la liste des fichiers du répertoire courant :
  - « ls -l >liste »
- Vérifier avec le ‘pager’ less :
  - « less liste »
- Filtrer ce fichier avec la commande grep :
  - « grep toto <liste »
- Idem en branchant ls et grep avec un tuyau :
  - « ls -l | grep toto »

# Information système

- « lsb_release -a » information sur la distribution
- Les messages du noyau depuis le démarrage :
  - « dmesg | less », filtrer les erreurs : « dmesg | grep error »
- Visualiser le matériel présent vu par Linux
  - « lscpu » description du CPU
  - « free -h » taille & occupation mémoire
  - « lsusb » périphériques du bus USB
  - « lspci » périphériques du bus PCI

# Information réseau

- Visualiser le matériel réseau
  - « lspci | grep -i -e ethernet -e network »
- État du réseau :
  - « ip -a » état des périphériques réseau
  - « ip -r » état du routage réseau
- État de la configuration
  - « cat /etc/resolv.conf » fichier renseigné automatiquement

# Les logiciels en paquets

- Il existe plusieurs système de gestion, et ou d'installation des logiciels (en général binaires), selon les familles de distributions GNU/Linux.
  - Pour les binaires : les paquets Debian (.deb), les .rmp ...
  - Pour les sources : les tarball, le clonage git, les fichiers zip, ...
  - Une installation à partir de codes sources impose de disposer des outils de compilation ou de l'interpréteur idoine pour les codes en questions.
- Toute installation, autre que personnelle à l'utilisateur, requiert des droits d'administration.
- Dans la famille Debian un logiciel est constitué d'un ou plusieurs paquets (pkg) et leurs dépendances éventuelles. Ces paquets sont téléchargeables depuis des dépôts présents sur des sites miroirs

# Gestion des paquets Debian

- Depuis le shell, passer administrateur (‘root’), ou préfixer les commandes par « sudo »
- Mises à jour des paquets Debian :
  - Copier la liste des paquets du dépôt en local : « apt update »
  - Puis mettre à jour tous ce qui peut l’être: « apt upgrade »
- Rechercher, installer un logiciel :
  - Chercher un logiciel: « apt search kiki »
  - S’informer sur un logiciel: « apt show kiki »
  - Installer un logiciel : « apt install kiki »
  - Installer hors dépôt : « dpkg -i kiki.deb »
  - Pour régler les problèmes de dépendances « apt install -f »