Université Nice Sophia Antipolis

Parcours des écoles d'ingénieurs Polytech (PeiP1)

Les Boitiers de Vote Electroniques (Clickers)

Pour dynamiser et rendre interactif un cours en amphi

Qu'est ce qu'un Boitier de Vote ?

- Une télécommande comme pour la télévision
  - Communication sans-fil
  - Transmission bidirectionnelle
    - Envoi au boitier des réponses possibles (« sens interdit » si réponse pas prévue)
    - Envoi par le boitier de la réponse et du code d'identification boitier
- Vous allez utiliser le boitier
  - Pour transmettre votre réponse aux questions
  - Possibilité de répondre anonymement ou pas (défini par le récepteur)
- Résultats
  - « Qui (ou pas) » répond « Quoi » et « Quand »

Pourquoi utiliser les Boitiers de Vote ?

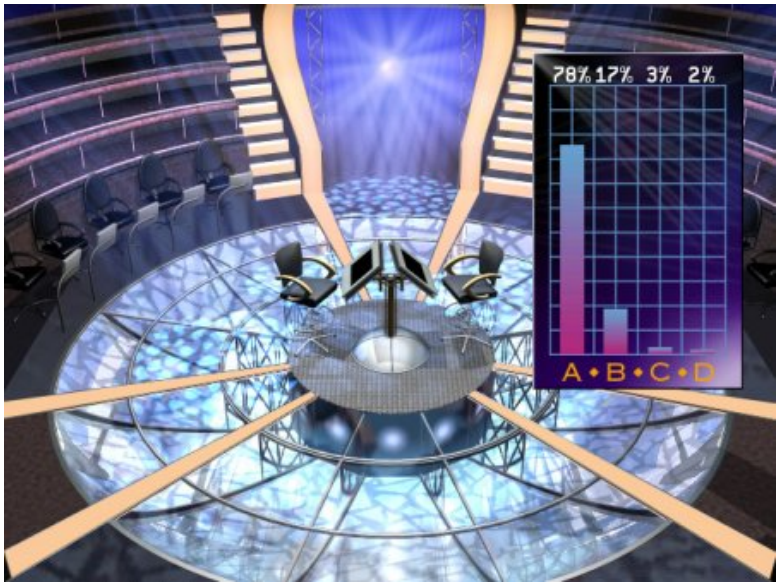
- Introduction de la technologie dans les cours pour :
  - Dynamiser un cours avec un large public
  - Provoquer la réflexion et le débat, même en grand nombre
- Utilisé outre-Atlantique depuis 20 ans
  - Canada et Etats-Unis
  - Dans les plus grandes universités: MIT, CMU, Berkley, …
  - Dans de nombreuses disciplines
    - Mathématiques, Physique, Chimie, Biologie
    - Droit, Economie, Gestion, …
- « *Serious Game* »
- Meilleurs résultats… alors pourquoi pas ?

Configuration du Boitier de Vote

- Retirer les caches plastiques pour les piles
- Réglage du canal de communication :
  - Appuyer sur Channel
  - Composer le numéro de canal
  - Appuyer sur Channel
  - La LED du boitier devient verte
- Votre boitier est prêt ?
- Nous pouvons faire un premier test

Un jeu sérieux (Serious Game)

- Ce cours n'est pas une évaluation
  - Vous aurez deux évaluations à venir
    - Examen sur table: questions ouvertes, QCM, problèmes, … tout est possible
    - Portera sur tout ce que l'on a vu depuis le début
- Mais un jeu sérieux
  - On va travailler et réfléchir
- Et une compétition amicale
  - Compétition individuelle
  - Vous pouvez communiquer entre vous.



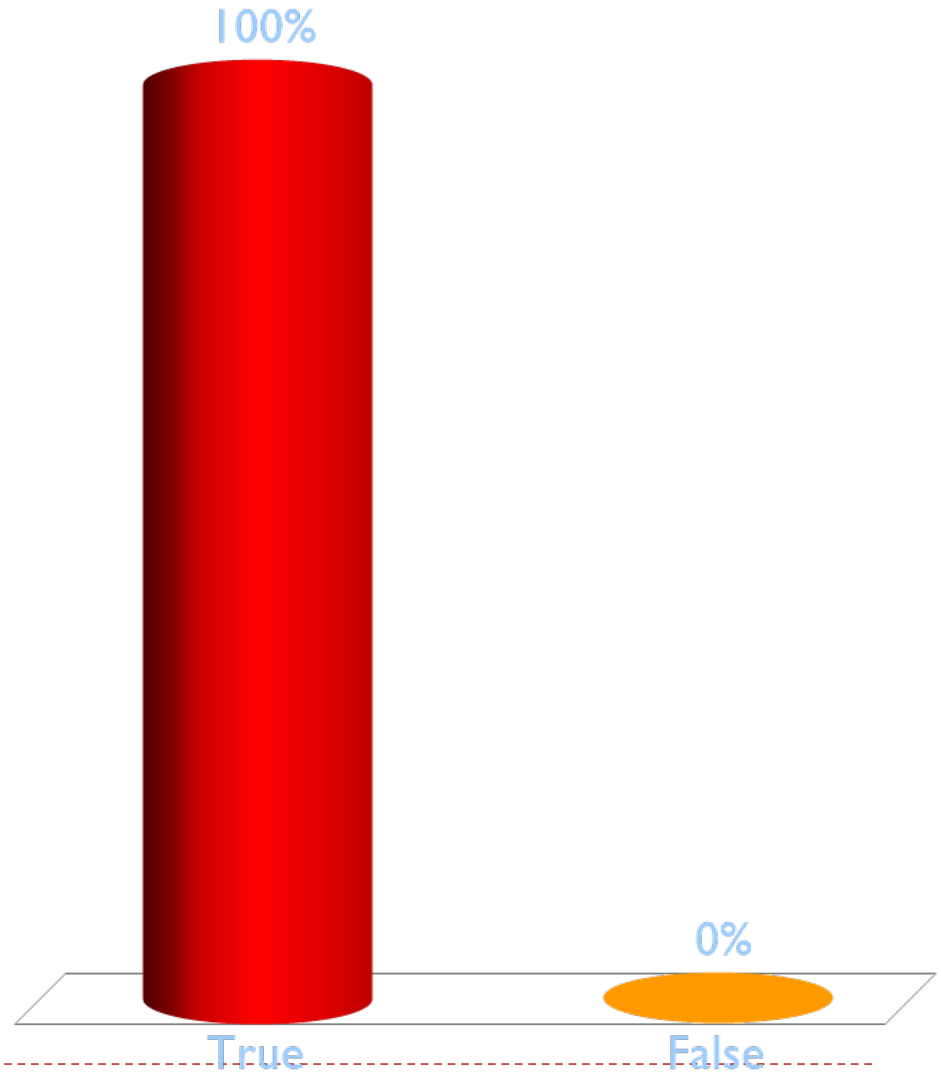
Est-ce que vous êtes prêts ?

- Etes-vous prêts ?

1. True
2. False

- Combien ont voté ?

1 sur 120



Université
Nice
Sophia Antipolis

Environnement Informatique 1

Objectifs du cours

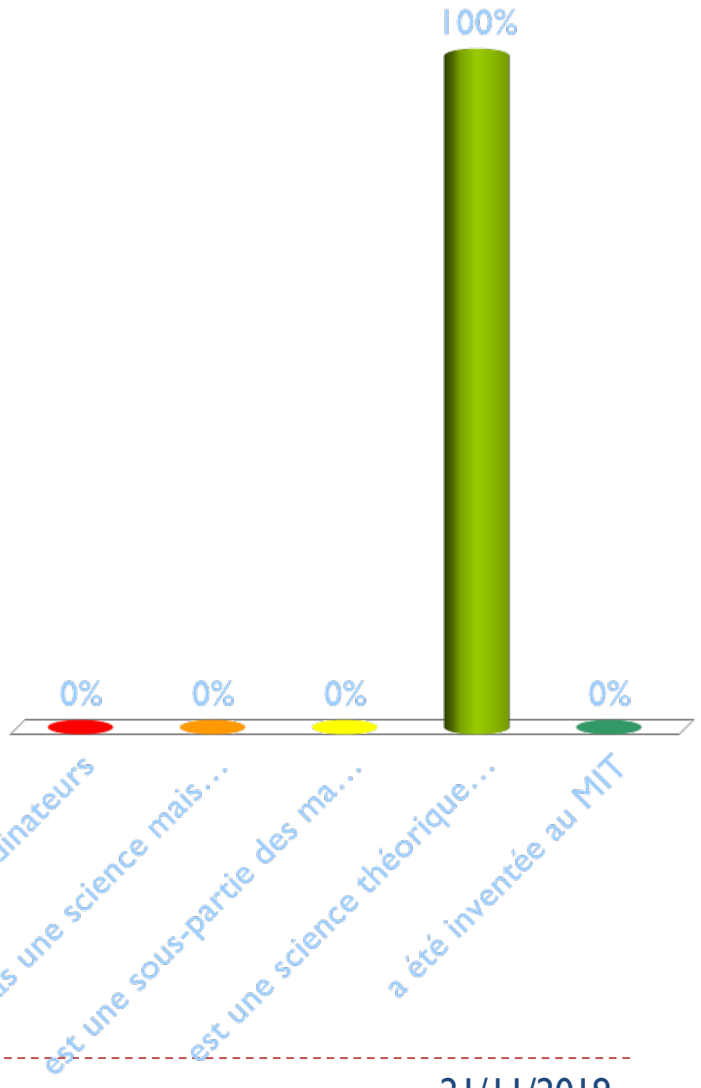
Capacités et compétences d'un ingénieur

- Extrait de la définition par la CTI :
  - L'acquisition des connaissances scientifiques et techniques et la maîtrise de leur mise en œuvre
    - La connaissance et la compréhension d'un large champ de sciences fondamentales et la capacité d'analyse et de synthèse qui leur est associée.
    - L'aptitude à **mobiliser les ressources d'un champ scientifique et technique liées à une spécialité**.
    - La maîtrise des méthodes et des outils de l'ingénieur : identification, modélisation et résolution de problèmes même non familiers et non complètement définis, **l'utilisation des outils informatiques**, l'analyse et la conception de systèmes.
    - La maîtrise de l'expérimentation, dans un contexte de recherche et à des fins d'innovation et **la capacité d'en utiliser les outils: notamment la collecte et l'interprétation de données**, la propriété intellectuelle.
  - L'adaptation aux exigences propres de l'entreprise et de la société
  - La prise en compte de la dimension organisationnelle, personnelle et culturelle

Qu'est ce qui est vrai dans les affirmations suivantes ?

- L'informatique…

1. est la science des ordinateurs
2. n'est pas une science mais de la technologie
3. est une sous-partie des mathématiques
4. est une science théorique et appliquée
5. a été inventée au MIT



Environnement Informatique 1

- Buts du module
  - Se familiariser avec les environnements informatiques
    - Unix (GNU/Linux) et Windows (Windows 7)
    - Savoir installer le système dont on a besoin sur sa machine
    - Maîtriser l'utilisation des machines virtuelles
  - Utiliser l'outil informatique autrement
    - Apprentissage des commandes de base d'un système
    - Pour automatiser des traitements sur des fichiers et données
  - Utiliser et comprendre l'outil informatique en réseau
    - Comment fonctionne un ordinateur en réseau
  - Voir des concepts fondamentaux de la science informatique
    - Codage de l'information
  - Maîtriser les outils bureautique (C2i niveau 1)

Université Nice Sophia Antipolis

Commandes, Fichiers, Répertoires et Permissions

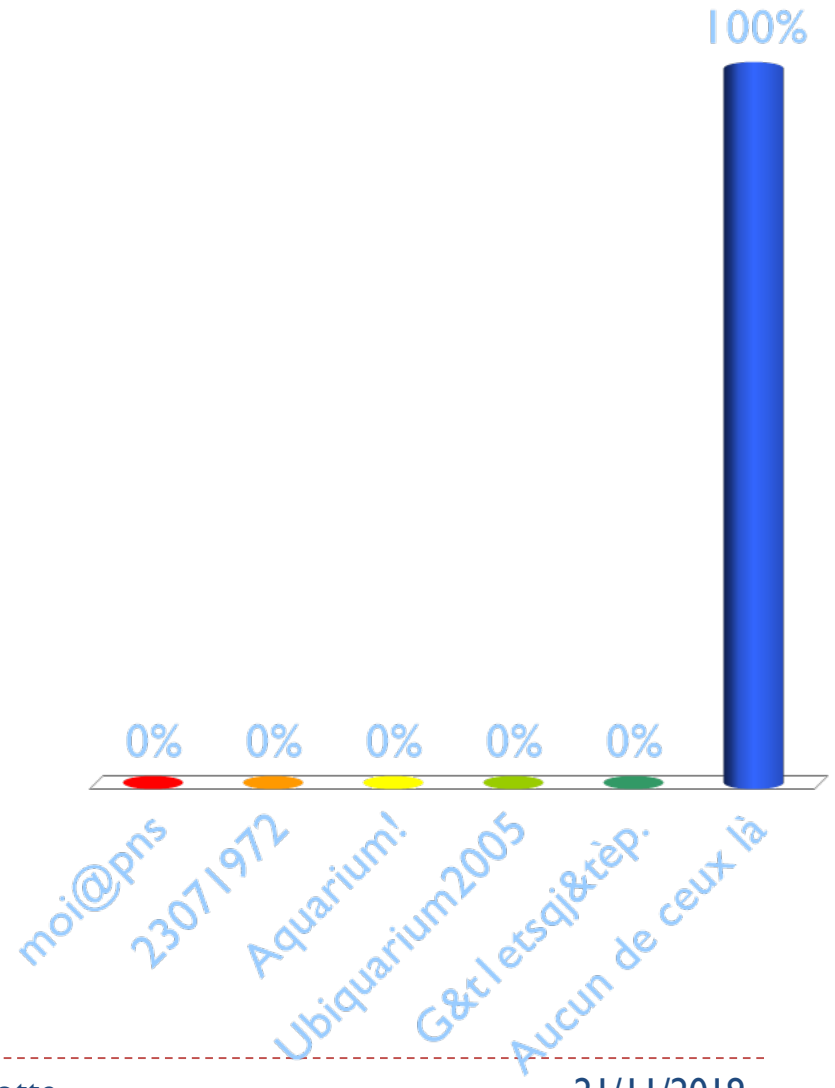
Environnement Informatique 1

Configuration des comptes utilisateurs

- Après l'installation de la machine
  - Création et configuration du compte sur la machine
  - Connexion au réseau (wifi)
  - Configuration des comptes en ligne
    - Espace Numérique de Travail (ENT), configuration sésame, adr mail, …
    - Bureau Virtuel (BV): accès à la boîte mail, agenda, échange doc, …
    - Inscription à Moodle: rendu des TDs
- Pour tous ces comptes
  - Besoin d'un mot de passe
  - Bien vivre avec quelques mots de passe…

Quel mot de passe utiliseriez-vous ?

1. **moi@pns**
2. **23071972**
3. **Aquarium!**
4. **Ubiquarium2005**
5. **G&t1etsqj&tèp.**
6. Aucun de ceux là

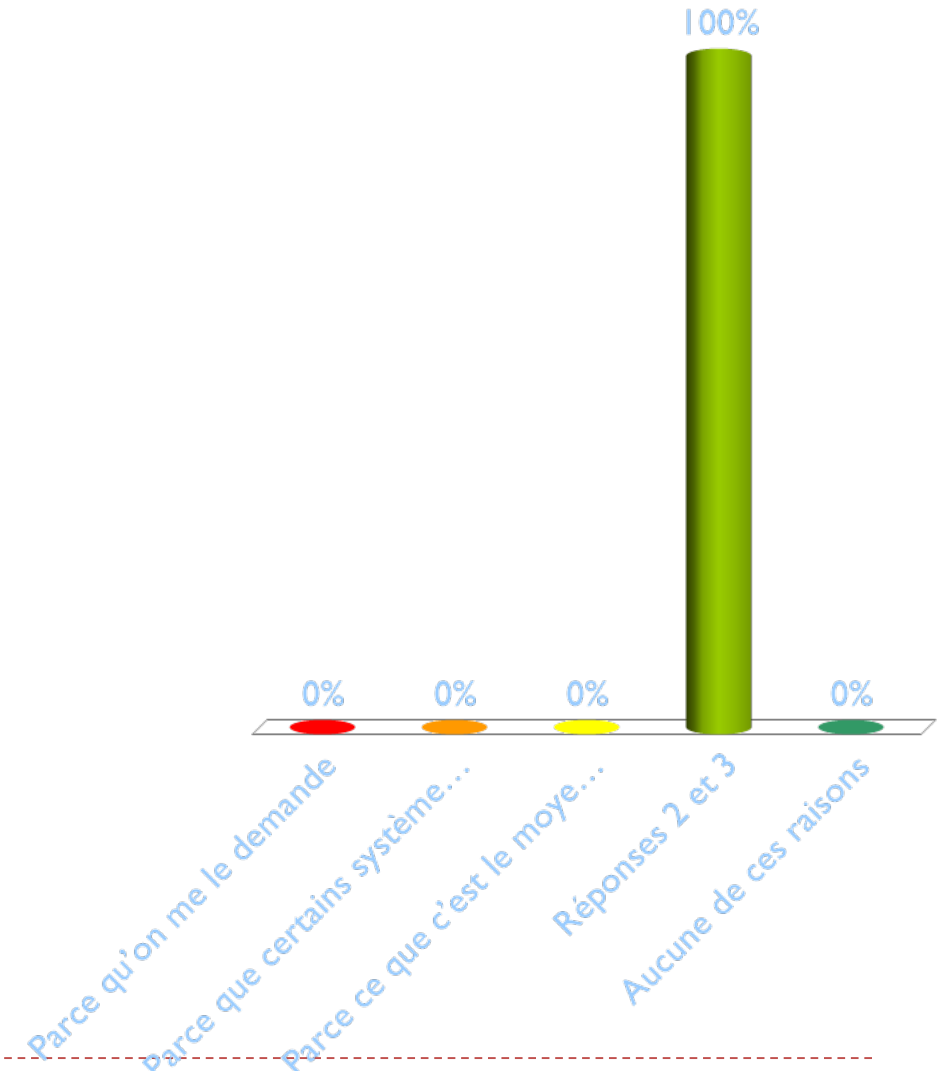


Commandes sous Unix

- Format d'une commande
  - Forme Unix des commandes : `cmd -opt1 --opt2 arg1 arg2 arg3 …`
- Quelques commandes pour bien débuter sous Unix
  - **apropos** *subjet*
  - **man** *cmd*
  - **cd** *chemin*
  - **ls** –l –a
  - **mkdir**
  - **rmdir**
  - **cp**
  - **mv**
  - **rm**
  - **touch** *fichier*
  - **cat** *fichier*
  - **pwd**
- Et quelques autres pour bien continuer
  - **cut**
  - **tr**
  - **wc**
  - **sort**
  - **uniq**

Mais pourquoi utiliser des commandes ?

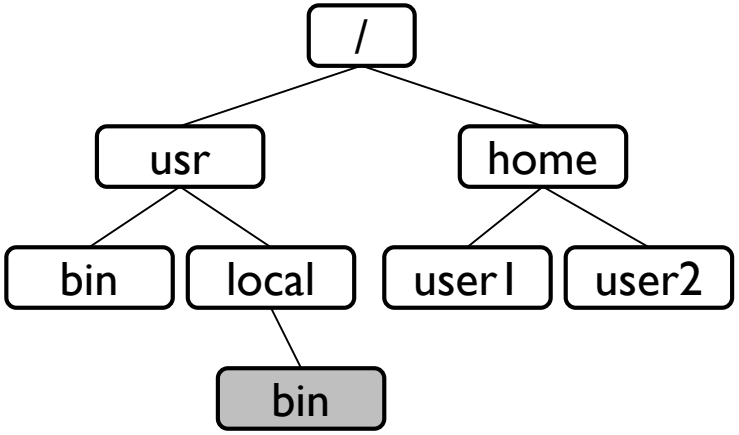
1. Parce qu'on me le demande
2. Parce que certains systèmes n'ont pas d'interface graphique
3. Parce ce que c'est le moyen d'automatiser des traitements
4. Réponses 2 et 3
5. Aucune de ces raisons



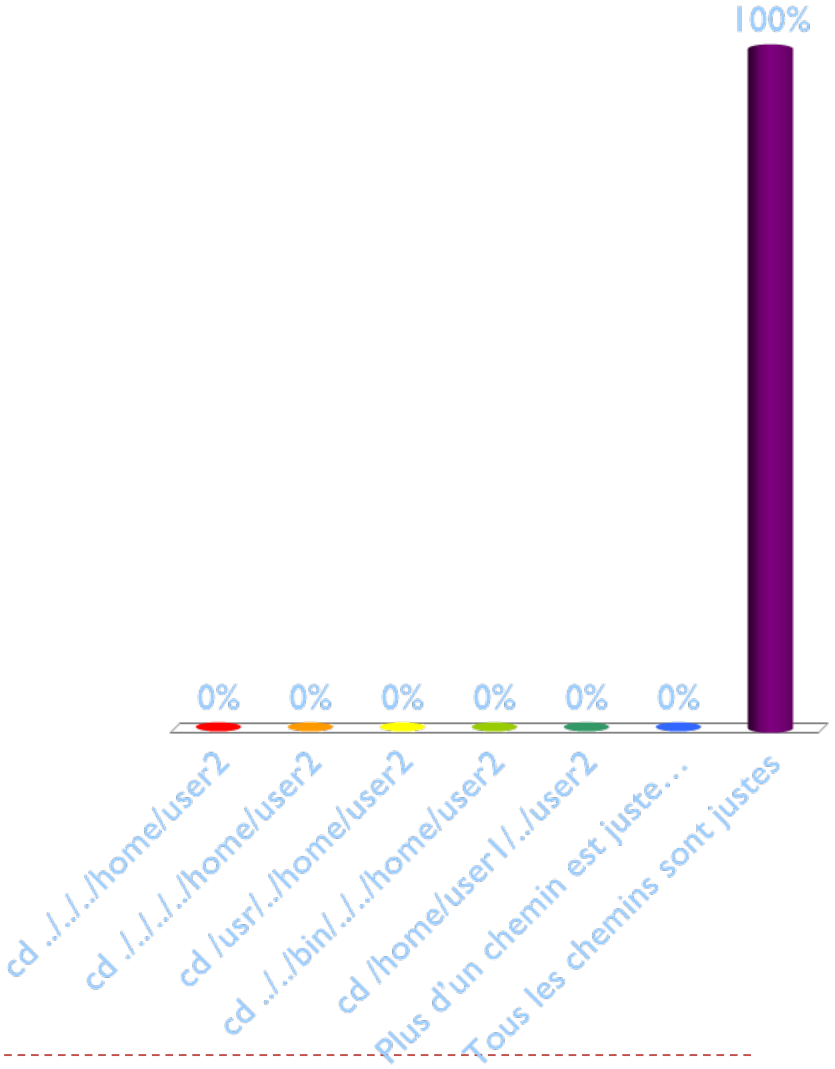
Qu'est ce qu'un chemin ?

- Racine du système
  - Une racine unique sous Unix: /
- Dossiers/Répertoires
  - Répertoire courant : .
  - Répertoire parent : . .
  - Répertoire personnel : ~
- Deux types de chemins
  - Chemin relatif
  - Chemin absolu

Quel chemin vous permet de vous rendre dans le répertoire /home/user2 ?

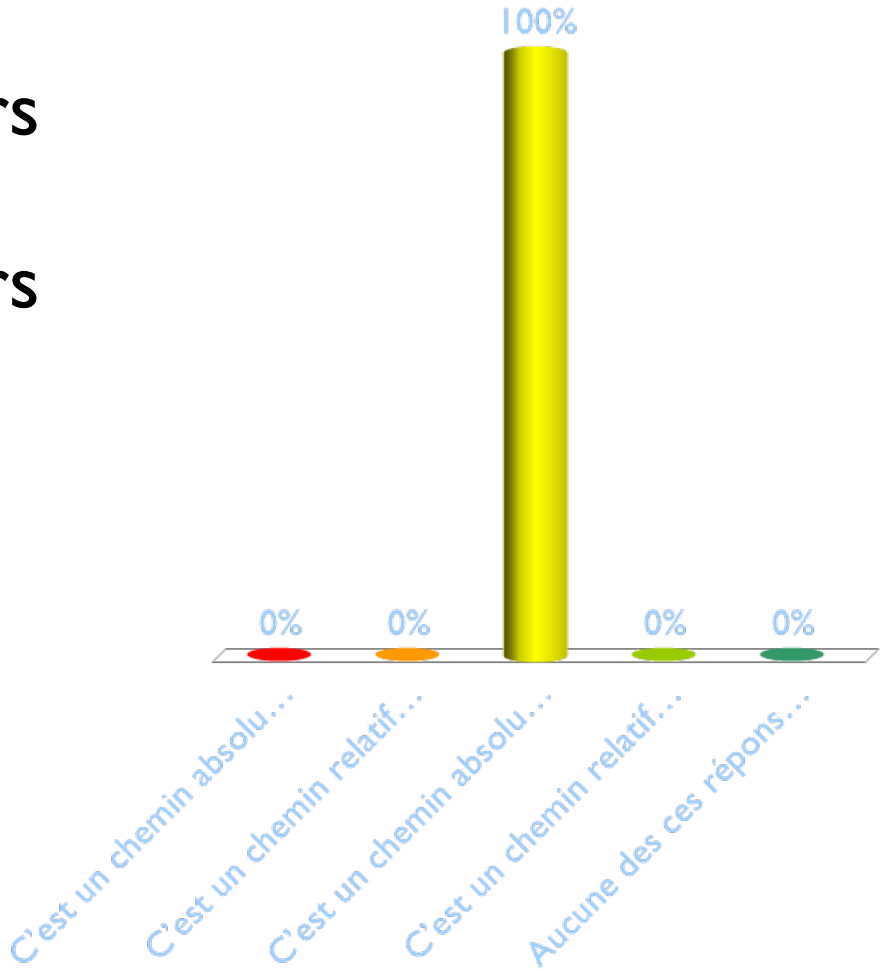


1. cd ../../../home/user2
2. cd ./../../../home/user2
3. cd /usr/../home/user2
4. cd ../../bin/../../home/user2
5. cd /home/user1/../user2
6. Plus d'un chemin est juste mais pas tous
7. Tous les chemins sont justes



Sous Unix, est que ~ est un chemin relatif ou absolu ?

1. C'est un chemin absolu pour tous les utilisateurs
2. C'est un chemin relatif pour tous les utilisateurs
3. C'est un chemin absolu pour un utilisateur donné
4. C'est un chemin relatif pour un utilisateur donné
5. Aucune des ces réponses n'est juste



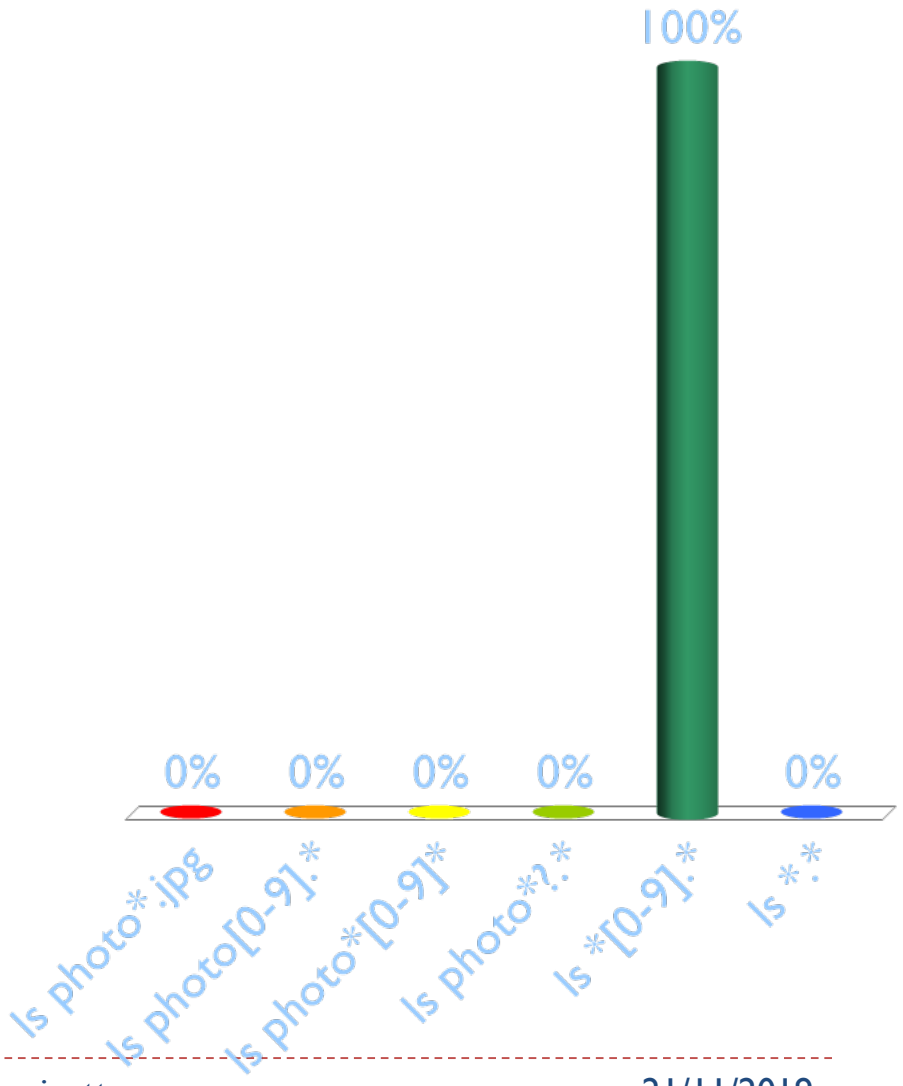
Les jokers ou motifs dans les chemins

- L'interpréteur de commandes (ou shell)
  - Interprète certains caractères spéciaux (?, *, [)dans les arguments avant de lancer la commande
  - Ces caractères sont remplacés par les fichiers ou répertoires qui peuvent correspondre au motif spécifié
- Les caractères spéciaux sont remplacé par
  - ? : n'importe quel caractère
  - * : une suite quelconque de 0, I ou n caractères
  - [*abc*] : un des caractères spécifiés dans les crochets
  - [^*abc*] ou [!abc] : n'importe quel caractère sauf un des caractères spécifiés dans les crochets
  - [a-z] : n'importe quel caractère dans la plage entre a et z
  - [:classe:] : un caractère de la classe (alnum, digit, lower, punct, space, upper, …)

Quelle expression ne permet de lister que les fichiers contenant un chiffre avant l'extension ?

- `photo1.jpg`
- `photo1-moi.jpg`
- `photo2013-10-04.jpg`
- `photo2.png`
- `photoA.gif`

1. ls photo*.jpg
2. ls photo[0-9].*
3. ls photo*[0-9]*
4. ls photo*?.*
5. ls *[0-9].*
6. ls *.*

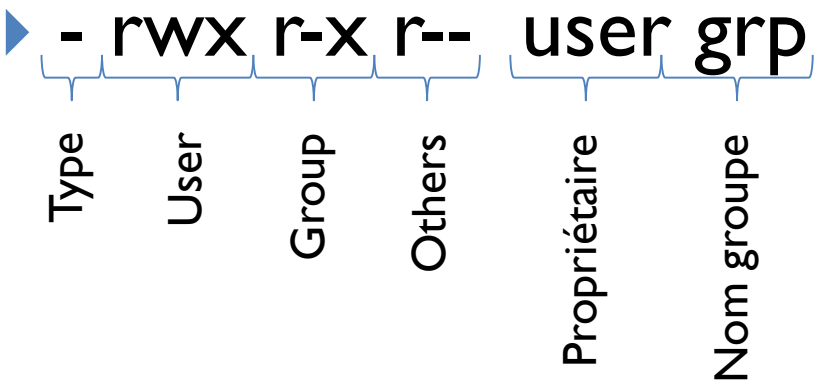


Fichiers et Dossiers

- Fichier
  - Fichier de données (information non exécutables)
  - Fichier programme (exécutable sur l'ordinateur)
- Dossier
  - Regrouper des fichiers
  - Pour organiser, regrouper les fichiers et ne pas tout avoir en vrac
- Lien
  - Physique
  - Symbolique

Les permission sur les fichiers

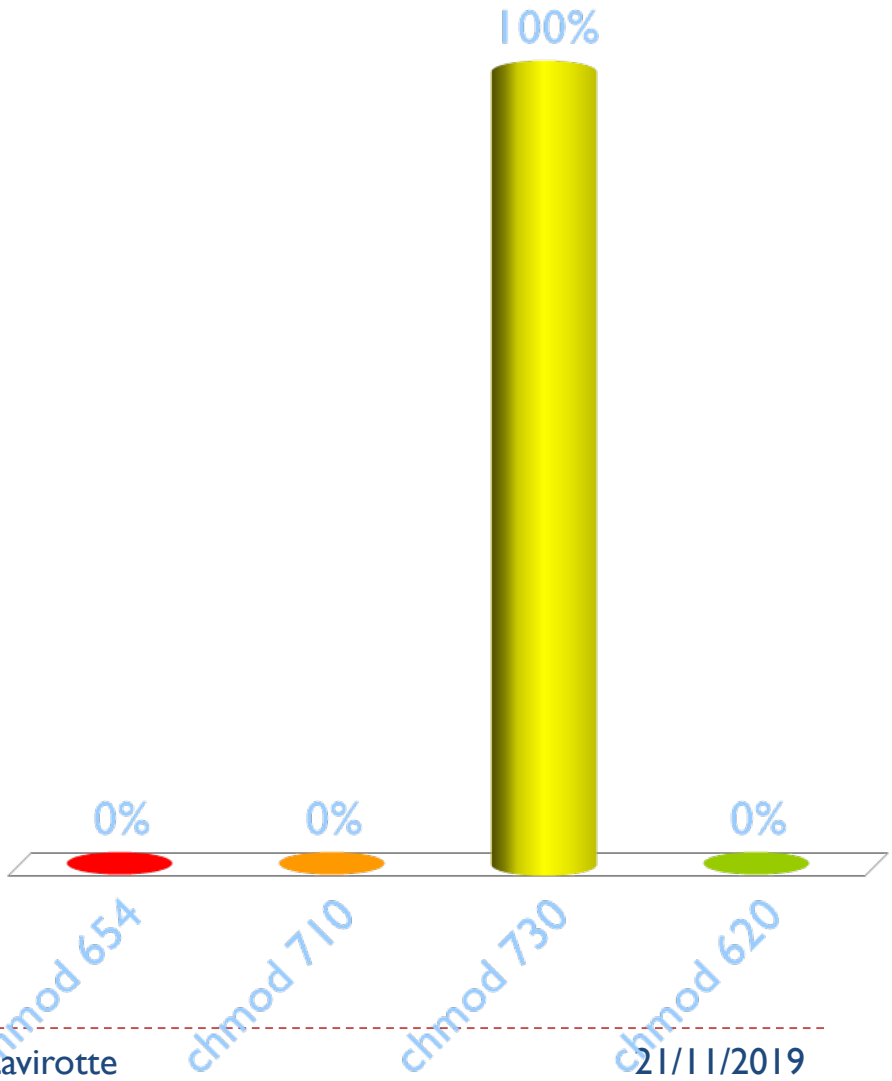
- Les fichiers possèdes des droits
  - Pour autoriser ou pas certaines catégories d'utilisateurs à faire certaines actions
  - Catégories
    - U: utilisateur (User), l'utilisateur à qui appartient le fichier
    - G: groupe, (Group) le groupe auquel appartient le fichier
    - O: autres (Others), tous les autres utilisateurs
  - Permissions:
    - R: lecture (Read)
    - W: écriture (Write)
    - X: exécution (eXecute)



Quelle est la commande équivalente que l'on aurait pu utiliser ?

- Soit un fichier avec les droits :
  - `-rwxrwxr-x user1 group1`
- On exécute la commande
  - `chmod a-rw,u+rw,g+w,o-x`
- Quelle commande arrive au même résultat ?

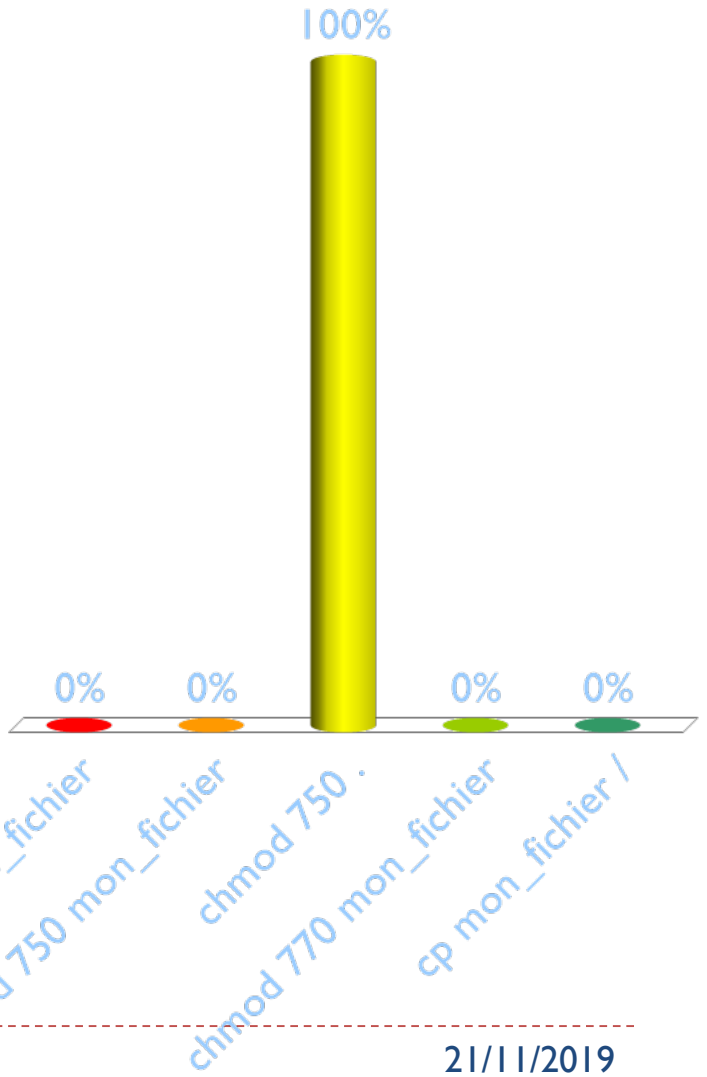
1. chmod 654
2. chmod 710
3. chmod 730
4. chmod 620



Quelle commande permet d'empêcher les gens de mon groupe de supprimer un fichier ?

- Soit un fichier `mon_fichier`. Je me trouve dans le dossier contenant ce fichier.
- Quelle commande permet d'empêcher sa suppression par les gens de mon groupe ?

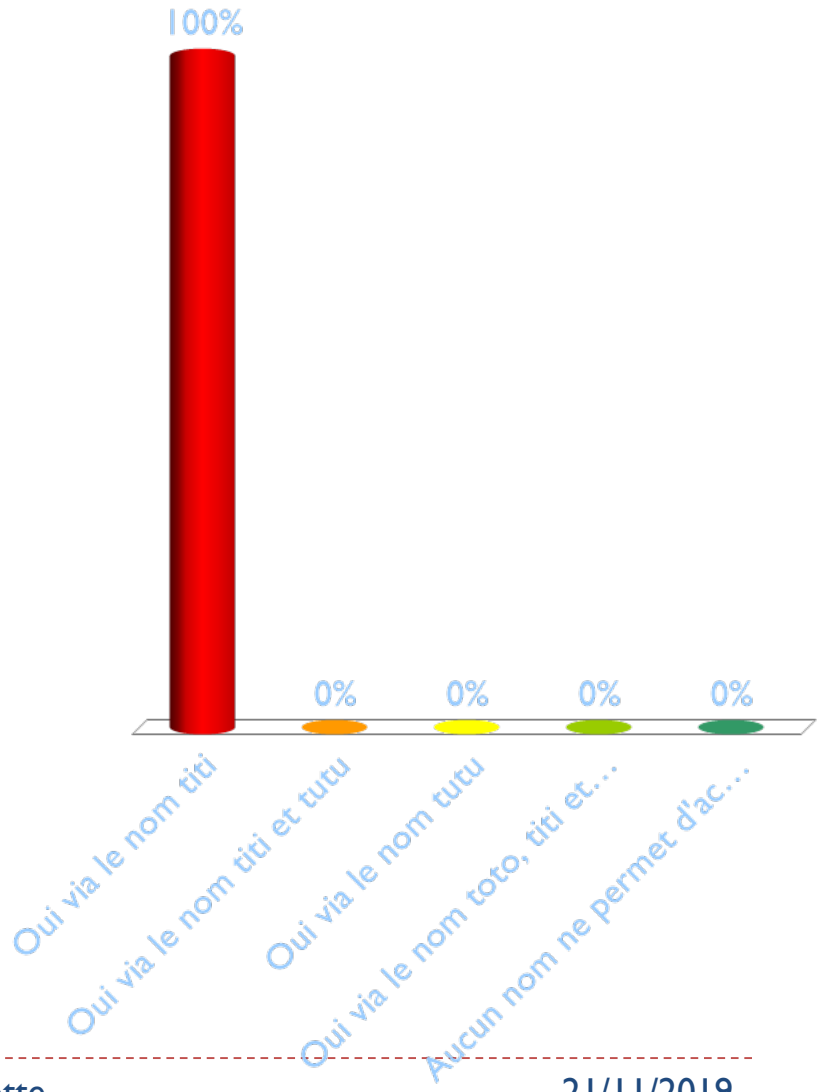
1. chmod g-w mon_fichier
2. chmod 750 mon_fichier
3. chmod 750 .
4. chmod 770 mon_fichier
5. cp mon_fichier /



Puis-je toujours accéder au contenu du fichier créé au départ ?

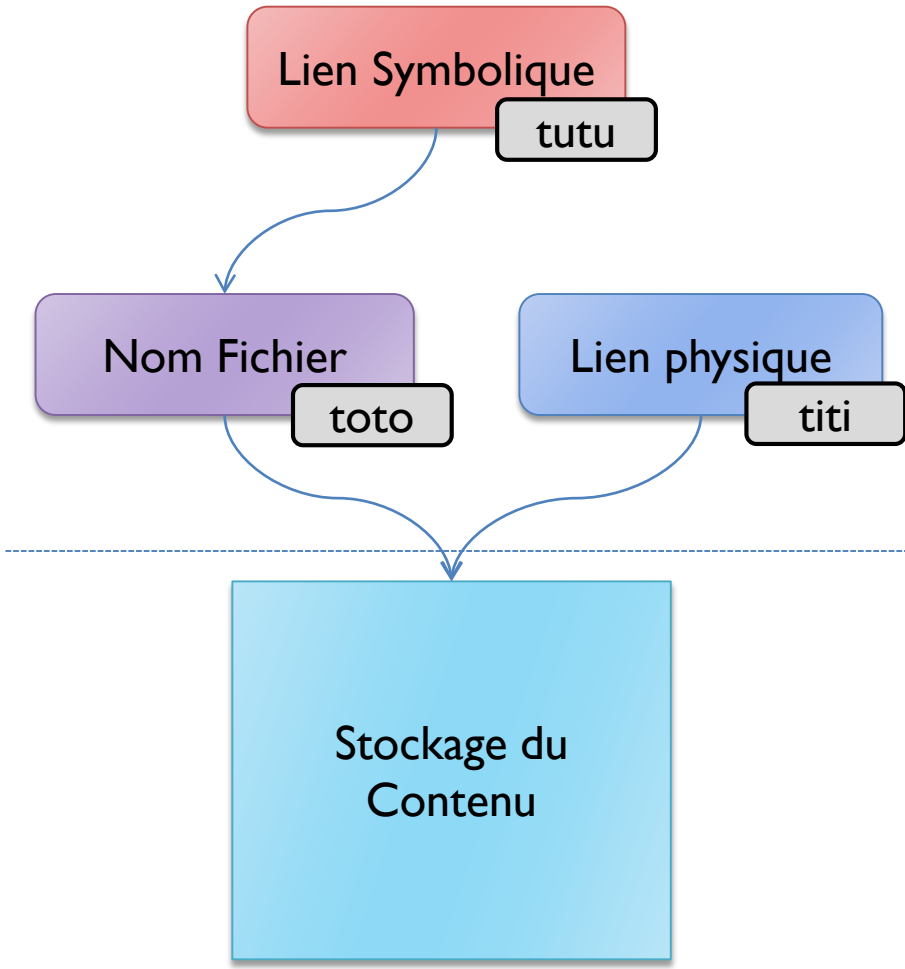
- Soient les commandes suivantes :
  - `touch toto`
  - `ln toto titi`
  - `ln -s toto tutu`
  - `rm toto`

1. Oui via le nom titi
2. Oui via le nom titi et tutu
3. Oui via le nom tutu
4. Oui via le nom toto, titi et tutu
5. Aucun nom ne permet d'accéder au contenu



Représentation schématique des liens physiques et symboliques

- Représentation schématique
  - on ne descend pas jusqu'à la représentation du contenu
- Soient les commandes suivantes :
  - `touch toto`
  - `ln toto titi`
  - `ln -s toto tutu`
  - `rm toto`

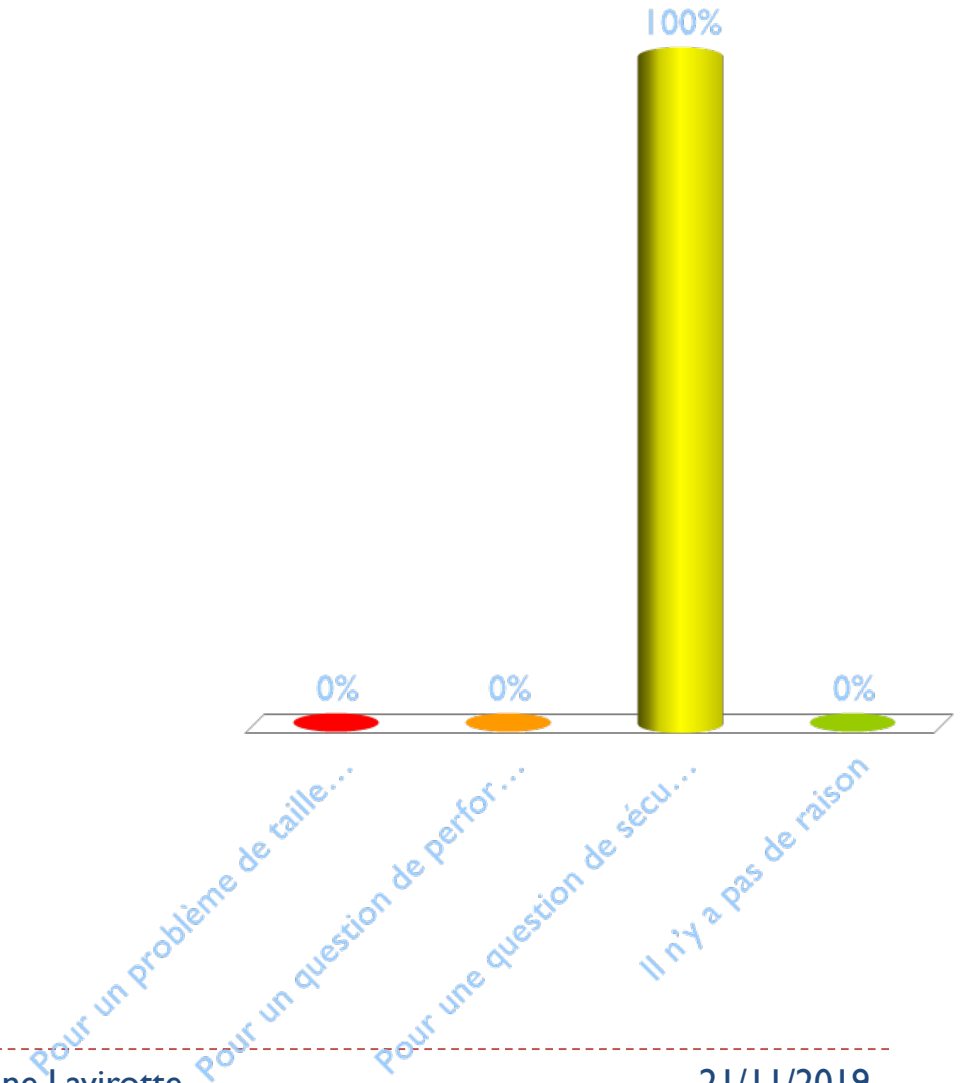


Gestion des Utilisateurs

- Possibilité d'avoir plusieurs utilisateurs sur un système
  - Création des utilisateurs: `useradd` et `adduser`
  - Suppression des utilisateurs: `userdel` et `deluser`
  - Information sur les utilisateurs dans `/etc/passwd`
- Création ou pas du dossier personnel (voir options)
- Information sur les utilisateurs stockées dans:
  - `/etc/passwd` : liste les utilisateurs et quelques informations
  - `/etc/shadow` : contient les mots de passe encryptés
  - Attention: ne pas modifier les permissions sur `/etc/passwd` et `/etc/shadow` !
    - `/etc/passwd` doit être en lecture pour tous !
    - `/etc/shadow` ne doit pas être en lecture pour tous (seulement root et les utilisateurs du groupe shadow)

Pourquoi ne pas avoir mis toutes les informations utilisateurs dans un fichier?

1. Pour un problème de taille de fichier
2. Pour un question de performance du système
3. Pour une question de sécurité
4. Il n'y a pas de raison



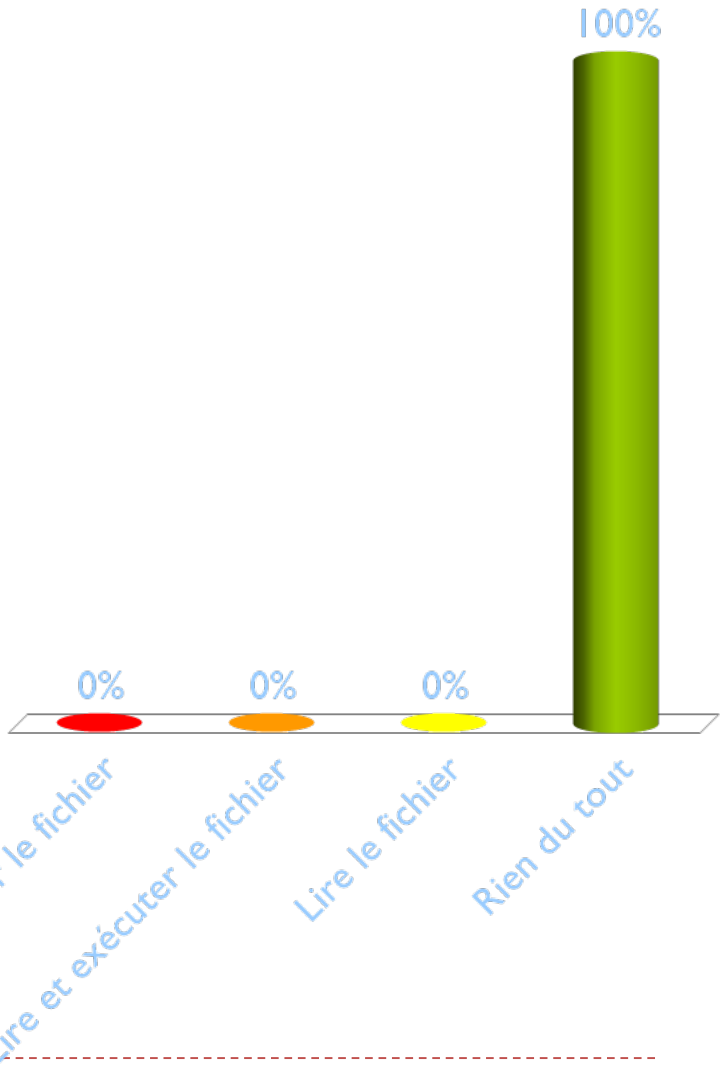
Gestion des Groupes

- Possibilité de rattacher un utilisateur à un ou plusieurs groupes:
  - Créer un groupe: `groupadd` et `addgroup`
  - Supprimer un groupe: `groupedel` ou `delgroupe`
  - Rattacher un utilisateur à un groupe: modifier le fichier `/etc/group`
- Informations sur les groupes stockées dans:
  - `/etc/group`: liste des groupes et des utilisateurs dans les groupes
  - `/etc/gshadow`: contient les infos cachées des groupes
  - Attention: ne pas modifier les permissions sur `/etc/group` et `/etc/gshadow` !

Que puis-je faire sur le fichier suivant ?

- `-rw-r-x--- 1 actor movie film.mkv`
- Je suis l'utilisateur walle qui appartient au groupe pixar

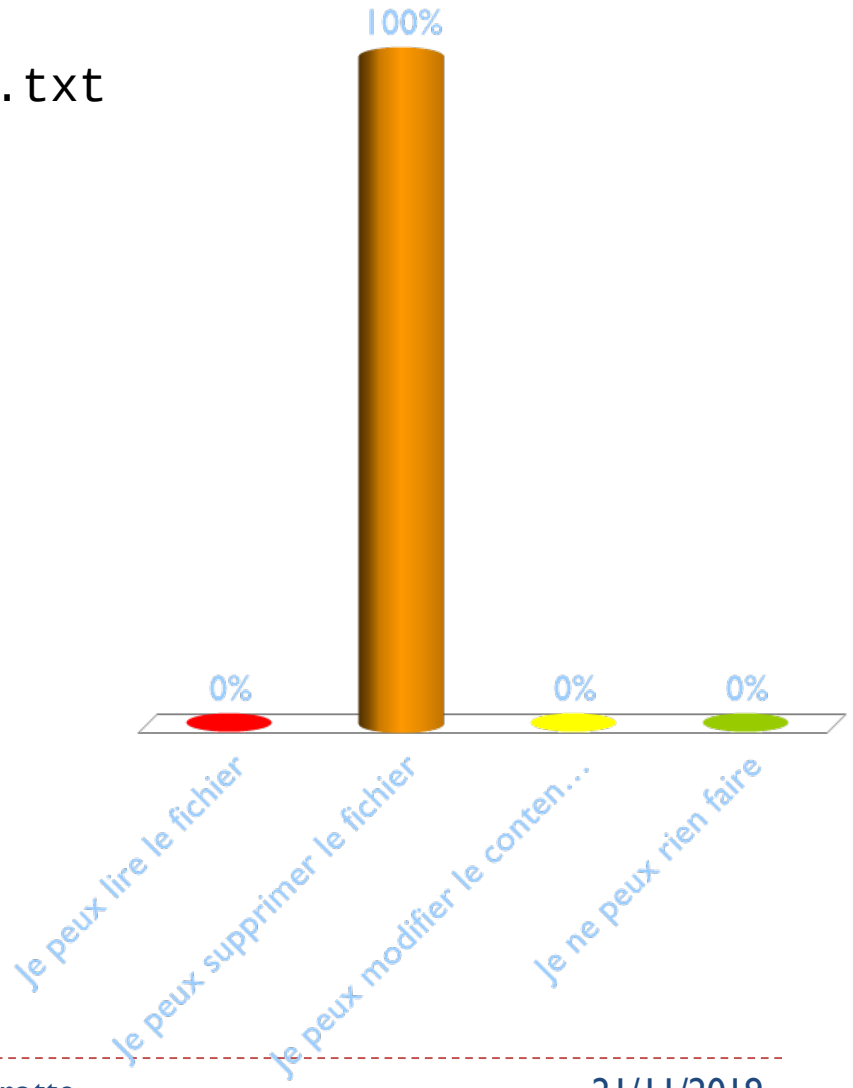
1. Lire et écrire sur le fichier
2. Lire et exécuter le fichier
3. Lire le fichier
4. Rien du tout



Que puis-je faire sur le fichier suivant ?

- `drwxrwxr-x 2 walle pixar .`
- `-rw-r-x--- 1 actor movie cours.txt`
- Je suis l'utilisateur walle qui appartient au groupe pixar

1. Je peux lire le fichier
2. Je peux supprimer le fichier
3. Je peux modifier le contenu du fichier
4. Je ne peux rien faire



Changer à qui appartient un fichier

- Après la création d'un fichier, on peut changer
  - Son propriétaire
  - Son groupe
- Commande: chown
  - `chown user file`
  - `chown user:group file`
  - Attention aux droits pour le faire ! Qui a le droit ?

Université
Nice
Sophia Antipolis

Programmation Shell

Environnement Informatique 1

Faire un Script Shell

- Un script Shell c'est quoi ?
  - Le regroupement de commandes dans un fichier
- Un fichier « script Shell » doit
  - Indiquer le programme qui permettra d'exécuter les commandes
  - Contenir les commandes que l'on souhaite exécuter
  - Être un fichier exécutable
- On n'est pas obligé de le nommer `fichier.sh`
  - Il peut avoir le nom que l'on souhaite (même sans extension)

Entête d'un Script Shell

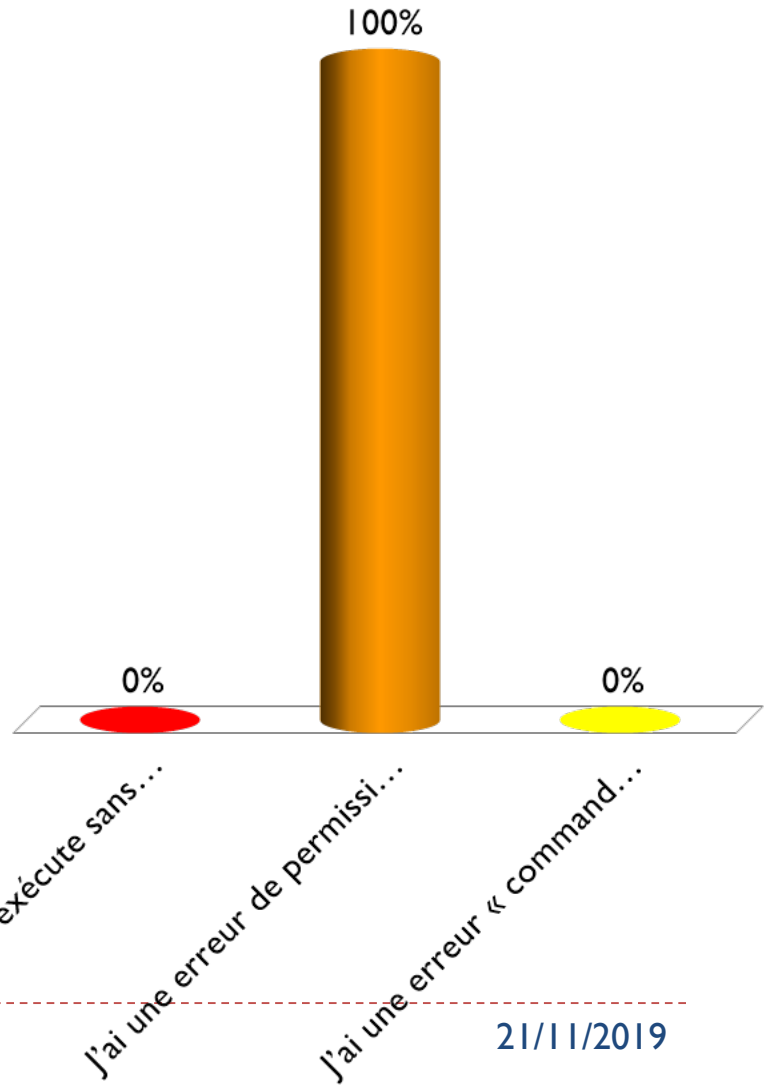
- Qu'est signifie la ligne suivante dans un script Shell:
  - #/bin/bash

1. Le nom du programme exécutant les commandes
2. Le chemin de recherche des commandes
3. Un commentaire
4. L'affectation d'une variable
5. L'affichage du texte `/bin/bash`

Exécution d'un script Shell

- Soit le script Shell créé de la manière suivante:
  - touch mon_script.sh
  - gedit mon_script.sh
  - ./mon_script.sh
- Qu'est ce qui se passe ?

1. Mon script s'exécute sans problème
2. J'ai une erreur de permissions
3. J'ai une erreur « commande introuvable »



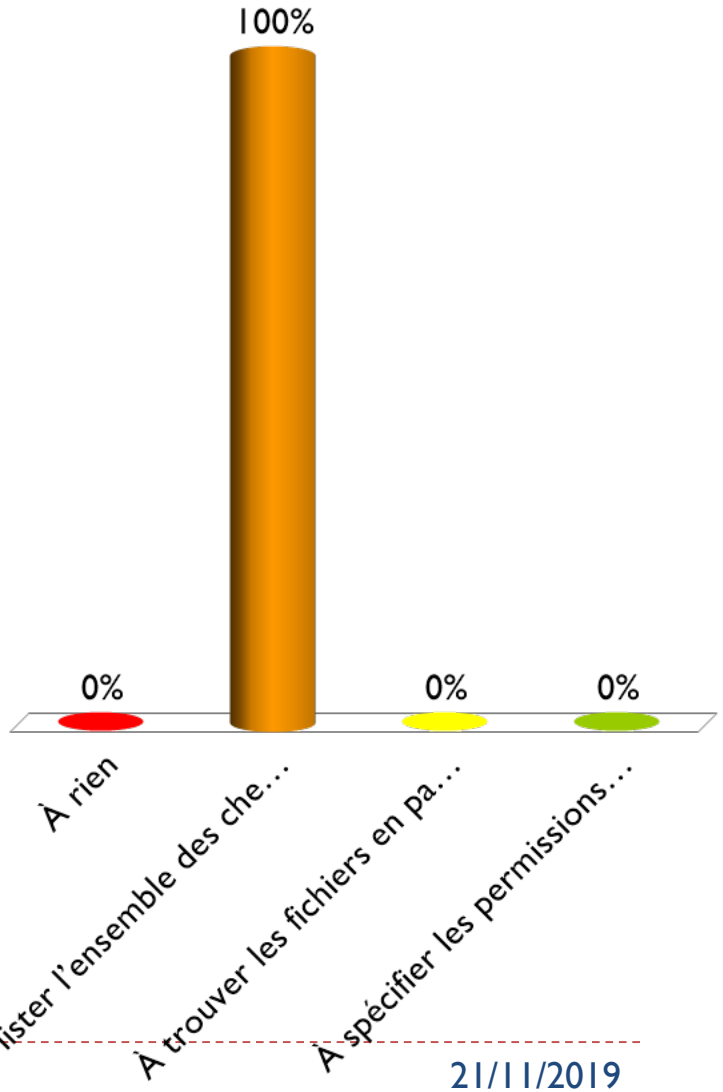
Variables Shell

- Deux types de variables
  - Variables utilisateur
    - Pour vous et dans vos scripts
    - En général en minuscule dans vos scripts
  - Variables d'environnement
    - Des variables « standards » utiles au fonctionnement du système
    - En général définies en majuscule
    - USER, HOME, PATH, …

- Affectation et valeur d'une variable
  - Attention pas d'espaces avant et après le =
  - X=22
  - Y=$X

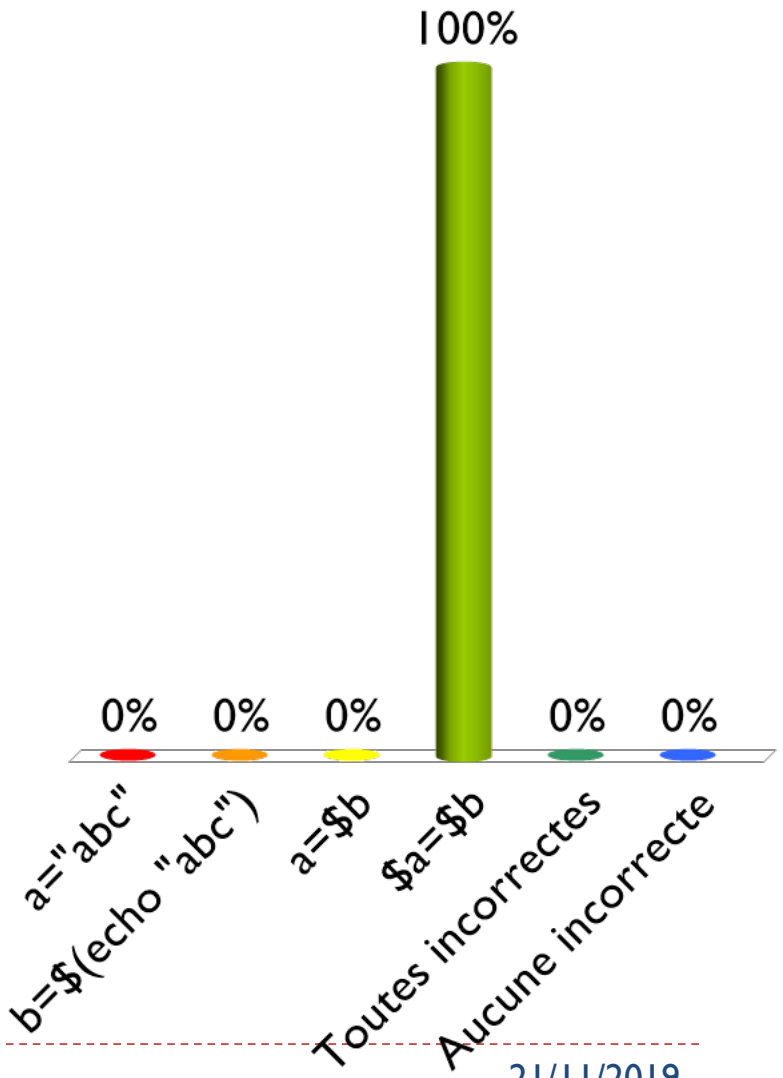
À quoi sert la variable d'environnement PATH ?

1. À rien
2. À lister l'ensemble des chemins où aller chercher les commandes
3. À trouver les fichiers en paramètre d'une commande
4. À spécifier les permissions des programmes



Quelle(s) est (sont) l'(les) initialisation(s) incorrecte(s) d'une variable ?

1. a="abc"
2. b=$(echo "abc")
3. a=$b
4. $a=$b
5. Toutes incorrectes
6. Aucune incorrecte



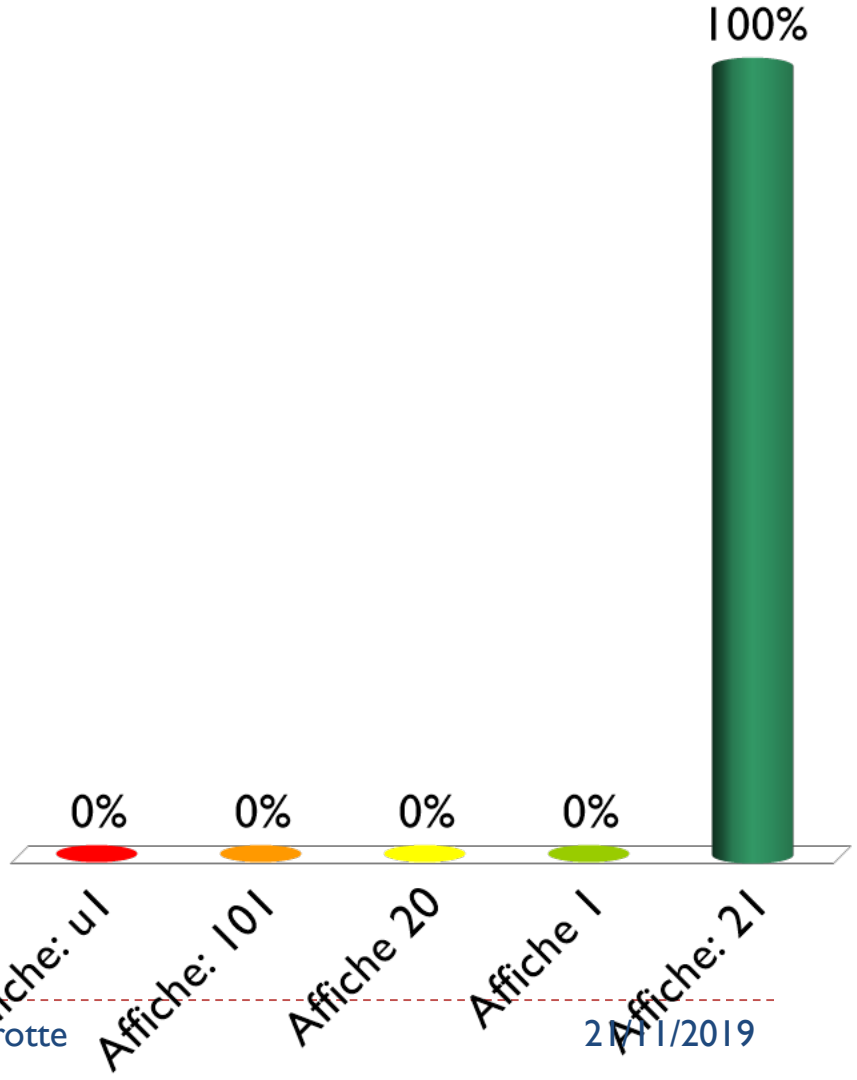
Nommage des variables

- En général les noms de variable de scripts Shell
  - sont en minuscule
    - mais j'ai le droit d'utiliser des majuscules si je ne redéfinis pas un nom de variable d'environnement existant
  - utilisent de préférence des caractères alphabétiques
    - mais je peux tout de même utiliser des chiffres et le caractère _
- On utilise les accolades pour définir sans ambiguïté le nom de variable
  - ${ } à ne pas confondre avec $( )

Nom de variables

- Soit les commandes suivantes:
  - `u1=10`
  - `u=2`
- Quel sera le résultat de:
  - `echo ${u}1`

1. Affiche: u1
2. Affiche: 101
3. Affiche 20
4. Affiche 1
5. Affiche: 21



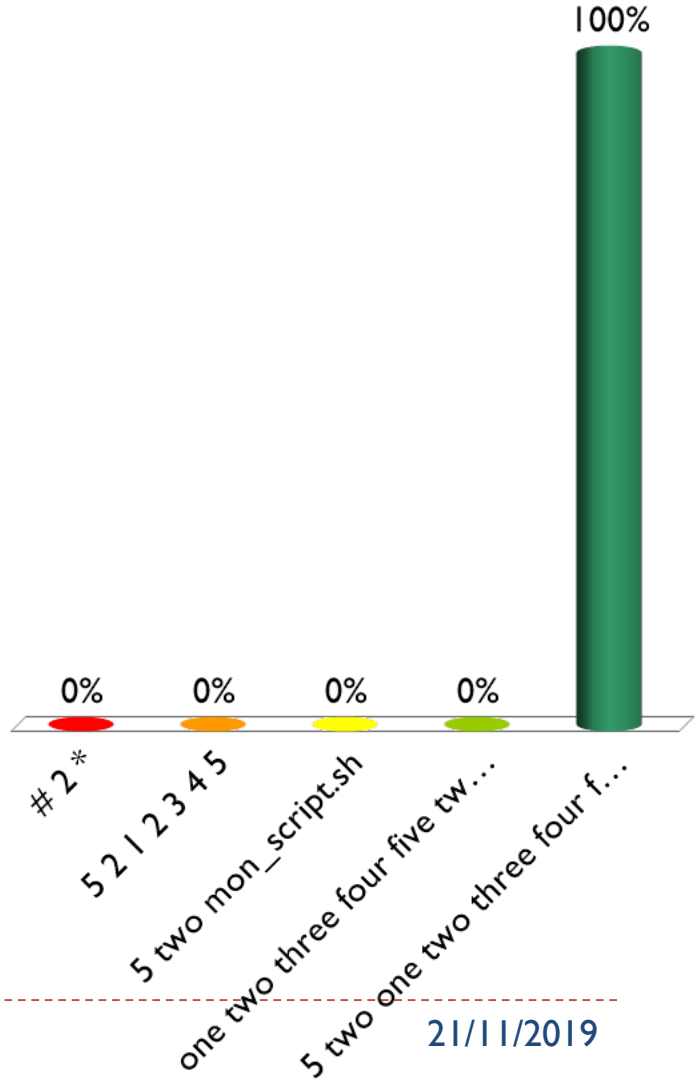
Les variables disponibles pour le script

- Des variables particulières sont créées automatiquement pour votre script Shell
  - $0 $1$2 $3 …
  - $#
  - $*
- Mais aussi
  - $? : affiche le code d'erreur d'exécution de la commande précédente
- Mais à quoi correspondent ces variables ?

Comment récupérer un paramètre passé à mon script ?

- Soit le lancement de `mon_script.sh` de la manière suivante:
  - `mon_script.sh one two three four five`
- Si dans mon script j'ai la commande
  - `echo $# $2 $*`
- qu'est ce qui sera affiché ?

1. # 2 *
2. 5 2 1 2 3 4 5
3. 5 two mon_script.sh
4. one two three four five two 5
5. 5 two one two three four five

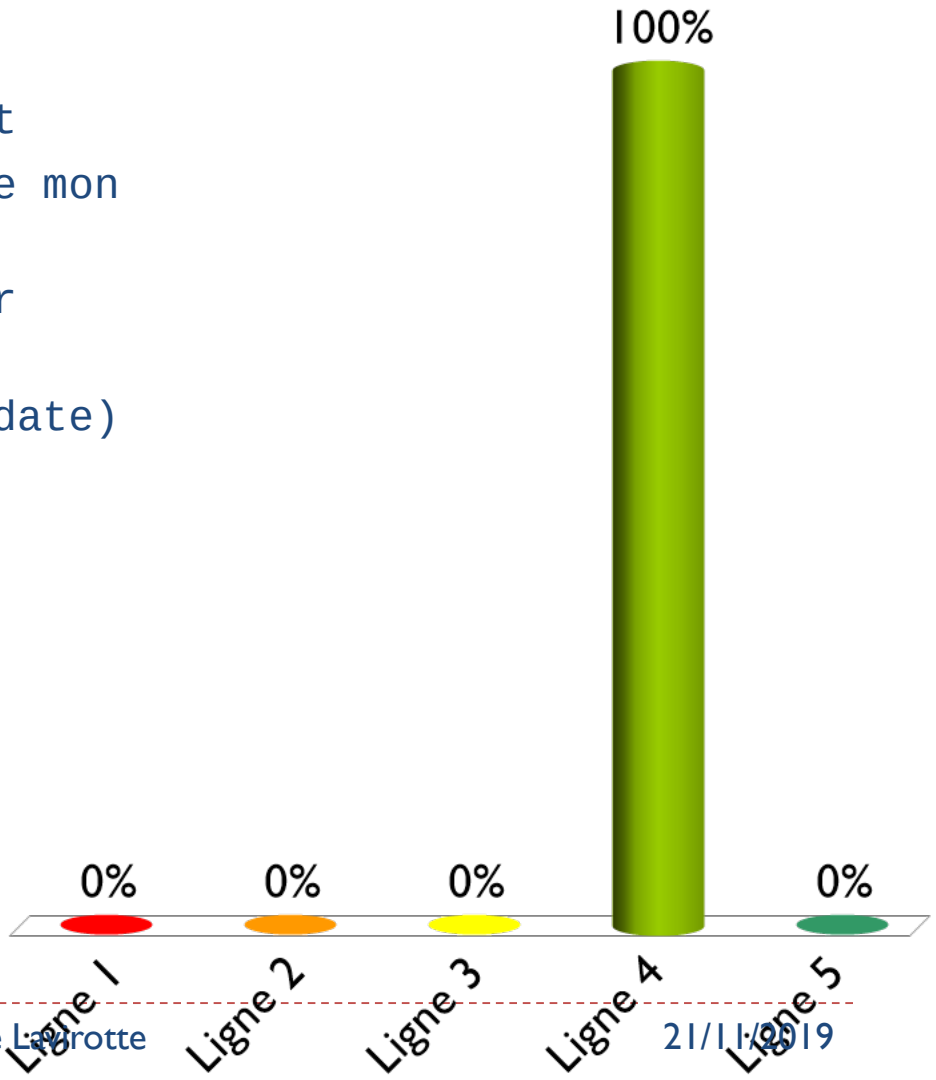


Exemple de script Shell. Où est l'erreur?

- Soit le script:

```
1. #!/bin/bash
2. read –p "Entrez un mot: " mot
3. echo "Le premier paramètre de mon
   script est: $1"
4. echo "Le mot saisi au clavier
   est: " mot
5. echo –n "Nous sommes le " $(date)
```

1. Ligne 1
2. Ligne 2
3. Ligne 3
4. Ligne 4
5. Ligne 5



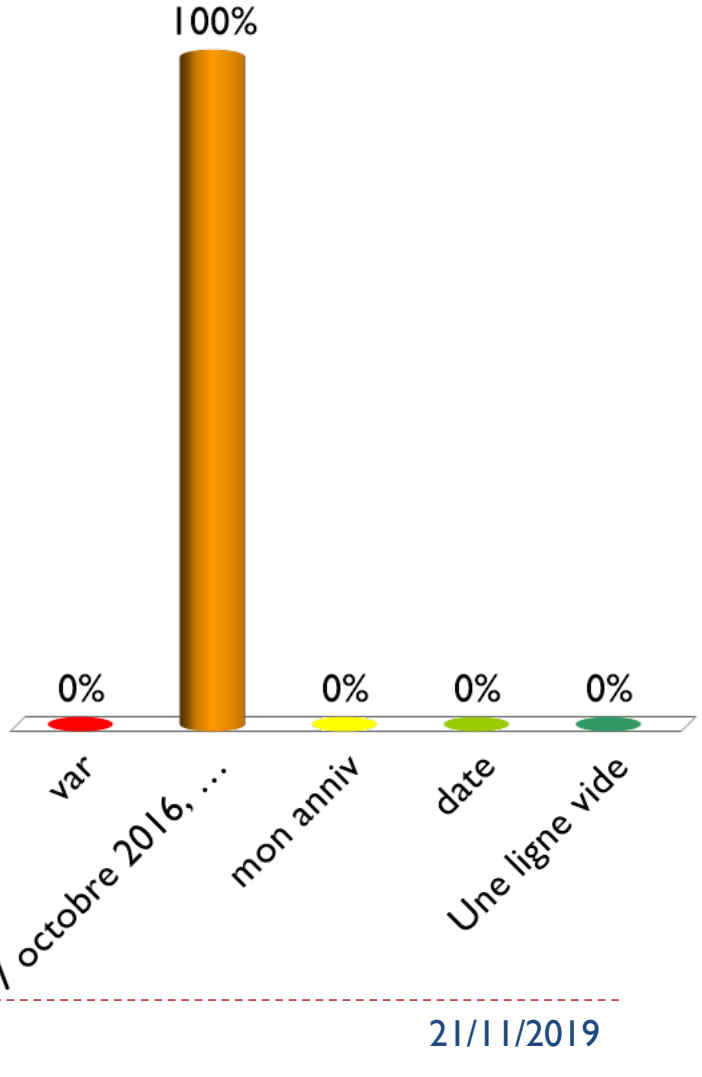
Quel est le résultat obtenu ?

- Soit la suite de commandes suivantes:
  - `date`

```
lundi 17 octobre 2016,16:44:02 (UTC+0200)
```

  - `date="mon anniv"`
  - `var=$(date)`
- Qu'est ce qui est affiché par:
  - `echo ${var}`

1. var
2. lundi 17 octobre 2016, …
3. mon anniv
4. date
5. *Une ligne vide*



Université
Nice
Sophia Antipolis

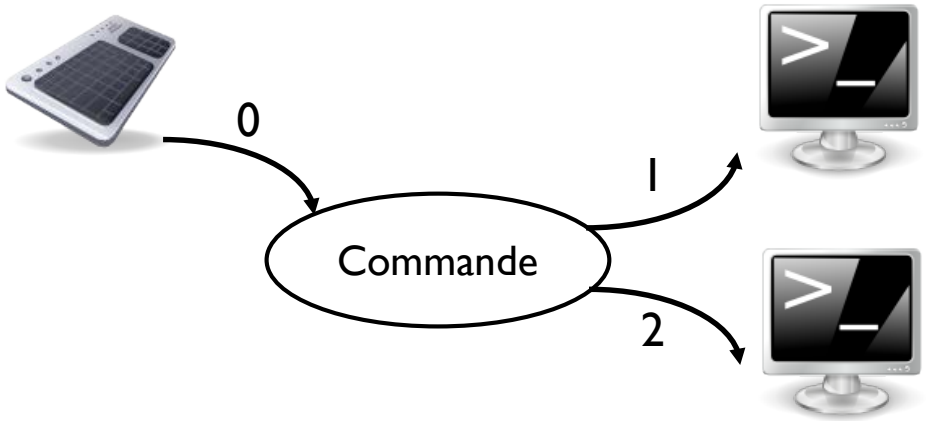
Redirections

Comment gérer les flux de données grâce aux commandes Unix

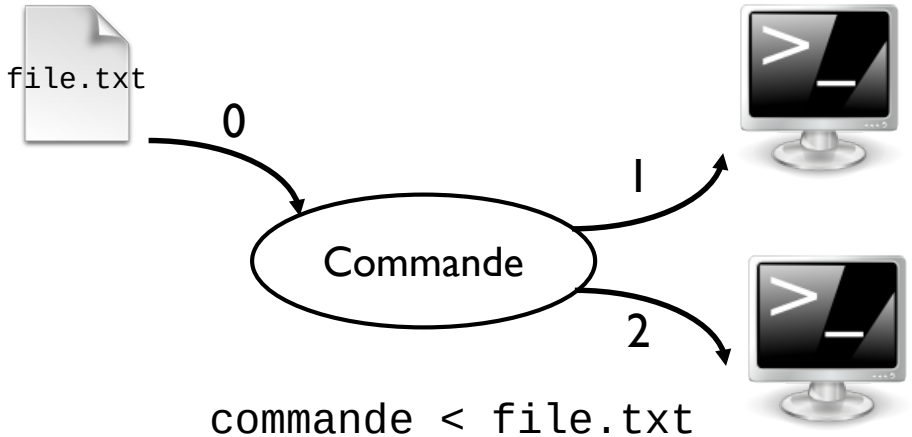
Les Redirections

- Chaque programme a des canaux de communication par défaut:
  - 0: Entrée standard (par défaut clavier)
  - 1: Sortie standard (par défaut le terminal)
  - 2: Sortie standard d'erreur (par défaut le terminal, donc écran)
- Il est possible de rediriger ces canaux de communication
  - Vers ou bien à partir d'un fichier (<, >, >>)
  - Vers un autre canal du même programme (>&)
  - Vers le canal d'un autre programme (|)
- Si on ne spécifie pas de numéro avec la redirection > ou >> c'est le canal 1 qui est celui par défaut

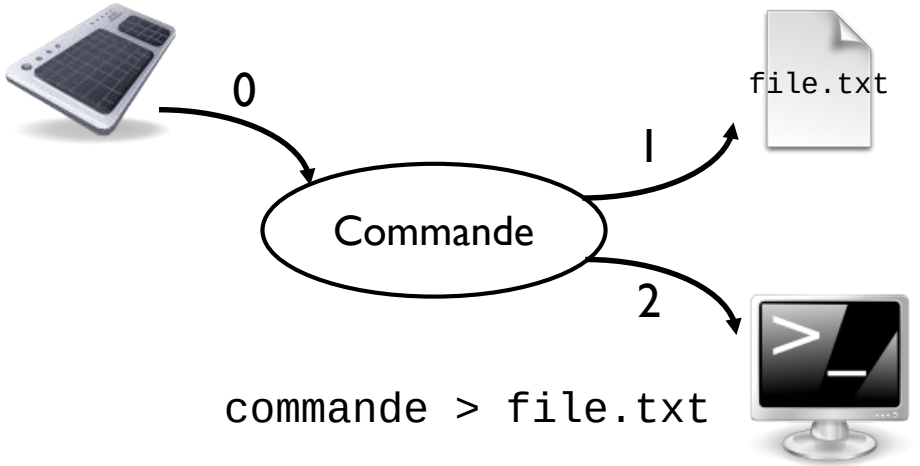
Les Redirections en Images
Vers des fichiers



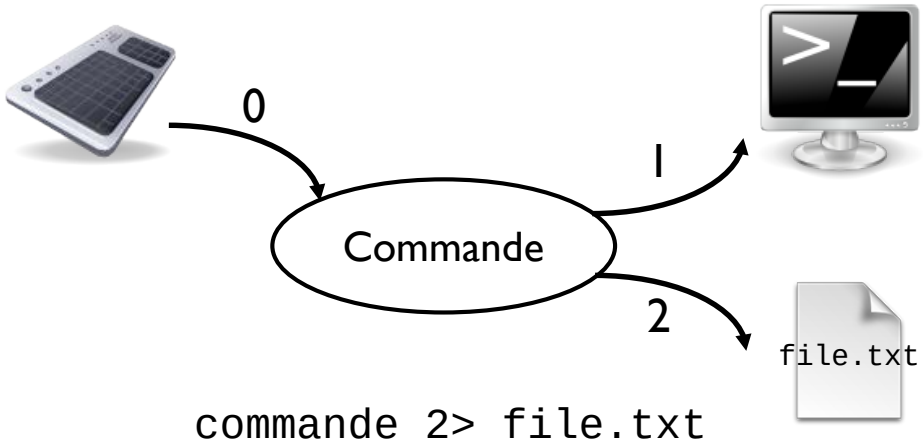
Comportement normal de toute commande



`commande < file.txt`



`commande > file.txt`



`commande 2> file.txt`

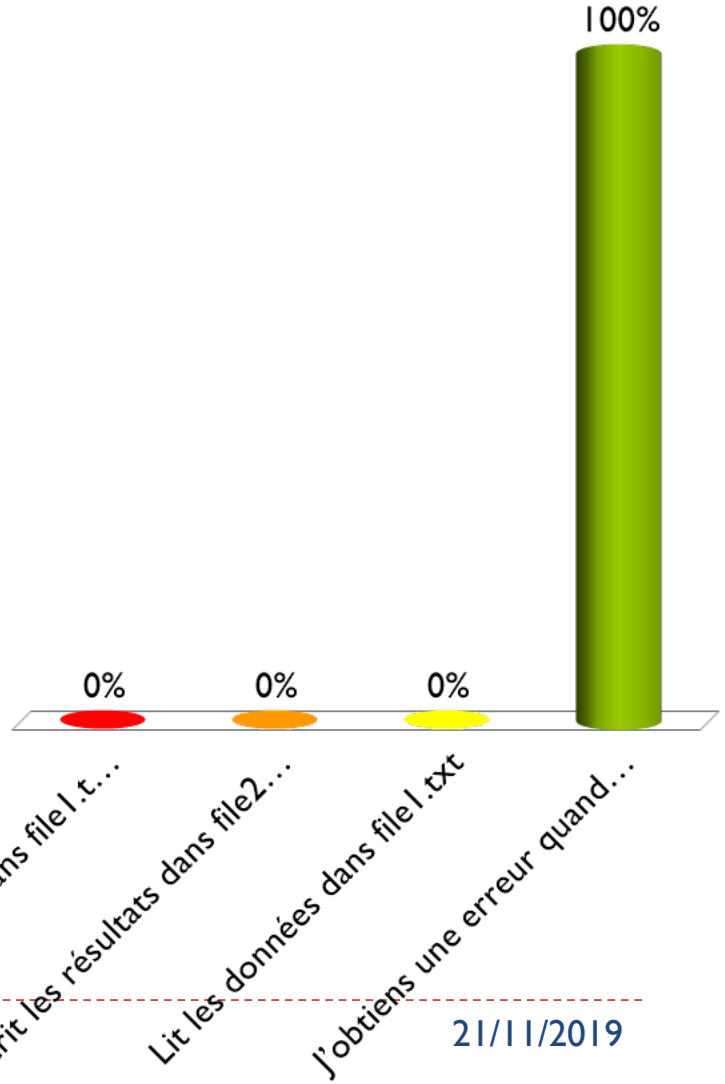
Quel est le résultat de la commande suivante ?

- Soit l'exécution suivante :

```
file1.txt < mon_script.sh > file2.txt
```

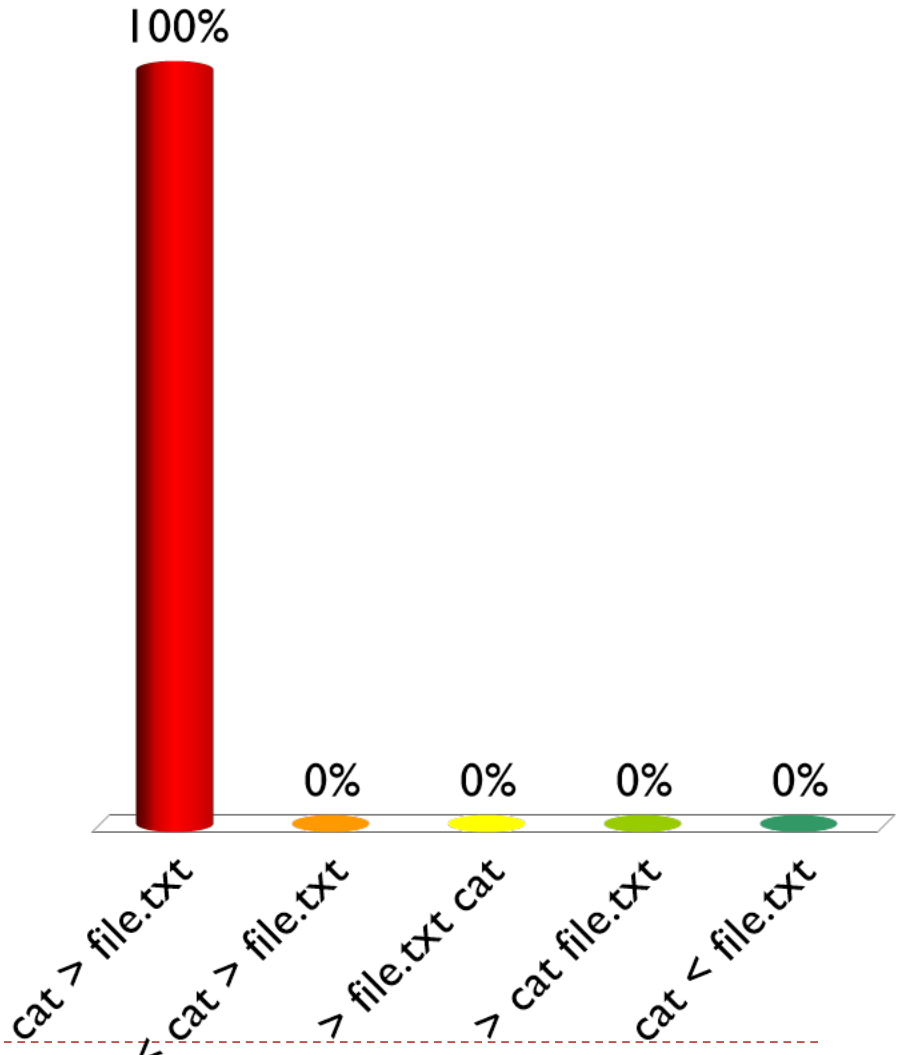
Que fait ce script avec ces deux redirections ?

1. Lit les données dans `file1.txt` et écrit les résultats dans `file2.txt`
2. Ecrit les résultats dans `file2.txt`
3. Lit les données dans `file1.txt`
4. J'obtiens une erreur quand j'exécute cette commande

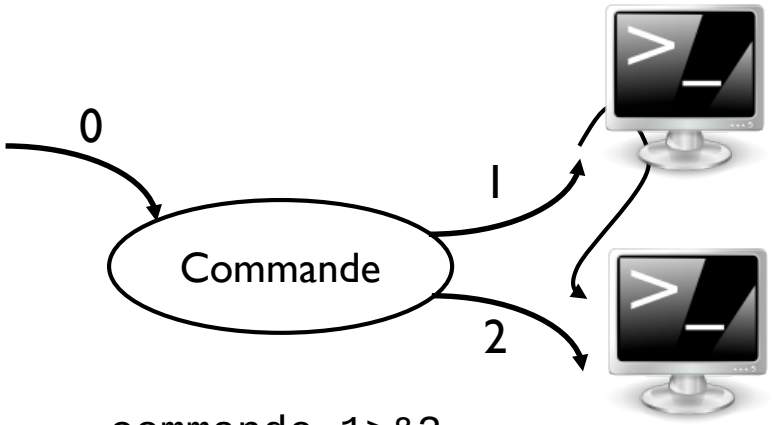


Quelle commande permet de remplir le contenu d'un fichier à partir du clavier ?

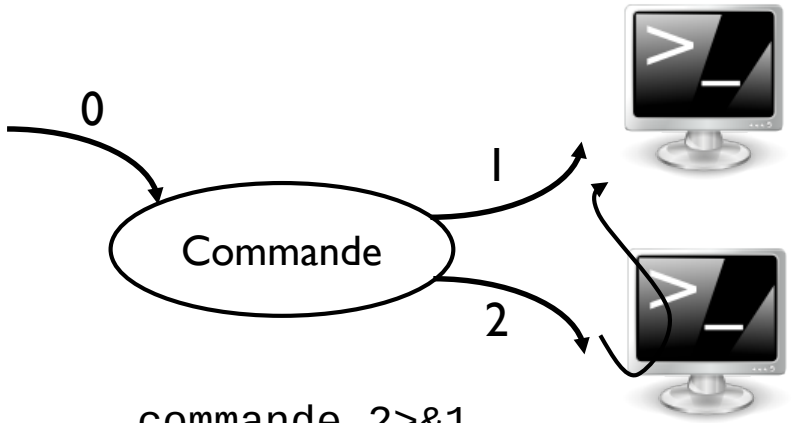
1. cat > file.txt
2. < cat > file.txt
3. > file.txt cat
4. > cat file.txt
5. cat < file.txt



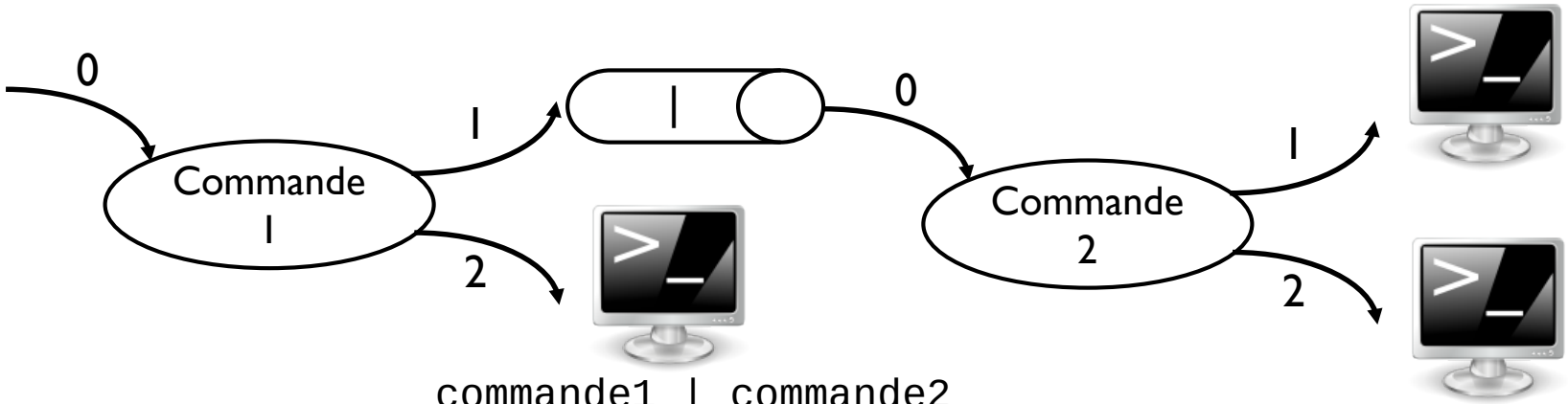
Les Redirections en Images
Vers des canaux



commande 1>&2



commande 2>&1



commande1 | commande2

Que fait la redirection suivante ?

- Soit le script `mon_script.sh`:

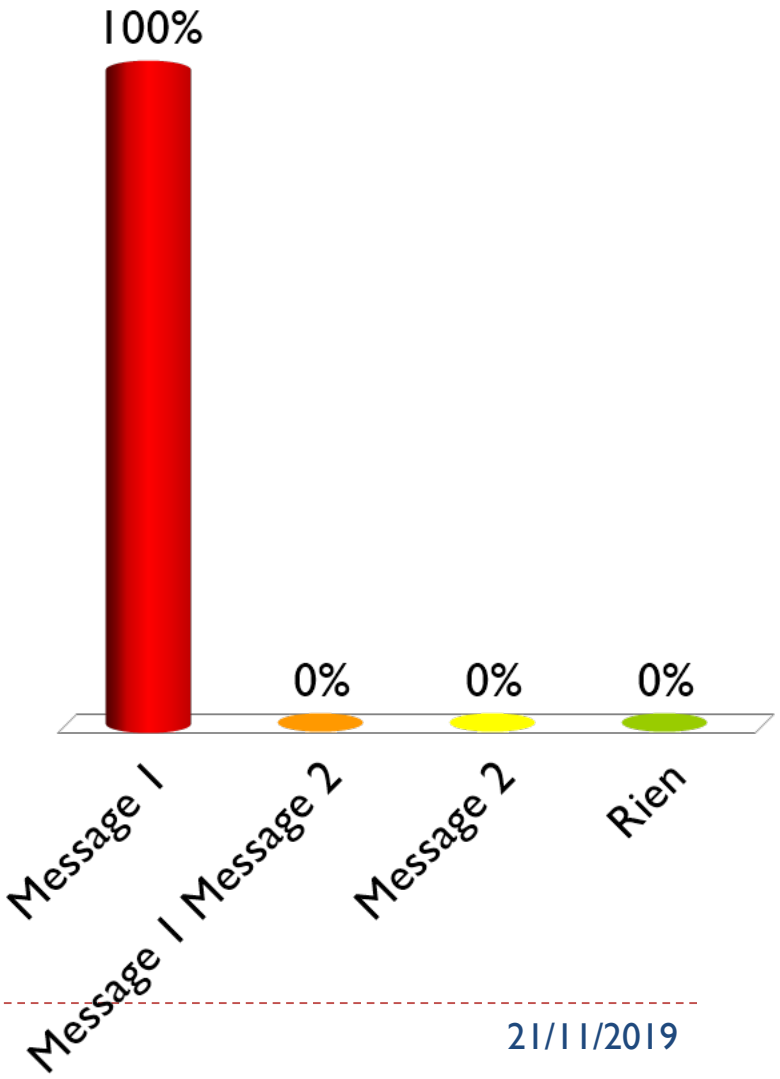
```
#!/bin/bash
echo –n "Message 1 "
echo "Message 2" >&2
```

et l'exécution suivante:

```
mon_script.sh > file.txt
```

Que contient le fichier `file.txt` ?

1. Message 1
2. Message 1 Message 2
3. Message 2
4. Rien



Que fait l'enchainement de commandes suivantes ?

- Soit le fichier `/etc/passwd` suivant:

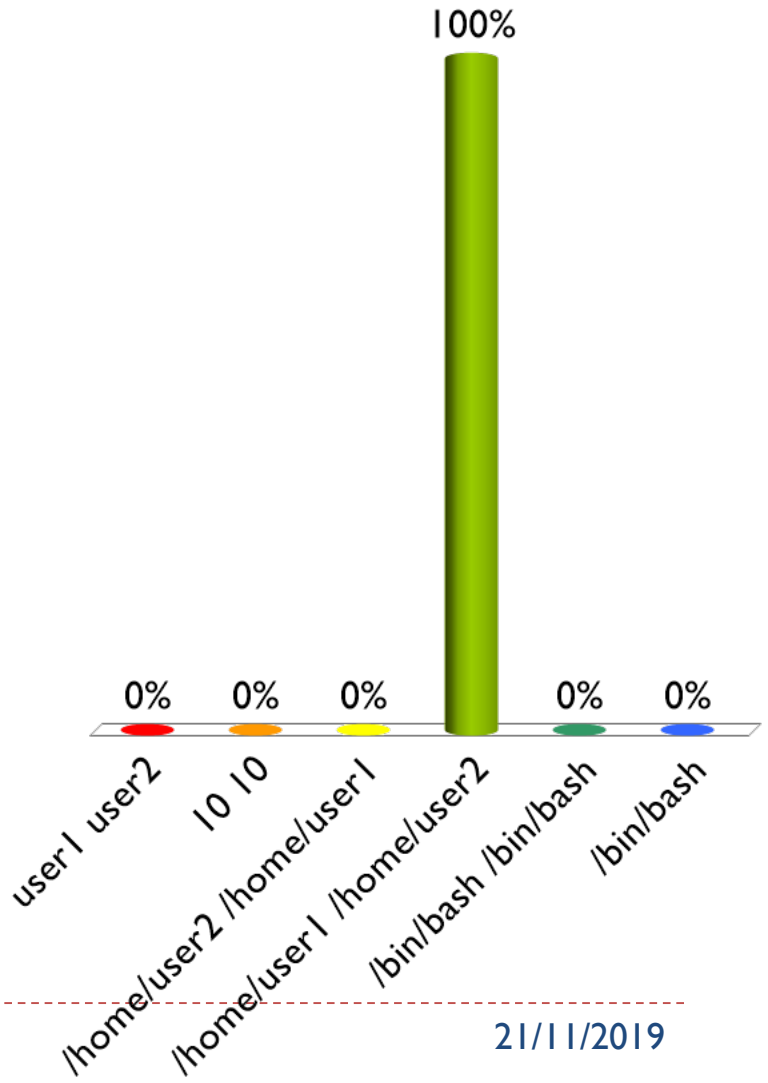
```
user2:x:11:21::/home/user2:/bin/bash
user1:x:10:10::/home/user1:/bin/bash
```

et l'exécution de commandes suivantes:

```
cat /etc/passwd | cut –d ":" –f 6 | sort |
uniq
```

Qu'est ce qui sera affiché ?

1. user1 user2
2. 10 10
3. /home/user2 /home/user1
4. /home/user1 /home/user2
5. /bin/bash /bin/bash
6. /bin/bash



Des suites de commandes pour « tout faire »!

- Des traitements complexes peuvent être réalisés grâce à la redirections de flots de données
  - Compter le nombre de fichiers (ou dossiers) à partir d'un dossier
    - `ls –alR /etc | grep '^-' | wc –l`
  - Créer la liste des utilisateurs (ou des groupes) de votre machine
    - `cat /etc/passwd | cut –d ":"  -f 1 | sort`
  - Connaître le nombre de comptes qui ont bien un mot de passe
    - `sudo cat /etc/shadow | cut –d ":"  -f 2 | grep –v "^[^*!]$"  | wc -l`
  - Compter le nombre de mots dans un document
    - … vous le ferez en TD, je ne vais pas vous donner la solution !

Et pour la suite du cours EnvInfo1

- La partie suivante du cours portera sur deux points
  - Voir un certain nombre d'outils pour le multimédia
    - Chaîne de traitement multimédia
    - Texte, son et images
    - De l'acquisition à la restitution en passant pas l'encodage et l'édition des données
  - Le codage de l'information
    - Comprendre la représentation des information dans la machine
    - Les bits, octets, kibi-octets, mébi-octets gibi-octets, …
    - Nombre en décimal, binaire, octal, hexadécimal, …
- 3 Cours/TD à venir sur le sujet…

Université Nice Sophia Antipolis

Merci et RdV au prochain cours

Pensez à rendre votre TD de cette semaine au plus tard dimanche soir 23h59…