

**cd** [ espace/~ / /home/username] : revenir à son répertoire courant; **cd ..** : revenir au répertoire précédent; **man nom\_commande** : obtenir le manuel d'une commande ; **nom\_commande --help** : affiche une aide succincte, Quel est le Shell utilisé sur votre ordinateur ? **Echo \$ SHELL**; afficher la version du système que vous utilisez **uname -v**; Donner le chemin absolu du fichier commande (cat ) **wheries cat**; liste toutes les commandes récemment exécutées, avec un numéro pour chaque commande **history**; **bashrc** un fichier de configuration shell cache situe a la racine de votre home ~; **sudo apt install/remove <logiciel>** Mise à jour vérifier les versions installées vs disponibles **sudo apt update date** affiche la date système **at**, **batch** programme une action.

**Changement des droits d'accès chmod mode fichier** : changer les droits d'accès d'un fichier mode : **a+r** : tout le monde a le droit de lire **og-w** : les autres et le groupe n'ont pas le droit d'écrire **u+x** : l'utilisateur a le droit d'exécuter **744** : l'utilisateur a tous les droits, le groupe et les autres ne peuvent que consulter le fichier **a** : tout le monde **o** : les autres **g** : le groupe **u** : l'utilisateur **r** : lire (4) **w** : écrire (2) **x** : exécuter (1) ;+ : donne le droit ; - : enlève le droit expl: hmod 740 toto.c ; chmod u+rwx toto.c ; chmod g+r toto.c ; chmod g-wx toto.c ; chmod o-rwx toto.c

**Affichage: ls options chemin** : affiche le contenu du répertoire; **-a** : tous les fichiers (même cachés); **-l** : affichage détaillé et droit; **-t** : trie en fonction de la date ; **-h** : affiche les informations de taille en octet **-F** : ajoute des signes(@#) pour différencier les types ; **-lt** : affiche en ordre date de modification **-la** caché et détaillé ; **-lh** : montre en plus le poids d'occupation; s **-R** : affiche une liste des caractéristiques de chaque fichier dans tous les répertoires en partant d'où nous sommes ; **-IR** : affiche en plus les droits, **pwd** : affiche le répertoire courant;

**Manipulation de fichiers: mkdir** -p *répertoire* : créer un répertoire (-p si on veut en créer plusieurs en une commande expl: **mkdir -p** Brice Renaud BP et sous repertoire en meme temps separe de / expl: **OBRP/Brice**) **touch fichier** : Si le fichier existe, met à jour la date de dernière modification Sinon, crée un fichier vide ; **file fichier** : identifie le type de fichier ; **mv [source] [destination]** : déplace (renomme) un fichier (un répertoire)expl: **mv /home/utilisateur/fichier.txt Cours/Unix(mv /home/utilisateur/fichier.txt /home/utilisateur/newfichier.txt))** ; **rm [options] fichier1 fichier2 ...** : sans option il supprime des fichiers seulement et **rmdir** sans option supprime des répertoires vides ;les option que voici marche avec **rm** : **-r** ou **R** : supprime récursivement un répertoire non vide **-i** : supprime en demandant une confirmation avant chaque effacement ; **cp [options] source destination** : Si destination = fichier : copie le fichier et le renomme en destination au nom écrit expl: **cp /home/brice04/fichier.txt /home/brice04/Cours/newnomdefichier**); Si destination = répertoire : recopie le fichier dans le répertoire expl: **cp /home/brice04/fichier.txt /home/brice04/Cours**); les option que voici marche avec **cp**: **-i** : confirmation en cas d'écrasement **-p** : préservation des dates d'accès **-r** : copie récursive (pour les répertoires imbriqués copie le répertoire entier repertoire et tout son contenu dans le répertoire de destination);**du -h** :taille du fichier ou compare si l'on mes deux; **diff** :permet de comparer deux fichiers; **lpr** : imprimer un fichier; **lpq** : afficher les fichiers en attente d'impression.;**lprm** : détruire des fichiers en attente d'impression **//1. rm \*.oAction** : Supprime tous les fichiers ayant une extension .o.

**Explication** :Le caractère générique \* remplace n'importe quel nom de fichier..o cible uniquement les fichiers dont l'extension est .o.Exemple :Fichiers supprimés : file.o, test.o, program.o.2. **rm a\*Action** : Supprime tous les fichiers dont le nom commence par la lettre a.Explication :a\* : Correspond à tous les fichiers commençant par a, suivis d'une chaîne quelconque de caractères.Exemple :Fichiers supprimés : apple, alpha.txt, a123. 3. **rm \*Action** : Supprime tous les fichiers du répertoire courant.Explication :Le caractère \* correspond à tous les fichiers, quel que soit leur nom ou extensionAttention :Cette commande supprime tous les fichiers sans confirmation par défaut.Les sous-répertoires ne sont pas supprimés, sauf si vous ajoutez une option comme -r.4. **rm \*.?Action** : Supprime tous les fichiers ayant une extension composée d'une seule lettre.Explication :\* : Correspond à une chaîne quelconque de caractères pour le nom du fichier. ? : Correspond à une extension formée d'un seul caractère (par exemple, .c, .h, .o).Exemple Fichiers supprimés : main.c, data.h, mais pas file.txt.5. **Caractères génériques utilisés dans les commandes :\*(astérisque)** :Représente une chaîne quelconque de caractères, y compris aucun caractère.Exemple :a\* correspond à : a, apple, alpha.txt, etc.? (point d'interrogation) :Représente exactement un caractère quelconque.Exemple :file.? correspond à : file.c, file.h, mais pas file.txt.

**Manipulation de texte :vi fichier** : ouvrir et permet d'écrire dans un fichier; **xemacs <nom\_fichier>**; **cat fichier** : affiche le fichier (-n inclut aussi le numero de ligne), **more fichier** : affiche le fichier page par page volumineux; **head -n x fichier** : affiche les x premières lignes **tail -n x fichier** : affiche les x dernières lignes. **wc option fichier** : **-c (-m)** : affiche le nombre de bytes (caractères). **-w** : affiche le nombre de mots, **-l** : affiche le nombre de lignes; **// grep options motif fichier** : cherche les lignes contenant le motif dans les fichiers(grep "error" log.txt Affiche toutes les lignes contenant le mot error dans le fichier log .txt) , **-c** : compte le nombre de lignes contenant le motif(grep -c "TODO" notes.txt Compte le nombre de lignes contenant TODO dans notes .txt); **-n** : indique compte le numéro des lignes trouvés f(grep -n "main" program.c Affiche les lignes contenant main avec leur numéro dans program .c); **-l** : affiche le nom des fichiers qui contient etc(grep -l "printf" \*.c Affiche la liste des fichiers .c contenant la chaîne printf); **-v** affiche les lignes ne contenant pas la chaîne (grep -v "debug" script.sh Affiche toutes les lignes de script .sh ne contenant pas le mot debug); **-x** ligne correspondant exactement à la chaîne (**grep -x 'hello' hp1.txt** Seule la ligne contenant exactement "hello" sera affichée. La ligne "hello world" ne correspond pas, car elle contient des caractères supplémentaires. ; **grep -x " hp1.txt**Ligne vide) **grep -x '^\$' hp1.txt > hp1bis.txt**, **grep** : Recherche les lignes correspondant à un motif. **-x** : Indique que la ligne entière doit correspondre exactement au motif (ni plus, ni moins). **^\$** : ^ : Début de la ligne.\$ : Fin de la ligne.Ensemble, **^\$** correspond à une ligne strictement vide (aucun caractère, même pas d'espaces ou de tabulations).**hp1.txt** : Fichier source où la recherche est effectuée.> : Redirige la sortie vers le fichier hp1bis.txt, en écrasant son contenu si le fichier existe.; **grep [1V] hp1** Les crochets [ ] permettent de rechercher l'un des caractères listés.Les plages (ex. : [0-9]) facilitent la recherche de séries contiguës.Les caractères spéciaux dans les crochets doivent être échappés avec \. Trouver les lignes contenant une voyelle (a, e, i, o, u).Expression : [aeiou] Trouver les lignes contenant un chiffre.Expression : [0-9]Trouver les lignes contenant une lettre (majuscule ou minuscule).Expression : [A-Za-z]Trouver les lignes contenant ., (, ou ).Expression : [\.\(\)\]

**find répertoire de recherche critère de recherche** : trouver des fichiers à partir de critères Les critères de recherche sont les suivants : **-name** : recherche sur le nom du fichier(Recherche pour affiche ses racine le fichier document .txt depuis le répertoire /home: **find /home -name "document.txt"**); **-type** : recherche sur le type (**d**=répertoire, **c**=caractère (**find /dev -type c** Cela recherche dans le répertoire /dev tous les fichiers spéciaux de type caractère.), **f**=fichier normal(**find . -type f** Liste tous les fichiers dans le répertoire actuel ou presice repertoire en remplaçant le point)), **-perm** recherche sur les droits d'accès du fichier, **-links** recherche sur le nombre de liens du fichier, **-user** recherche sur le propriétaire du fichier (**find /home -user alice** Liste les fichiers appartenant à l'utilisateur alice), **-group** recherche sur le groupe auquel appartient le fichier, **-size** recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets **find /tmp -size +2048k** Cherche les fichiers de plus de 2 048 Ko dans /tmp),**atime** recherche par date de dernier accès en lecture du fichier, **-mtime** recherche par date de dernière modification du fichier( **find /var -mtime -7** Cherche les fichiers modifiés il y a moins de 7 jours dans /var), **-ctime** recherche par date de création du fichier; **find . -name "hp1.txt"** :Recherche le fichier nommé hp1.txt dans le répertoire courant (.) et ses sous-répertoires. **-exec ls -l {} \;** :Pour chaque fichier trouvé ({} représente le fichier trouvé), exécute la commande ls -l.Le \; :Indique la fin de la commande -exec. **// sort options fichier(s)** : trier des fichiers ou leurs contenus **-b** : Option à utiliser lorsque le critère de tri est alphanumérique et que les valeurs dans les champs sont calés à droite et complétés avec des espaces à gauche **-k** : champ [.car] [options] , champ [.car] [options] Spécifier le ou les champs à prendre en compte en tant que critère de tri. Le critère de tri peut commencer ou finir à une certaine position de caractère - du champ. **-n** : Préciser que le critère de tri doit être traité comme valeur numérique et non comme chaîne de caractères. **-r** : Demander un tri décroissant **-t sep** : Préciser le caractère séparateur de champs (espace par défaut) **-u** : Suppression des doublons;

**sed options 'fonctions' fichier** : permet d'appliquer un certain nombre de commandes sur un fichier et d'afficher le résultat (sans modifier le fichier de départ)**Exemple**: **sed 's/Paris/Marseille/g' FichierEntree.txt > FichierSortie.txt** Ici, « s » signifie que l'on substitue les données, « g » que la fonction est globale, on le fait donc dans tout le fichier. Globalement, cette fonction va remplacer toutes les occurrences du mot **Paris** du texte **FichierEntree.txt** par **Marseille**, et va placer

cette version du fichier dans **FichierSortie.txt**. Archivage : **tar -cvf archive.tar répertoire** ; Restitution : **tar -xvf archive.tar** ; Archivage + Compression : **tar -cvzf archive.tar.gz répertoire** **tar -cvzf archive.tgz répertoire** ;Décompression + Restitution :**tar -xvzf archive.tar.gz**

La commande **ln**: Permet de faire un lien sur un fichier avec possibilité de lui donner plusieurs noms et économiser de la place; **-s** : Option pour créer un lien symbolique (au lieu d'un lien matériel). **ln -s /etc/config.txt my\_config** permet de créer un **lien symbolique** (ou "lien doux") nommé my\_config, qui pointe vers le fichier /etc/config.txt //&& : (ET logique) Exécute la deuxième commande uniquement si la première commande réussit (Retourne un code de sortie 0). commande1 && commande2, mkdir test\_dir && echo "Dossier créé avec succès" Si mkdir test\_dir réussit, alors echo "Dossier créé avec succès" s'exécute.n// || : ou logique Exécute la **deuxième commande uniquement si la première commande échoue** (retourne un code de sortie non nul). commande1 || commande2, **mkdir test\_dir || echo "Échec"** de la création du dossier" Si mkdir test\_dir échoue (par exemple, si le dossier existe déjà), alors echo "Échec de la création du dossier" s'exécute.**Pipe (|)** : Prend la sortie de la première commande et la transmet comme **entrée de la deuxième commande commande1 | commande2**.Afficher les fichiers page par page, : ls | more, Compter les utilisateurs connectés : **who | wc -l**, Compter les mots dans la liste des fichiers : **ls | wc -w**; **ls | grep ".txt"** Liste les fichiers du répertoire courant (ls) et affiche uniquement ceux contenant ".txt" grâce à grep. // ; (Séparateur de commandes)Permet d'exécuter plusieurs commandes l'une après l'autre, indépendamment de la réussite ou de l'échec des précédentes. commande1 ; commande2, mkdir test\_dir ; echo "Commande suivante exécutée" **mkdir test\_dir ; echo "Commande suivante exécutée"** echo s'exécutera même si mkdir échoue./// **mkdir test\_dir && echo "Succès" || echo "Échec"**, Si mkdir test\_dir réussit, affiche "Succès".Si mkdir test\_dir échoue, affiche "Échec". **ls | grep ".txt" | wc -l touch fichier.txt** Liste les fichiers contenant ".txt" et affiche leur nombre total (wc -l). **mkdir test\_dir ; cd test\_dir ; touch fichier.txt** ;Crée un répertoire, se déplace dedans, puis crée un fichier vide.

<, Redirection de l'entrée standard, **sort < fichier.txt**, Trie les lignes du fichier fichier.txt.// >, Redirection de la sortie standard, **echo "Hello, World!" > fichier.txt**, Écrit "Hello, World!" dans le fichier fichier.txt, écrasant son contenu s'il existe déjà.// 2>, Redirection de la sortie erreur standard, **ls non\_existant\_fichier 2> erreurs.txt**, Envoie le message d'erreur ("fichier non trouvé") vers erreurs.txt.// >>, Concaténation de la sortie standard, **echo "Ligne supplémentaire" >> fichier.txt**, Ajoute "Ligne supplémentaire" à la fin de fichier.txt, sans écraser son contenu.// 2>>, Concaténation de la sortie erreur standard, **ls non\_existant\_fichier 2>> erreurs.txt**, Ajoute les messages d'erreur de la commande à la fin de erreurs.txt.// > **fichier 2>&1**, Redirection des sorties standard et erreur dans le même fichier, **commande > fichier.txt 2>&1**, Combine la sortie standard et les erreurs dans un seul fichier (fichier.txt).

Expressions régulières : ^ début de ligne; . un caractère quelconque \$ fin de ligne **x\*** zéro ou plus d'occurrences du caractère x; **x+** une ou plus occurrences du caractère x; **x?** une occurrence unique du caractère x; [...] plage de caractères permis; [^...] plage de caractères interdits; \ {n\} pour définir le nombre de répétition n du caractère placé devant ,**Exemple** : **^[A-Z][5-9].\$** Ici on cherche une chaîne de 3 caractères. Le premier est une majuscule, le second un chiffre entre 5 et 9 et le dernier est un caractère quelconque

**Secure shell (SSH)**

**ssh** *NomUtilisateur@NomServeur* : Permet de se connecter à un serveur distant, en renseignant le login et le nom du serveur auquel on veut se connecter (cette commande vous demandera le mot de passe de la session s'il en possède un). **Who** : liste des utilisateurs en ligne et date de creation

**scp** *options fichier serveur:/DossierDeDestination* : Permet de déplacer un fichier dans le serveur auquel vous êtes connectés, à la destination souhaitée  
*Script Shell ./<Nom du script>.sh* : pour exécuter le script # **!/bin/bash** : entête de chaque fichier **let `c=a+b`** : pour les calculs de nombres **echo** texte « ls » ou 'ls' affiche le mot ls 'ls' exécute la commande ls \$a : affiche la variable a

Variables utiles; **\$0** : contient le nom du programme en cours; **\$1...\$9** : contient les différentes valeurs des arguments;  **\$#** : contient le nombre d'arguments/paramètres; **\$\*** : contient une liste de l'ensemble des paramètres;  **\$\$** : contient le numéro du processus en cours

Conditions	Boucles for	Boucles while
if [ condition ] && [ condition ]	for <nom de la variable> in `seq 1 10`	while [condition]
then	do	do
traitement	traitement	traitement
then	done	done
traitement		
fi		

Conditis nmbrs **-eq** : vérifie l'égalité entre 2 nmbrs **-ne** : vérifie l'inégalité entre 2 nmbrs **-lt/le** : < ou <= **-gt/ge** : > ou >=  
Conditions chaines = / != : vérifie si 2 chaines sont identiques/différentes **-z** : vérifié si la chaîne est vide **-n** : opposé de -z  
Conditions sur les fichiers **-e** : vérifie si le fichier existe **-d** : vérifie si le fichier est un répertoire **-f** : vérifie si le fichier est un fichier **-L** : vérifie s'il s'agit d'un lien symbolique **-r/-w/-x** : vérifie si le fichier dispose des droits r/w/x **-nt/ot** : vérifie si le 1<sup>er</sup> fichier est plus récent/ancien que le second

**Autres**: **sudo** **sudo <commande>** lance la commande en tant qu'administrateur; Installation/suppression de logiciel : Passer en mode « super utilisateur », utile si vous n'avez pas la permission de faire une commande, **last \$1** : affiche la date de la dernière connexion de l'utilisateur passé en argument (\$1), **export PS1 = 'NouveauPrompt - '** : modifie le nom du prompt du terminal (ici en NouveauPrompt - ) **alias** **NomNouvelleCommande='Commande'** : permet de créer de nouvelles commandes, en renseignant entre l'action que la commande réalise. **unalias** **NomNouvelleCommande** : Supprime un alias que vous avez créé. **ps** : affiche les processus en cours d'exécution **top** : similaire à ps, affiche en temps réel les processus actifs **du** : estimation de l'espace disque occupé par un fichier.