1. Description des droits rattaches a un répertoire :

Lorsqu'on exécute la commande ls avec son option -l, on peut voir les fameux neufs (9) caractères qui assignent le droit sur un fichier ou un répertoire, ces caractères se décomposent en 3 champs égales dont lesquels on peut identifier le droit du propriétaire qui se trouve dans le 1er champ, du groupe propriétaire sur le 2e champ et les autres utilisateurs sur le dernier champ. Chaque champ peut contenir des symboles différentes comme :

r : si le propriétaire concerné a le droit de lire (afficher) le fichier ou le répertoire.
w : si le propriétaire concerné a le droit d'écrire (modifier) le fichier ou le répertoire.
x : si le propriétaire concerné a le droit d'exécuter le fichier ou le répertoire.
- : si le propriétaire concerné n'a aucun droit sur le fichier ou le répertoire.

2. /etc/passwd :

Le fichier /etc/passwd est un fichier texte, chaque enregistrement décrivant un compte d'utilisateur. Chaque enregistrement se compose de 7 champs séparés par un deux-points.

Voici un exemple d'un enregistrement :

raja:x:1001:1000:ceci est un commentaire:/home/raja:/bin/bash	
raja	le nom de l'utilisateur (login name). C'est la chaîne de caractères que l'utilisateur entre lorsqu'il se connecte au système. Le nom d'utilisateur doit être unique dans le système.
x	les informations utilisées pour valider le mot de passe d'un utilisateur. Dans la plupart des usages modernes, ce champ contient "x" (ou "*", ou un autre indicateur) et les informations de mot de passe étant stockées dans un fichier de mots de passe séparé. Sur les systèmes Linux, "*" empêche les connexions d'un compte tout en conservant son nom d'utilisateur, alors que "*NP*" indique d'utiliser un serveur NIS pour obtenir le mot de passe. Avant le masquage du mot de passe (c'est-à-dire dans les premières implantations de Unix), ce champ contenait un hachage cryptographique du mot de passe de l'utilisateur (en combinaison avec un sel).
1001	l'identifiant d'utilisateur. Un nombre utilisé par le système d'exploitation à des fins internes. Il n'est pas nécessairement unique.
1000	l'identifiant de groupe. Un nombre qui identifie le groupe principal de l'utilisateur. Tous les fichiers créés par cet utilisateur peuvent être initialement accessibles à ce groupe.
ceci est un commentaire	un commentaire qui décrit la personne ou le compte. Généralement, il s'agit d'un ensemble de valeurs séparées par des virgules, fournissant le nom complet de l'utilisateur et ses coordonnées.
/home/raja	le chemin vers le répertoire personnel de l'utilisateur.
/bin/bash	le programme qui est lancé chaque fois que l'utilisateur se connecte au système. Pour un utilisateur interactif, il s'agit généralement d'une interface en ligne de commande (shell utilisé).

3. La notion d'archive sous GNU/Linux (avec exemple) :

Le principe est simple, prendre un grand nombre de fichiers et/ou répertoires et les regrouper en un seul gros dans le but de minimiser la taille qu'ils occupent sur l'espace disque (si on utilise un algorithme de compression comme gzip) et aussi les protéger contre les malware. Autrement dit, c'est une méthode permettant le stockage a long terme des données numériques.

Exemple :
tar est une commande qui permet de manipuler les archives

Extraction : tar xvf archive.tar	extrait les fichiers archive.tar, en affichant les noms des fichiers.
Compression : tar cvfz archive.tar dossier	crée un fichier tar contenant le dossier.

4. Description de la commande : grep -v ^root /etc/shadow (en se référant a la syntaxe Gle)

grep	C'est la commande de base indispensable.
-v	Une des options de grep, ceci est facultatif.
^root	Paramètre de l'option -v, ce paramètre est obligatoire afin que la commande n'a aucune erreur syntaxique. A noter que certaines options n'ont pas besoin de paramètre mais d'autres ont un ou plusieurs paramètre qui sont lui propre.
/etc/shadow	Le nom de fichier comme argument de la commande de base, obligatoire pour la plupart de la commande bash.

5. Processus : c'est un programme en cours d'exécution avec son environnement, c'est a dire dans le système d'exploitation. De façon plus précise : un ensemble d'instruction a exécuter.

6. Différence entre whatis et which

whatis	which
Cette commande retourne la description d'une commande qu'on a mis comme argument. Exemple : whatis nano : Nano's ANOther editor, an enhanced free Pico clone	Cette commande retourne l'emplacement du fichier binaire d'une commande qu'on a mis comme argument qui se trouve dans l'arborescence du système de fichier GNU/Linux. Exemple : which nano : /usr/bin/nano

7. UMASK (exemple d'utilisation)

Définition / principe	Le masque de protection est le fichier permet de définir les droits par défaut de tout crée. Ce masque comporte comme un filtre et utilise la notation numérique. On parle de filtre car il ne contient pas la série des 3 chiffres octaux correspondants aux droits à allouer aux fichiers, mais celle correspondant aux droits à ne pas allouer (complémentation de la vraie valeur en binaire). Le système UNIX affecte à un fichier les droits globaux résultant de la soustraction des droits maximaux 777 par le masque de protection.
Exemple d'utilisation	Si le masque de protection vaut 037 alors 740 (=777-037) seront les droits alloués à tout nouveau fichier. Syntaxe : umask 037