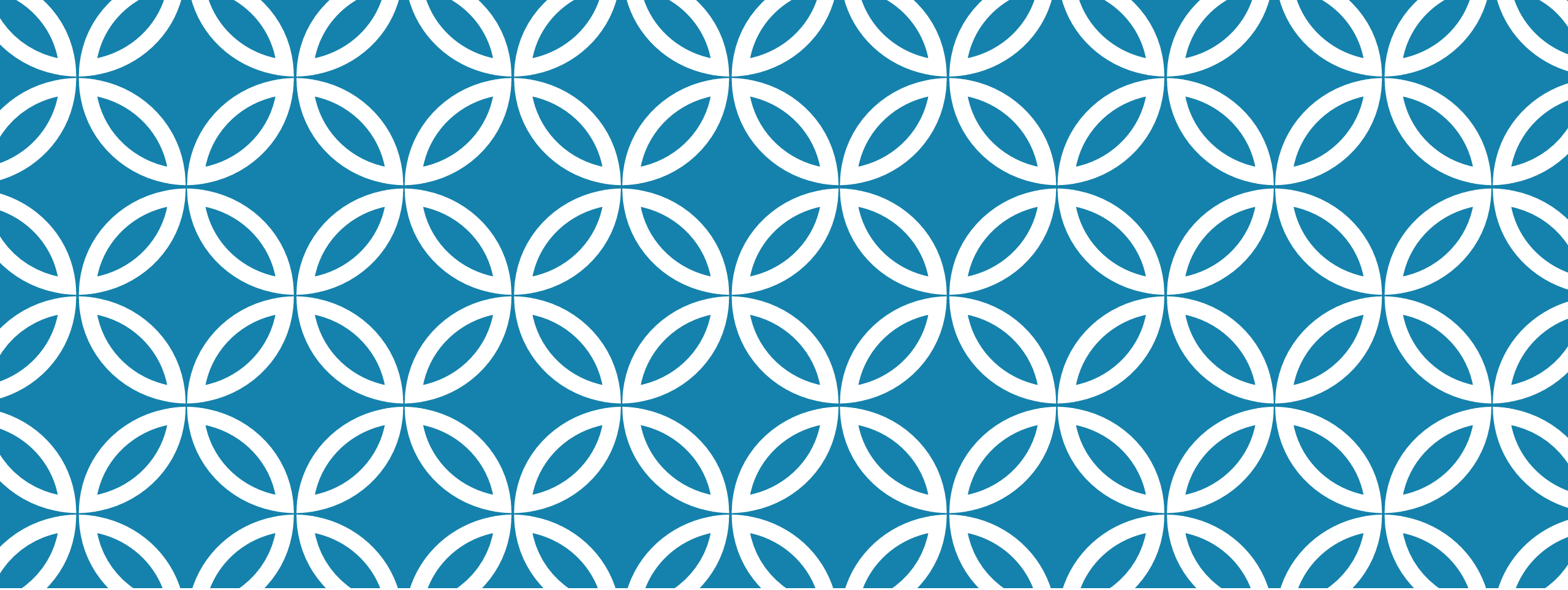


UNIX

Jérémy PERROUAULT

UN PEU D'HISTOIRE

Historique et contexte

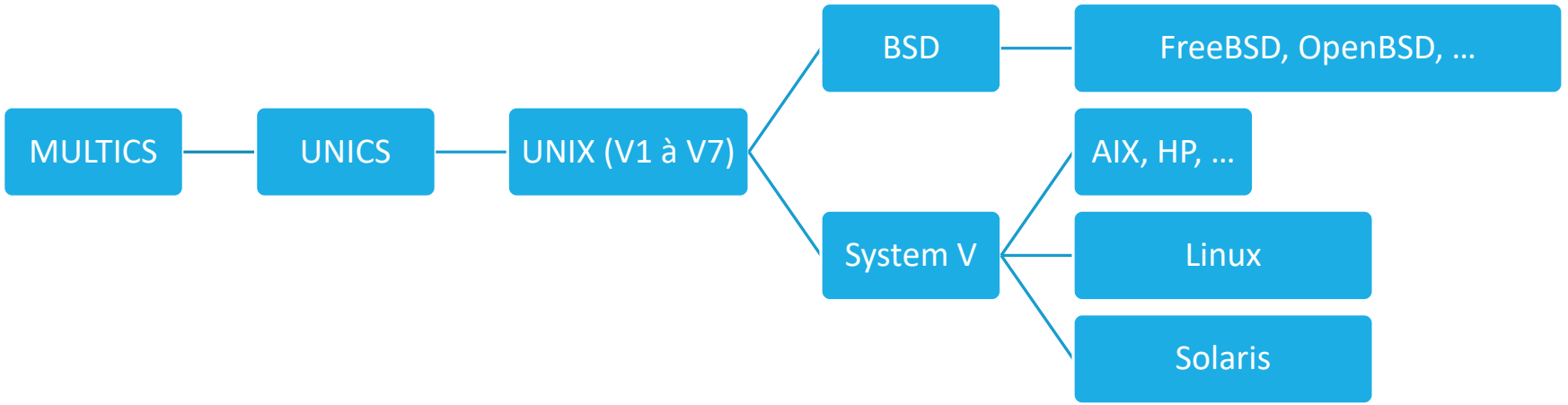
UN PEU D'HISTOIRE

1964	Projet *MULTICS* (MULTiplexed Information and Computing Service) Développé par un consortium : MIT, General Electric, Laboratoires Bell et AT-T
1969	Projet *UNICS* (UNiplexed Information and Computing Service) Développé en Assembleur par Ken Thomson et les Laboratoires Bell
1971	Publication du manuel « The UNIX Programmer »
1973	Réécriture de UNIX en C
1979	Reprise du projet par le monde académique (Université de Californie à Berkeley)

UN PEU D'HISTOIRE

Après 7 versions de UNIX (elles-mêmes après MULTICS et UNICS)

- UNIX BSD en 1977
  - Qui pose les bases des systèmes tels que FreeBSD, NetBSD, OpenBSD et MacOSX
- UNIX System V en 1983
  - Qui fournit le socle aux générations futures telles que AIX, HP-UX, IRIX, UnixWare
  - Mais est également précurseur de deux autres branches, très connues par le grand public : Linux (ou GNU/Linux) et Solaris



UN PEU D'HISTOIRE – LINUX OU GNU/LINUX ?

Les systèmes UNIX étaient payants et devenaient de plus en plus cher

- Il fallait créer une alternative : le projet **GNU** (GNU's Not UNIX) est né
  - Cette solution est gratuite et libre
  - C'est une « copie » de UNIX

Linus Torvald, étudiant Finlandais, crée dans le même temps son système d'exploitation libre, qui prend le nom de Linux

Les deux projets sont complémentaires, et ont fusionné pour donner GNU/Linux

- GNU (programmes libres : copie des fichiers, suppression des fichiers, éditeur de texte, …)
- Linux (noyau d'OS)

UN PEU D'HISTOIRE – GNU/LINUX ?

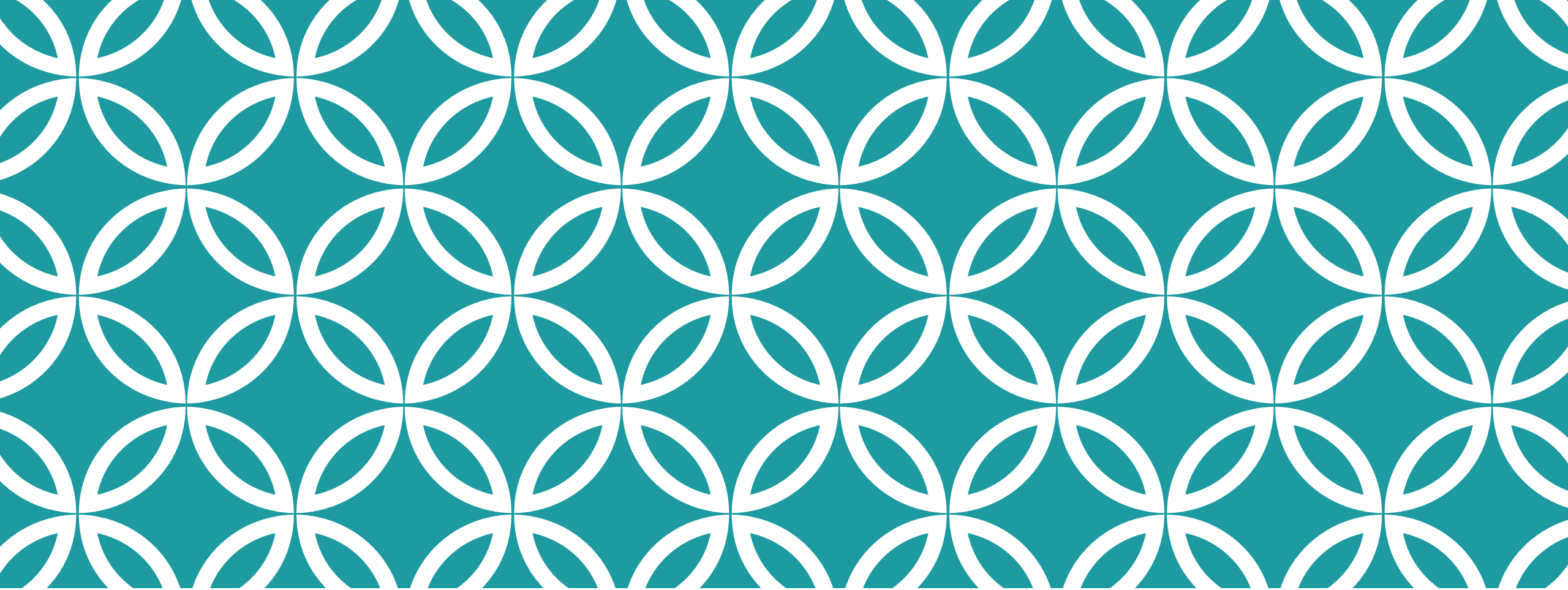
Quant à GNU/Linux, ce système donnera naissance à de multiples distributions

- Slackware
- Mandriva
- Red Hat
- SuSE
- Debian
  - Ubuntu
  - Knoppix
  - …

UTILISATION

UNIX se distingue par les caractéristiques suivantes

- Multi-utilisateurs (plusieurs utilisateurs peuvent utiliser, simultanément, le système)
- Multi-tâches (un utilisateur peut exécuter plusieurs programmes en même temps)
- Repose sur un noyau (kernel) qui utilise 4 concepts
  - Fichiers
  - Droits d'accès
  - Processus
  - Communication interprocessus (IPC)



LES FONDAMENTAUX

La structure UNIX
Les commandes UNIX

LE SYSTÈME DE FICHIERS

Arborescence à parcourir à partir du chemin racine, appelé **root**

La racine est accessible avec le chemin « / »

Chemin absolu	Utilité
/bin	Exécutables essentiels au système, utilisable par un utilisateur
/boot	Fichiers permettant au système de démarrer
/dev	Points d'entrée des périphériques (dev comme device)
/etc	Fichiers et exécutables liés à l'administration du système (utilisateurs, groupes, réseau, …)
/home	Répertoire personnel des utilisateurs
/lib	Bibliothèques partagées essentiels au système au démarrage
/mnt (ou /media)	Points de montage des partitions temporaires (CD, Clé USB, lecteur réseau, …)

LE SYSTÈME DE FICHIERS

Chemin absolu	Utilité
/opt	Packages d'application supplémentaires
/proc	Fichiers mémoire, E/S, périphériques, compatibilité avec le noyau
/root	Répertoire de l'administrateur principal **root**
/usr	Exécutables ou autres fichiers liés aux utilisateurs
/var	Données variables, on y trouve notamment les fichiers journaux (logs)
/tmp	Données temporaires

LE SYSTÈME DE FICHIERS

Dans un système UNIX, tout est fichier

- Un périphérique est considéré comme tel
- Son point de montage (ou point d'accès) est mappé sur un répertoire
- Un répertoire est un fichier signé « d » (comme directory)
- Les fichiers classiques ne sont pas signés (signés « - »)

LE SYSTÈME DE FICHIERS

Lorsque qu'un périphérique est inséré (clé USB par exemple)

- Son point d'accès se trouvera dans « /dev »
- Puisqu'il s'agit d'un disque SCSI, son nom commencera par « sd » pour « **S**CSI **D**isk »
  - Chaque disque a sa lettre : a, b, c, d, …
  - Chaque partition de chaque disque a un numéro : 1, 2, 3, …
- Donc
  - La première partition du premier disque SCSI inséré pointera sur /dev/sda1
  - La troisième partition du deuxième disque pointera sur /dev/sdb3

- Il faudra ensuite « monter » la partition souhaitée dans un répertoire
  - Souvent dans « /mnt » ou dans « /media »
  - /dev/sda1 → /media/projets

L'INTERPRÉTEUR

Pour communiquer avec le système

- Utilisation d'une interface graphique (type bureau)
- Utilisation d'un interpréteur de commandes Shell
  - Interface entre l'utilisateur et le système d'exploitation
  - Application chargée d'interpréter les commandes des utilisateurs et de les transmettre au système
  - Plusieurs types de shell existent
    - bsh ou sh Bourne shell, développé par Stephen Bourne
    - bash Bourne again shell (version améliorée de sh)
    - csh C shell (évolution de sh, syntaxe proche du langage C)
    - …
  - Le nom du shell correspond, en général, au nom de son fichier exécutable
    - /bin/sh
    - /bin/bash

L'INTERPRÉTEUR

En bash, l'invite de commande se présente comme suit

- login@machine$ ou login@machine#
  - $ si utilisateur normal
  - # si super-utilisateur (administrateur)

Quelques raccourcis clavier

- Ctrl+C Terminer le programme en cours
- Ctrl+Z Mettre en attente le programme en cours
- Ctrl+D Terminer l'entrée standard
- Tabulation Compléter le nom d'un répertoire, d'un fichier ou d'une commande existante

L'INTERPRÉTEUR

Lorsqu'une commande est exécutée, un processus est créé, qui ouvre trois flux

- stdin Entrée standard (clavier par défaut), identifié par 0
- stdout Sortie standard (écran par défaut), identifié par 1
- stderr Sortir d'erreur standard (écran par défaut), identifié par 2

Il est possible de rediriger les flux

- > Redirection de la sortie standard vers un fichier
- < Redirection de l'entrée standard
- >> Redirection de la sortie standard vers un fichier, avec concaténation
- >& Redirection de la sortie standard et de la sortie d'erreur standard
- >! Redirection avec écrasement de fichier
- | Redirection de la sortie standard vers l'entrée standard (*pipe* ou *tube*)

L'INTERPRÉTEUR

```
echo "Bonjour"
```
Imprime « Bonjour » sur l'écran

```
echo "Bonjour" > coucou.txt
```
Ecrit « Bonjour » dans le fichier « coucou.txt »

```
echo "Bonjour !" >> coucou.txt
```
Ajoute « Bonjour ! » dans le fichier « coucou.txt »

L'INTERPRÉTEUR – EXERCICE

Démarrer la machine avec le gestionnaire Hyper-V

S'identifier

Ecrire « Hello World » dans le fichier hello.txt

LES COMMANDES FONDAMENTALES

Il faut savoir que sous UNIX, on se déplace dans des répertoires

- Chemins absolus
  - Retour à la racine avec le « / » en tout premier lieu
- Chemins relatifs
  - On peut retourner au répertoire de l'utilisateur en cours en utilisant le tilde « ~ »
  - On peut poursuivre à partir du répertoire sur lequel on est positionné, en utilisant le point « ./ » ou sans rien utiliser
  - On peut remonter au répertoire parent en utilisant deux fois le point « ../ »

LES COMMANDES FONDAMENTALES

Aide

- **man**ual

```
man commande
man man
```

Déterminer le répertoire dans lequel vous êtes

- **p**rint **w**orking **d**irectory

```
pwd
```

Se déplacer dans l'arborescence

- **c**hange **d**irectory

```
cd
```

Exemples

`man pwd`	Aide sur la commande **pwd**
`pwd`	/home/ajc
`cd projets`	/home/ajc/projets
`cd ../`	/home/ajc
`cd /usr/local`	/usr/local

LES COMMANDES FONDAMENTALES – EXERCICE

Vérifier le répertoire en cours

Revenir à la racine

Revenir au répertoire précédent

- Soit avec un chemin relatif (relatif à l'utilisateur)
- Soit avec un chemin absolu

Vérifier le répertoire en cours

LES COMMANDES FONDAMENTALES

Certaines commandes ou programmes peuvent avoir des options

- Les options se précèdent généralement
  - D'un tiret « - » suivi d'une lettre (nom raccourcis de l'option)
  - De deux tirets « -- » suivi d'un mot (nom complet de l'option)
- Il est possible de faire suivre plusieurs options avec un seul tiret
  - -ab
  - -Rf

LES COMMANDES FONDAMENTALES

Lister le contenu d'un répertoire

- **lis**t

```
ls [options] [chemin]
```

```
ls -la
```
Liste les fichiers du répertoire courant avec des détails (à quoi correspondent-ils ?)

Visualiser le contenu d'un ou de plusieurs fichiers

- con**cat**enate

```
cat [options] <fichier> [fichiers]
```

LES COMMANDES FONDAMENTALES – EXERCICE

Lister tous les fichiers en liste du répertoire /dev

Ecrire cette liste de fichiers dans un fichier « fichiers.txt »

Imprimer le contenu de ce fichier

LES COMMANDES FONDAMENTALES

Visualiser le contenu d'un fichier avec une pagination (si celui-ci est trop long)

```
more [options] <fichier> [fichiers]
```

Visualiser un fichier avec pagination, navigation et recherche (dans un flux)

```
less [options] <fichier> [fichiers]
```

Editer un fichier

```
vi [options] <fichier>
```

```
nano [options] <fichier>
```

Statistiques sur un fichier

```
wc <option> <fichier>
```

```
wc -l fichier
```

Nombre de lignes d'un fichier

```
wc -c fichier
```

Nombre de caractères d'un fichier

LES COMMANDES FONDAMENTALES – EXERCICE

Créer un fichier « super.txt » avec nano (y mettre du contenu)

Enregistrer le fichier

Afficher le contenu des fichiers « super.txt » et « fichiers.txt »

LES COMMANDES FONDAMENTALES

Il est possible de donner à un autre programme le résultat de la commande

- Utilisation du *pipe* (ou *tube*) par le symbole trait vertical « | »

```
programme1 | programme2
```

- Ici, **programme2** va utiliser le résultat de **programme1** comme s'il s'agissait du contenu d'un fichier

LES COMMANDES FONDAMENTALES – EXERCICE

Avec les commandes vues précédemment

- Combien de fichiers et répertoires sont contenus dans le répertoire racine ?

Placer ce résultat dans un nouveau fichier « count.txt »

- Vérifier son contenu

LES COMMANDES FONDAMENTALES

Copier un fichier

- **cop**y

```
cp [options] <chemin_source> <chemin_destination>
```

```
cp mon_fichier.txt mon_fichier.bck.txt
```

Copie mon_fichier.txt et le nomme ce nouveau fichier mon_fichier.bck.txt

```
cp -R projets /tmp
```

Copie le répertoire « projets » le répertoire « /tmp »

LES COMMANDES FONDAMENTALES

Déplacer un fichier

- **mov**e

```
mv [options] <chemin_source> <chemin_destination>
```

```
mv mon_fichier.txt mon_fichier.bck.txt
```

Renomme « mon_fichier.txt » en « mon_fichier.bck.txt »

```
mv projets /tmp
```

Déplace le répertoire « projets » dans le répertoire « /tmp »

LES COMMANDES FONDAMENTALES – EXERCICE

Renommer « count.txt » en « count-files.txt »

LES COMMANDES FONDAMENTALES

Supprimer un fichier

- **rem**ove

```
rm [options] <chemin>
```

```
rm mon_fichier.txt
```

Supprime « mon_fichier.txt »

```
rm -r projets
```

Supprimer récursivement le répertoire « projets »

```
rm -f *.txt
```

Supprime tous les fichiers se terminant par « .txt », sans demander confirmation

LES COMMANDES FONDAMENTALES

Créer un répertoire

- **m**a**k**e **dir**ectory

```
mkdir <chemin>
```

```
mkdir projets
```

Crée un répertoire « projets »

LES COMMANDES FONDAMENTALES – EXERCICE

Créer un répertoire « projets »

Déplacer tous les fichiers .txt dans ce nouveau répertoire

Vérifier le contenu du répertoire projets et de son répertoire parent

LES COMMANDES FONDAMENTALES

Rechercher un fichier ou répertoire

- **find**

```
find <chemin> [options]
```

```
find . -name toto
```
Cherche dans le répertoire en cours et les sous-répertoires, un élément nommé « toto »

```
find . -type d -name toto
```
Cherche dans le répertoire en cours et les sous-répertoires, un répertoire nommé « toto »

```
find . -type f -name toto
```
Cherche dans le répertoire en cours et les sous-répertoires, un fichier nommé « toto »

LES COMMANDES FONDAMENTALES – EXERCICE

Rechercher uniquement les fichiers dont le nom se termine par « txt »

LES COMMANDES FONDAMENTALES

Rechercher un motif dans un fichier

- **g**eneral **r**egular **e**xpression **p**rocessor

```
grep [options] <expression régulière> <chemin>
```

```
grep "bonjour" /home/ajc/*
```

Cherche dans tous les fichiers du répertoire ceux qui contiennent « bonjour »

LES COMMANDES FONDAMENTALES – EXERCICE

Lister tous les fichiers avec leur détails du répertoire /dev

- Ne garder que ceux qui font référence à « sda »
- Enregistrer le résultat de cette recherche dans un fichier « disks.txt »

LES COMMANDES FONDAMENTALES – EXERCICE

Un fichier carnet contient plusieurs lignes sous la forme nom:prenom:telephone

- Compter le nombre de personnes
- Afficher toutes les lignes correspondant à « DUPONT »
- Afficher toutes les lignes ne correspondant pas à « DUPONT »
- En utilisant les commandes **head** et **tail**, afficher la 5ème ligne du fichier

LES COMMANDES FONDAMENTALES

Créer un lien

- **lin**k

```
ln [options] <chemin_source_absolu> <lien>
```

```
ln /home/ajc/toto.txt /home/ajc.projets/toto.txt
```
Crée un lien

```
ln -s /home/ajc/toto.txt /home/ajc.projets/toto.txt
```
Crée un lien symbolique

- Il existe deux types de liens
  - Lien physique (ou matériel)
    - Permet de donner plusieurs noms à un même fichier
    - Tant qu'un emplacement n'est pas supprimé, le fichier existe toujours
  - Lien symbolique
    - Permet de donner un autre point d'accès à un fichier
    - Lorsque la source est supprimée, tous les liens symboliques sont supprimés

LES COMMANDES FONDAMENTALES

Connaître des informations sur les processus (programmes en cours d'exécution)

- **p**rocess **s**tatus

```
ps [options]
```

Connaître l'activité du système

```
top [options]
```

Tuer un processus

- **kill**

```
kill [options] <PID>
```

```
kill 1157
```

Tue le processus identifié par le numéro PID 1157

LES COMMANDES FONDAMENTALES – EXERCICE

Exécuter la commande suivante avec une pagination

```
ps -aux
```

Chercher un processus qui s'appelle « bash »

- Le tuer en utilisant l'option « -9 »
- Que s'est-il passé ?

LES COMMANDES FONDAMENTALES

Arrêter le système

- **shutdown**

```
shutdown [options]
```

Redémarrer le système

- **reboot**

```
reboot [options]
```

LES COMMANDES FONDAMENTALES – EXERCICE

Redémarrer le système immédiatement avec la commande « shutdown »

LES COMMANDES FONDAMENTALES

Exécuter une commande en tant que super-administrateur

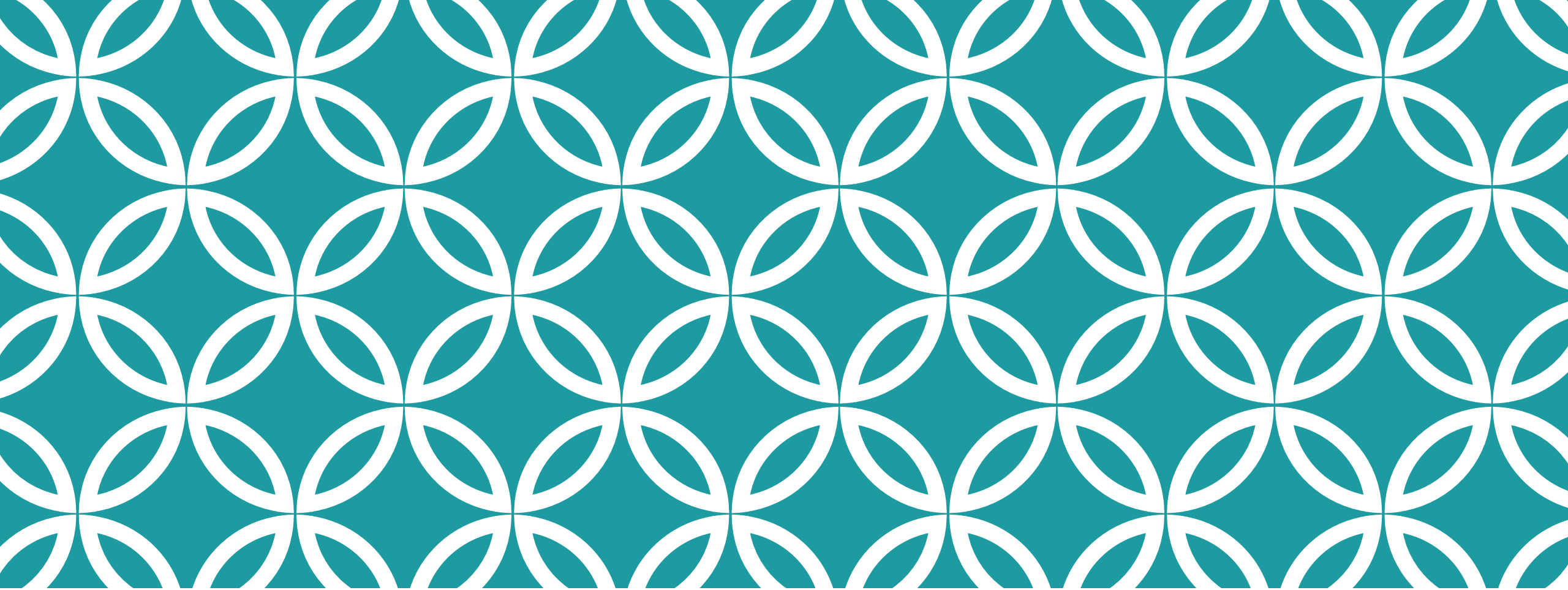
- **s**uper **u**ser **do**

```
sudo <commande>
```

Nettoyer la console

- **clear**

```
clear
```



ADMINISTRATION

Les commandes

LES COMMANDES D'ADMINISTRATION

Chaque fichier a des permissions

- Pour un utilisateur propriétaire (u)
- Pour un groupe (g)
- Pour les autres (o)
- Pour tout le monde (a)

Chaque permission a trois actions

- Lecture
- Ecriture
- Exécution

#	Définition	rwx
0	Aucune permission	---
1	Exécution	--x
2	Ecriture	-w-
3	Ecriture et exécution (1+2)	-wx
4	Lecture	r--
5	Lecture et exécution (1+4)	r-x
6	Lecture et écriture (2+4)	rw-
7	Lecture, écriture et exécution (1+2+4)	rwx

Un fichier aura par conséquent 3 numéros

- 777 Accès à tout, par tout le monde
- 644 Seul le propriétaire peut lire et écrire, les autres peuvent lire, personne ne peut exécuter
- 660 Le propriétaire et le groupe ont des droits en lecture et écriture

LES COMMANDES D'ADMINISTRATION

Modifier les droits d'un fichier

- **ch**ange **mod**e

```
chmod [options] <droits> <fichier> [fichiers]
```

`chmod +x script.jar`	Autorise le fichier en exécution pour tout le monde
`chmod 644 script.jar`	Le fichier n'est autorisé à être exécuté par personne
`chmod -R 777 projets`	Accès complet au répertoire projets, de façon récursive
`chmod o-x script.jar`	Retire le fichier en exécution, seulement pour les autres

LES COMMANDES D'ADMINISTRATION

Modifier le propriétaire d'un fichier

- **ch**ange **own**er

```
chown [options] <utilisateur> <fichier> [fichiers]
```

```
chown jeremy script.jar
```

Le nouveau propriétaire est maintenant jeremy

Modifier le groupe d'un fichier

- **ch**ange **grou**p

```
chgrp [options] <groupe> <fichier> [fichiers]
```

```
chgrp -R formateurs cours
```

Le nouveau groupe du répertoire (et de tout son contenu) est formateurs

LES COMMANDES D'ADMINISTRATION

Ajouter un utilisateur

- **user add**

```
useradd [options] <login>
```

Ajouter un groupe

- **group add**

```
groupadd [options] <group>
```

LES COMMANDES D'ADMINISTRATION

Modifier un utilisateur

- **user mod**e

```
usermod [options] <login>
```

```
usermod -a -G formateurs julie
```
Ajoute le groupe (-a comme append) « formateurs » à l'utilisateur « julie »

```
usermod -G formateurs julie
```
Modifier le groupe de julie

Modifier le mot de passe d'un utilisateur

- **passwor**d

```
passwd [options] <login>
```

LES COMMANDES D'ADMINISTRATION

Supprimer un groupe

- **group de**lete

```
groupdel [options] <group>
```

Supprimer un utilisateur

- **user del**ete

```
userdel [options] <login>
```

LES COMMANDES D'ADMINISTRATION

Les utilisateurs sont contenus dans le fichier

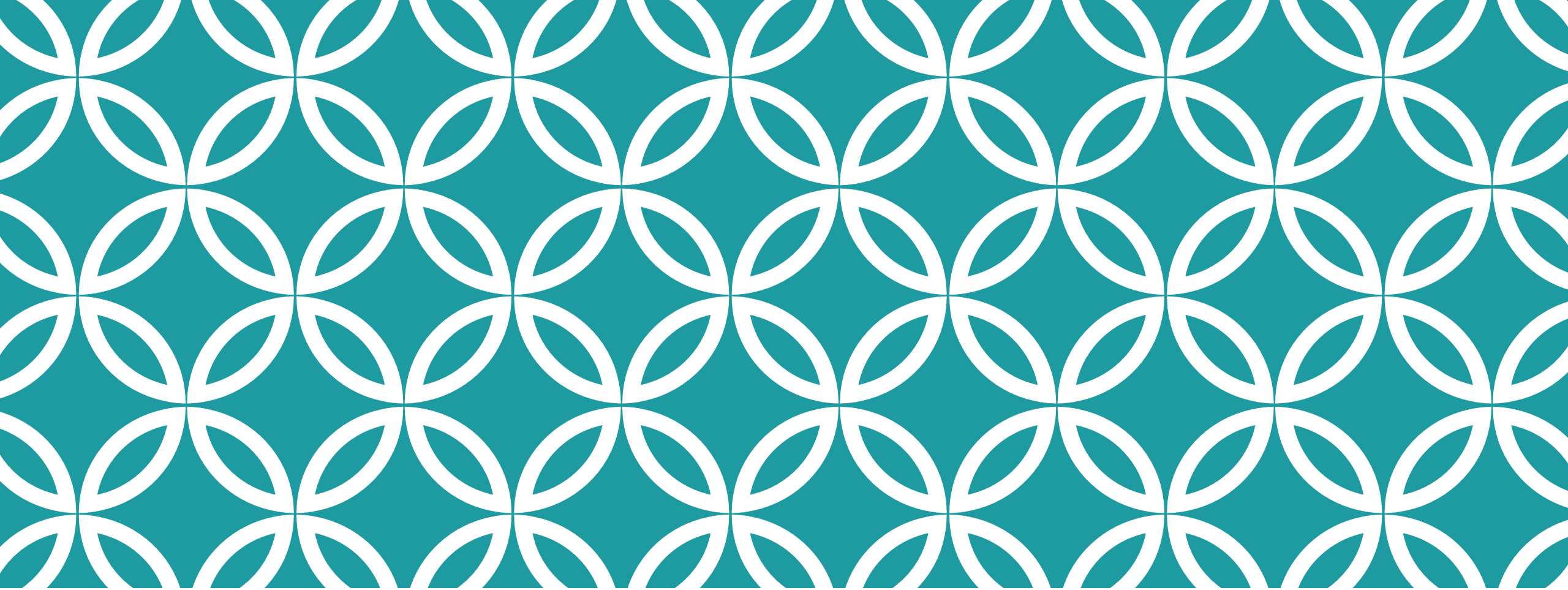
- /etc/passwd

Les mots de passe sont contenus dans le fichier

- /etc/shadow

Les groupes sont contenus dans le fichier

- /etc/group



SCRIPTS

Scripting shell

SCRIPTING

Avec le shell

- On exécute des commandes
- On exécute des instructions de programmation (conditions, boucles, …)

Le scripting est l'association de commandes et d'instructions dans un fichier

- Ce fichier doit être autorisé en exécution

Chaque script doit commencer par un « shebang »

- Permet d'indiquer quel interpréteur utiliser
- Commence par « #! » et est suivi de l'interpréteur

```
#!/bin/bash
#!/bin/sh
#!/bin/csh
```

SCRIPTING

```
nanoe prenom.bash
```

Editeur de texte pour le fichier « prenom.bash »

```
#!/bin/bash

echo "Indiquer le prénom"
read prenom
echo "Bonjour $prenom !"
```

Contenu du script

```
chmod u+x prenom.bash
```

Autorise l'exécution du fichier pour l'utilisateur propriétaire

```
./prenom.bash
```

Exécute le script

SCRIPTING – EXERCICE

Ecrire un programme *bash* « birthdate »

- Qui attend un prénom
- Qui attend un âge
- Ecrire le prénom et l'année de naissance dans un fichier « users.txt »
  - NOTE : Pour réaliser des calculs : $((18 - $var))

Vérifier le contenu du fichier « users.txt » après exécution

Supprimer le fichier « users.txt »

SCRIPTING – EXERCICE

Créer un programme JAVA « Hello You »

- Afficher le texte « Saisir le prénom »
- Afficher le texte « Hello prenom »

- Le compiler avec la ligne de commande `javac HelloYou.java`
- L'exécuter avec la ligne de commande `java HelloYou`

Créer un script *bash* qui exécute le programme JAVA

NB : Pour plus de facilité, contrôler la machine en SSH avec Putty

SCRIPTING

Pour exécuter un script, il faut être dans le répertoire dans lequel il se trouve

- SAUF s'il est présent dans le répertoire /usr/local/bin !

SCRIPTING – EXERCICE

Créer un lien symbolique du script *bash* pour l'exécuter comme une commande classique dans le répertoire /usr/local/bin