

# La commande grep

## Définition

- La commande **grep** permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes :

`grep options "recherche" chemin`

Avec :

- **options**: les options possibles, que nous verrons un peu plus tard
- **recherche** : le terme à rechercher entre guillemets
- **chemin** : le chemin du fichier (ou dossier) où faire la recherche

- ▶ La commande **grep** permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes :
- ▶ **-v** affiche les lignes ne contenant pas la chaîne
- ▶ **-c** compte le nombre de lignes contenant la chaîne
- ▶ **-n** chaque ligne contenant la chaîne est numérotée
- ▶ **-x** ligne correspondant exactement à la chaîne
- ▶ **-l** affiche le nom des fichiers qui contiennent la chaîne

## Les expressions régulières

Les expressions régulières sont des suites de caractères permettant de faire des sélections. Elles fonctionnent avec certaines commandes comme **grep**. Les différentes expressions régulières sont :

- ^ début de ligne
- .
- \$ 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

## Exercice

- Créer un nouveau fichier **Semaine** dont le contenu est
- Lundi
- Mardi
- Mercredi
- Jeudi
- Vendredi
- Samedi
- Dimanche

- Afficher le contenu du fichier **Semaine**
- Afficher toutes les lignes du fichier **Semaine** contenant **re**
- Afficher toutes les lignes du fichier **Semaine** contenant **di**
- Afficher toutes les lignes du fichier **Semaine** contenant **di** et **DI**

## Exercice 2

- Créer un dossier « TP\_Grep »
- Créer deux fichiers dans le dossier « TP\_Grep »

2- Créer deux fichiers dans le dossier TP\_GREP de noms « stagiaire.txt » et « etudiant.txt »

3- à l'aide d'un éditeur de texte (exemple : vi)

Ecrire le contenu suivant dans les deux fichiers

stagiaire.txt

Je suis un stagiaire  
Je suis marocain  
J'habite au Maroc  
J'ai la nationalité marocaine

etudiant.txt

Je suis un étudiant  
Je suis algérien  
J'habite à l'Algérie  
J'ai la nationalité algérienne



- 4- Rechercher dans le fichier stagiaire.txt les lignes qui contiennent le mot « maroc »
- 5- Rechercher dans le fichier stagiaire.txt les lignes qui contiennent le mot « maroc » sans faire la distinction entre le majuscule et le minuscule
- 6- Rechercher dans le fichier stagiaire.txt les lignes qui ne contiennent pas le mot « maroc »
- 7- Rechercher dans le dossier TP\_GREP les lignes de tous les fichiers qui contiennent le mot « nationalité »

## La commande find

## La commande find

- La commande **find** permet de retrouver des fichiers à partir de certains critères. La syntaxe est la suivante :

`find <répertoire de recherche> <critères de recherche>`

- Les critères de recherche sont les suivants :
- name** recherche sur le nom du fichier,
- 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,
- group** recherche sur le groupe auquel appartient le fichier,
- type** recherche sur le type (d=répertoire, c=caractère, f=fichier normal),
- size** recherche sur la taille du fichier en nombre de blocs (1 bloc=512octets),
- atime** recherche par date de dernier accès en lecture du fichier,
- mtime** recherche par date de dernière modification du fichier,
- ctime** recherche par date de création du fichier.

- ▶ La commande **find** doit être utilisé avec l'option **-print**. Sans l'utilisation de cette option, même en cas de réussite dans la recherche, **find** n'affiche rien à la sortie standard (l'écran, plus précisément le shell).
- ▶ La commande **find** est récursive, c'est à dire où que vous tapiez, il va aller scruter dans les répertoires, et les sous répertoires qu'il contient, et ainsi de suite.

### Exemples de la commande find

- ▶ Pour chercher un fichier dont le nom contient la chaîne de caractères **toto** à partir du répertoire **/usr**, vous devez taper :
- ▶ `find /usr -name toto -print`
- ▶ En cas de réussite, si le(s) fichier(s) existe(nt), vous aurez comme sortie :`toto`
- ▶ En cas d'échec, vous n'avez rien.

## Exemples de la commande find

- Pour rechercher tous les fichiers se terminant par **.c** dans le répertoire **/usr**, vous taperez :
- `find /usr -name "*.c" -print`
- Vous obtenez toute la liste des fichiers se terminant par **.c** sous les répertoires contenus dans **/usr** (et dans **/usr** lui même).

## Exemples de la commande find

- **Recherche suivant la date de dernière modification**
- Pour connaître les derniers fichiers modifiés dans les 3 derniers jours dans toute l'arborescence (**/**), vous devez taper :
- `find / -mtime 3 -print`

## ► Recherche suivant la taille

- Pour connaître dans toute l'arborescence, les fichiers dont la taille dépasse 1Mo (2000 blocs de 512Ko), vous devez taper :`find / -size 2000 -print`

## Exercice (find)

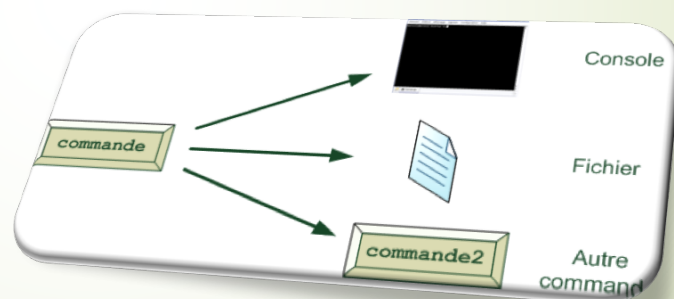
- Cherchez dans toute l'arborescence les fichiers dont le nom se termine par `.c`
- Cherchez dans toute l'arborescence les fichiers dont le nom commence par `x` ou `X`
- Cherchez dans toute l'arborescence les fichiers dont le nom ne contient pas un chiffre
- Chercher dans `/usr` les fichiers dont la taille dépasse 1Mo (2000 blocs de 500Ko)
- Chercher dans `/usr` les fichiers dont les droits sont fixés à 755 (`-rwxr-xr-x`).
- Chercher dans `/usr` les fichiers dont la taille dépasse 1Mo (2000 blocs de 500Ko) et dont les droits sont fixés à 755 (`-rwxr-xr-x`).
- Combien il y a de fichiers dans toute l'arborescence vous appartenant et ayant les droits fixés à 666 (`-rw-rw-rw-`).



- ▶ `find / -name "*.c" -print 2>/dev/null`
- ▶ `find / -name "[Xx]*" -print 2>/dev/null`
- ▶ `find / -name "[!0-9]" -print 2>/dev/null`
  
- ▶ `find /usr (-size +2000 -a -perm 755) -print 2>/dev/null`
- ▶ `find / (-user olivier -a -perm 666 -print) | wc -l`

## Redirection des entrées sorties sous Unix

A. Lasfar (ali.lasfar@gmail.com)



# Plan

- Introduction
- Redirection de l'entrée standard (**stdin**)
- Redirection de la sortie standard (**stdout**)
- Redirection de la sortie d'erreurs standard (**stderr**)
- Redirection d'une sortie standard vers une autre sortie standard

# Introduction

## Plan

- Introduction
- Redirection de l'entrée standard (`stdin`)
- Redirection de la sortie standard (`stdout`)
- Redirection de la sortie d'erreurs standard (`stderr`)
- Redirection d'une sortie standard vers une autre sortie standard

## Introduction

- Chaque processus sous UNIX possède trois canaux de communication :

Canal de communication	Fichier	Numéro logique
Entrée standard	<code>stdin</code>	0
Sortie standard	<code>stdout</code>	1
Sortie d'erreurs standard	<code>stderr</code>	2

## Introduction

- La sortie standard (écran) sur laquelle une commande va envoyer les réponses obtenues.



- Exemple 1 ls

```
Lasfar:TP2 lasfar$ ls
A      F1      F2      FA4      TP2
Lasfar:TP2 lasfar$
```

- Exemple 2 ks

```
Lasfar:TP2 lasfar$ ks
-bash: ks: command not found
Lasfar:TP2 lasfar$
```

## Introduction

- La redirection de ces canaux est très utilisée sous UNIX.
  - Le seul moyen de l'avoir dans un fichier est de rediriger la sortie standard.
- D'autres commandes lisent systématiquement sur leur entrée standard (comme les filtres).
  - Si l'on veut qu'elles prennent un fichier comme argument, il faudra rediriger l'entrée standard. :

## Redirection de l'entrée standard (stdin)

25

A. Lasfar

## Redirection de l'entrée standard (stdin)

### ► Syntaxe :

**commande < Nom\_fichier**

- Il est possible de rediriger l'entrée de toute commande devant lire des données depuis l'entrée standard, afin que la lecture se fasse sur un fichier grâce au symbole « < ».

### ► Exemple :

► mail lasfar < notes\_informatique

A. Lasfar

26

## Redirection de l'entrée standard (stdin)

### Exemple :

```
Lasfar:~ lasfar$ more liste.txt
N°      NOM ET PRENOM  D.NAISS.      CNE
1       AKOUDAD OMAR      21/05/96      1412534773
2       ARIBEH MOHAMED AMINE  20/10/97      1513538973
4       BASSIDI MOHAMED AMINE  26/10/96      1513607019
5       BENAKRACH YASIR    17/11/97      1513540449
6       BENCHAMEKH OTHMAN  16/12/96      1512557847
7       BERQIQCH YASSINE   04/07/96      1412525441
8       BOUFTICHA EL YAAGOUBI OUSSAMA 11/03/97      1513544003
Lasfar:~ lasfar$ mail lasfar <liste.txt
Lasfar:~ lasfar$
```

A. Lasfar

27

## Redirection de l'entrée standard (stdin)

### Exemple

```
Lasfar:~ lasfar$ mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/mail/lasfar": 3 messages 1 new 3 unread
U 1 lasfar@Lasfar.local Mon Feb 15 08:50 14/431 "tt"
U 2 lasfar@Lasfar.local Tue Feb 16 14:18 17/556 "test 2 ARI2"
>N 3 lasfar@Lasfar.locald Fri Feb 19 09:52 20/840
? 3
Message 3:
From lasfar@Lasfar.localdomain Fri Feb 19 09:52:28 2016
X-Original-To: lasfar
Delivered-To: lasfar@Lasfar.localdomain
To: lasfar@Lasfar.localdomain
Date: Fri, 19 Feb 2016 09:52:27 +0000 (WET)
From: lasfar@Lasfar.localdomain (Lasfar)

N°      NOM ET PRENOM  D.NAISS.      CNE
1       AKOUDAD OMAR      21/05/96      1412534773
2       ARIBEH MOHAMED AMINE  20/10/97      1513538973
4       BASSIDI MOHAMED AMINE  26/10/96      1513607019
5       BENAKRACH YASIR    17/11/97      1513540449
6       BENCHAMEKH OTHMAN  16/12/96      1512557847
7       BERQIQCH YASSINE   04/07/96      1412525441
8       BOUFTICHA EL YAAGOUBI OUSSAMA 11/03/97      1513544003
```

A. Lasfar

28

## Redirection de l'entrée standard (stdin)

► << : lire depuis le clavier progressivement

- Le << fait quelque chose d'assez différent : il vous permet d'envoyer un contenu à une commande avec votre clavier.

□ Exemple

```
Lasfar:~ lasfar$
Lasfar:~ lasfar$ wc -w <<STOP
> la redirection des entrées sorties Unix
> STOP
      6
Lasfar:~ lasfar$ wc -c <<STOP
> la redirection des entrées sorties Unix
> STOP
     41
Lasfar:~ lasfar$
```

A. Lasfar

29

## Redirection de l'entrée standard (stdin)

- Exemple : Trier dans le sens croissant une série de nombres saisie au clavier. La fin de la série est indiquée par le mot « Fin » (10; -2; 15; 3; 0; 100; Fin)

```
Lasfar:~ lasfar$ sort -n <<Fin
> 10
> -2
> 15
> 3
> 0
> 100
> Fin
-2
0
3
10
15
100
```

A. Lasfar

30

## Redirection de la sortie standard (stdout)

A. Lasfar

## Redirection de la sortie standard (stdout)

### Syntaxe : (redirection vers un fichier)

commande > nouvelle-sortie (Création/Réécriture)  
commande >> nouvelle-sortie (Ajout)

A. Lasfar

32



## Redirection de la sortie standard (stdout)

### ❑ `commande > nom_fichier`

Si le fichier n'existe pas au moment où la commande est exécutée, il est créé. Si le fichier existait, alors son contenu est écrasé par la sortie standard de la commande.

### ❑ `commande >> nom_fichier`

Si on souhaite que celle-ci vienne s'ajouter à la suite, afin de préserver son contenu initial, il suffit d'utiliser le double symbole « >> ». Dans le cas où le fichier n'existait pas, il sera créé.

A. Lasfor

33

## Redirection de la sortie standard (stdout)

### Exemples :

#### ❑ `ls -l > lister_dossier` :

le résultat de la commande `ls -l` ne sera pas affiché à l'écran, il sera enregistré dans le nouveau fichier `lister_dossier`

#### ❑ `date > date_aujourd'hui` :

la date du système sera enregistrée dans le fichier `date_aujourd'hui`

#### ❑ `who > utilisateurs_connectés` :

la liste des utilisateurs connectés au système sera enregistrée dans le fichier `utilisateurs_connectés`

A. Lasfor

34

## Redirection de la sortie standard (stdout)

Exemples :

❑ `ls -l > lister_dossier :`

```
lasfar:~ lasfar$
Lasfar:~ lasfar$ ls -l > lister_dossier
Lasfar:~ lasfar$ more lister_dossier
total 1200
-rw-r--r--  1 lasfar  staff   435 15 fév 08:46 1
-rw-r--r--  1 lasfar  staff   435 15 fév 08:45 2
drwxr-xr-x 17 lasfar  staff   578 27 mai  2015 2015
drwx-----  5 lasfar  staff   170  3 jan 20:00 Applications
drwxr-xr-x 22 lasfar  staff   748  4 mai  2015 BD LP LD
drwxr-xr-x+ 33 lasfar  staff  1122 18 fév 15:44 Desktop
drwxr-xr-x+ 37 lasfar  staff  1258 17 fév 08:16 Documents
-rwxr-xr-x+ 591 lasfar  staff 20094 19 fév 08:26 Downloads
```

❑ `date > date_aujourdhui :`

```
Lasfar:~ lasfar$ date > date_aujourdhui
Lasfar:~ lasfar$ more date_aujourdhui
Ven 19 fév 2016 09:22:14 WET
Lasfar:~ lasfar$
```

A. Lasfar

35

## Redirection de la sortie d'erreurs standard (stderr)

36

A. Lasfar

## Redirection de la sortie d'erreurs standard (stderr)

### Syntaxe :

**commande 2> nouvelle-sortie** (Création/Réécriture)

**commande 2>> nouvelle-sortie** (Ajout)

Pour rediriger la sortie d'erreurs standard dans un fichier, on utilise les chaînes « 2> » et « 2>> » suivie du nom du fichier.

## Redirection de la sortie d'erreurs standard (stderr)

### ❑ commande 2> nom\_fichier

- ❑ La plupart des commandes UNIX produisent des messages de diagnostic si un problème survient en cours d'exécution. La sortie des messages d'erreur se fait sur la sortie d'erreurs standard, qui, par défaut, est associée à l'écran.

### ❑ commande 2>> nom\_fichier

- ❑ Si on souhaite que celle-ci vienne s'ajouter à la suite, afin de préserver son contenu initial, il suffit d'utiliser le double symbole « >> ». Dans le cas où le fichier n'existait pas, il sera créé.

## Redirection de la sortie d'erreurs standard (stderr)

### Syntaxe :

commande 2> nouvelle-sortie (Création/Réécriture)

commande 2>> nouvelle-sortie (Ajout)

- Il ne doit **pas y avoir d'espace entre le « 2 » et le « > »**.
- Comme pour la redirection de la sortie standard, si le fichier n'existe pas, il est créé, sinon il est écrasé.
- Si l'on veut que les messages de diagnostics viennent s'ajouter en fin de fichier, il faut utiliser le double symbole de redirection « 2>> ».

A. Lasfar

39

## Redirection de la sortie d'erreurs standard (stderr)

### Syntaxe :

commande 2> nouvelle-sortie (Création/Réécriture)

commande 2>> nouvelle-sortie (Ajout)

- Exemple

```
Lasfar:~ lasfar$
Lasfar:~ lasfar$ llc 2>fichier_Err
Lasfar:~ lasfar$ more fichier_Err
-bash: llc: command not found
Lasfar:~ lasfar$
```

A. Lasfar

40

## Redirection des sorties standard et d'erreur

- Redirige la sortie d'erreur et la sortie standard vers fichier

**>fichier 2>&1**

```
465  KS  
466  ls -l >fichier 2>&1  
467  more fichier  
468  clear  
469  lls -l >fichier 2>&1  
470  more fichier  
471  history
```

- Redirige la sortie d'erreur et la sortie standard à la fin du fichier

>>fichier 2>&1

```
Lasfar:~ lasfar$ lls -l >>fichier 2>&1
Lasfar:~ lasfar$ more fichier
-bash: lls: command not found
-bash: lls: command not found
Lasfar:~ lasfar$ ls -l >>fichier 2>&1
Lasfar:~ lasfar$ more fichier
-bash: lls: command not found
-bash: lls: command not found
total 1232
-rw-r--r--    1 lasfar  staff    435 15 fév 08:46 1
-rw-r--r--    1 lasfar  staff    435 15 fév 08:45 2
drwxr-xr-x   17 lasfar  staff    578 27 mai  2015 2015
drwx-----    5 lasfar  staff    170  3 jan 20:00 Applications
drwxr-xr-x   22 lasfar  staff    748  4 mai  2015 BD LP LD
drwxr-xr-x+  44 lasfar  staff   1496 20 fév 19:16 Desktop
drwxr-xr-x+  37 lasfar  staff   1258 17 fév 08:16 Documents
```

A. Lasfar

43

## Résumé

- ❑ **commande > fichier** redirige la sortie standard de "**commande**" dans le fichier "**fichier**"
- ❑ **commande 2> fichier** redirige la sortie des erreurs de "**commande**" dans le fichier "**fichier**"
- ❑ **commande 2>&1** redirige la sortie des erreurs de "**commande**" vers la sortie standard,

A. Lasfar

44

## Résumé

- ❑ `commande < fichier` redirige l'entrée standard de "**commande**" dans le fichier "**fichier**",
- ❑ `commande1 | commande2` redirige la sortie standard de la commande "**commande1**" vers l'entrée standard de "**commande2**".
- ❑ `commande1 | & commande2` branche la sortie d'erreur de **commande1** sur l'entrée de **commande2**

A. Lasfar

45

46

## Les métacaractères du Shell

A. Lasfar

- 47
- ❑ \* désigne une chaîne de caractères quelconque ;
  - ❑ ? désigne un caractère quelconque ;
  - ❑ [...] désigne les caractères entre crochets, définis par énumération ou par un intervalle.
  - ❑ ^ désigne un début de ligne ;
  - ❑ \$ désigne une fin de ligne ;
  - ❑ {nombre} désigne la répétition du caractère précédent, qu'il soit caractère normal ou métacaractère

48

Exemples :

- ❑ [Aa] désigne les caractères **A** ou **a** ;
- ❑ [0-9a-zA-Z] désigne un caractère alphanumérique quelconque.

Remarque :

- ❑ [!0-9] désigne l'ensemble des caractères sauf les chiffres.

A. Lasfar