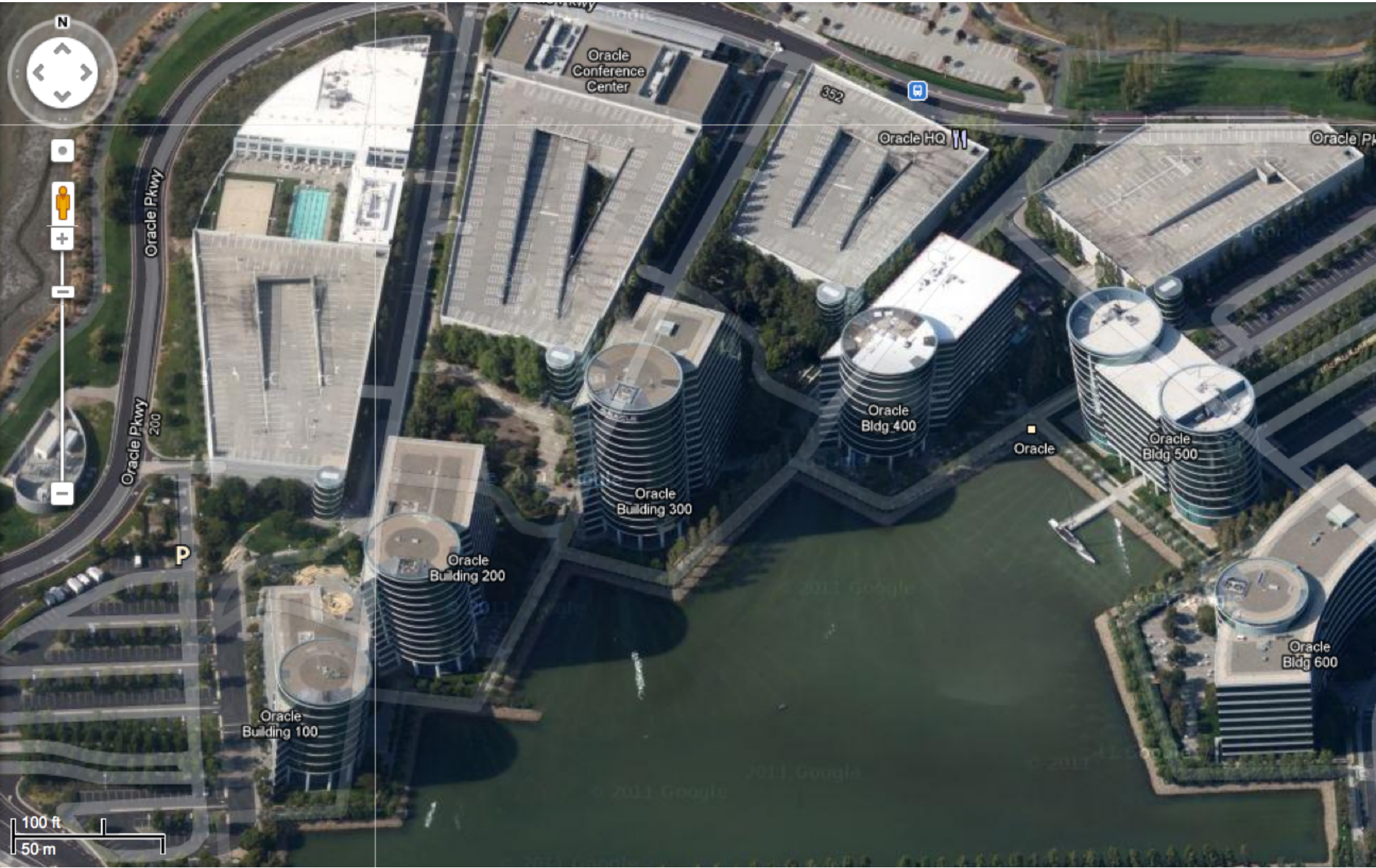


100 ft
50 m

© 2011 Google



Oracle
Conference
Center

352

Oracle HQ

Oracle Pkwy

Oracle Pkwy
200

P

Oracle

Oracle
Bldg 500

Oracle
Building 300

Oracle
Building 200

Oracle
Building 100

Oracle
Bldg 600

100 ft
50 m

Gestion de Masse de Données (GMD)

Cours 1

Adrien Coulet
adrien.coulet@loria.fr

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

a) bash

• Rappels

- un shell, c'ad un langage de script linux
- différents modes d'exécution

- dans une ligne de commande

Saisie

commande

commande &

commande > fichier

commande >> fichier

commande < fichier

commande1 | commande2

- dans un fichier de script

```
#!/bin/bash
```

```
var="Hello world!"  
echo $var
```

OU

```
#!/bin/sh
```

```
var="Hello world!";  
echo $var;
```

```
potrero:~ coulet$ ./test.sh  
Hello world!  
potrero:~ coulet$
```

Rappels des bases

Saisie	Signification
commande	Exécution normale de la commande ; le symbole d'invité de commande s'affiche lorsque la commande est terminée.
commande &	exécution en arrière-plan, càd que le shell reprend la main et affiche sons symbole d'invite de commande alors que la commande s'exécute en tâche de fond. [1] 2496 puis [1]+
commande > fichier	La sortie standard est dirigée vers le fichier indiqué. Seule la sortie d'erreur s'affiche à l'écran.
commande >> fichier	La sortie standard est ajoutée à la fin du fichier sans en écraser le contenu.
commande < fichier	Les informations d'entré de la commande sont lues depuis le fichier plutôt que depuis le terminal.
commande1 commande2	Les deux commandes sont exécutées en une seule fois, l'entrée standard de la seconde étant connectée par un <i>pipe</i> à la sortie standard de la première.

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

a) bash

- Rappels (suite)
 - par défaut les variables ne sont pas typées
 - ce sont par défaut des chaînes de caractères
 - affectation = création
 - appel d'une variable avec `$ma_variable`

```
potrero:~ coulet$ x="Hello"  
potrero:~ coulet$ echo $x  
Hello  
potrero:~ coulet$
```

- Naviguer dans le système de fichier
 - Base de base :
 - dossier courant `pwd`;
 - changer de dossier `cd`;
 - Lister le contenu d'un dossier `ls`.

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

a) bash

• Naviguer dans le système de fichier (suite)

– Commande **find**

```
potrero:dossier_test coulet$ find . -type d  
.  
./dir_a  
./dir_a/dir_aa  
./dir_b
```

```
potrero:dossier_test coulet$ find . -type f -name "*.java"  
./dir_a/dir_aa/fichier_aa1.java  
./dir_a/fichier_a2.java  
./dir_b/fichier_b1.java  
./fichier4.java  
./fichier6.java
```

```
potrero:dossier_test coulet$ ls -al  
total 0  
drwxr-xr-x 10 coulet  staff  340 Feb 17 10:46 .  
drwxrwxrwx  5 coulet  staff  170 Feb 17 10:35 ..  
drwxr-xr-x  5 coulet  staff  170 Feb 17 10:47 dir_a  
drwxr-xr-x  4 coulet  staff  136 Feb 17 10:46 dir_b  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:39 fichier1  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:39 fichier2  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:39 fichier3.txt  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:40 fichier4.java  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:40 fichier5.txt  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:40 fichier6.java  
potrero:dossier_test coulet$ find .  
.  
./dir_a  
./dir_a/dir_aa  
./dir_a/dir_aa/fichier_aa1.java  
./dir_a/fichier_a1  
./dir_a/fichier_a2.java  
./dir_b  
./dir_b/fichier_b1.java  
./dir_b/fichier_b2  
./fichier1  
./fichier2  
./fichier3.txt  
./fichier4.java  
./fichier5.txt  
./fichier6.java  
potrero:dossier_test coulet$
```

```
potrero:dossier_test coulet$ find . -type f ! -name "*.java" -exec chmod 777 {} \;
```

```
drwxr-xr-x 10 coulet  staff  340 Feb 17 10:46 .  
drwxrwxrwx  5 coulet  staff  170 Feb 17 10:35 ..  
drwxr-xr-x  5 coulet  staff  170 Feb 17 10:47 dir_a  
drwxr-xr-x  4 coulet  staff  136 Feb 17 10:46 dir_b  
-rwxrwxrwx  1 coulet  staff   0 Feb 17 10:39 fichier1  
-rwxrwxrwx  1 coulet  staff   0 Feb 17 10:39 fichier2  
-rwxrwxrwx  1 coulet  staff   0 Feb 17 10:39 fichier3.txt  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:40 fichier4.java  
-rwxrwxrwx  1 coulet  staff   0 Feb 17 10:40 fichier5.txt  
-rw-r--r--  1 coulet  staff   0 Feb 17 10:40 fichier6.java
```

- Plus d'options : !, -exec

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

- Naviguer dans le système de fichier (suite)

a) bash

–Commande **locate**

- Exemple

```
potrero:GMD coulet$ locate "/Users/coulet/research/*.pdf"
/Users/coulet/research/20101101_jbi/coulet-nlp-jbi.pdf
/Users/coulet/research/20101118_jbs/couletetal_jbs.pdf
/Users/coulet/research/20101118_jbs/first_submission/couletetal_jbs.pdf
/Users/coulet/research/20101118_jbs/first_submission/fig/pd.png.pdf
/Users/coulet/research/20101118_jbs/second_submission/couletetal_submission/couletetal_jbs_manuscript.pdf
/Users/coulet/research/20101118_jbs/second_submission/couletetal_submission/couletetal_jbs_response_to_reviews.pdf
/Users/coulet/research/20101129_biordf/101129AdrienCouletBioRDF.pdf
/Users/coulet/research/20101214_max_plank/101214_adrien.pdf
/Users/coulet/research/20110118_springer/proof/209587_1_En_36.pdf
/Users/coulet/research/20110118_springer/proof/209587_1_En_36_DeltaPDF.pdf
/Users/coulet/research/20110118_springer/proof/209587_1_En_36_with_ac_comments.pdf
/Users/coulet/research/ANR/anr-hybride-2011.pdf
/Users/coulet/research/ANR/cv_adrien.pdf
/Users/coulet/research/CPER/2011-13/presentation_adrien.pdf
/Users/coulet/research/cluster_mbi/clustermbi.pdf
/Users/coulet/research/israel/biblio/BNA-4.pdf
/Users/coulet/research/rapport_activités/2009/hal_orpailleur_2010.pdf
/Users/coulet/research/rapport_activités/2009/orpailleur_ra2009.pdf
/Users/coulet/research/these_ac/these_adrien.pdf
```

- Recherche uniquement sur les noms des chemins
- Pas sur les attributs des fichiers (cf. find)
- Utilise une base de données mais par un script cron
- locate -S

```
potrero:GMD coulet$ locate -S
Database: /var/db/locate.database
Compression: Front: 13.87%, Bigram: 60.16%, Total: 9.92%
Filenames: 613499, Characters: 62088628, Database size: 6159949
Bigram characters: 2454277, Integers: 98098, 8-Bit characters: 1128
```

- Si recherche sur des fichiers récents: updatedb

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

- Accéder au contenu d'un fichier a) bash
 - Commande cat
 - Lecture de fichiers
 - cat fichier1
 - cat fichier1 > fichier2
 - cat fichier1 >> fichier2
 - Concaténation (utile pour la transformation)
 - cat fichier1 fichier2 fichier3 | sort > fichier4

- Extraction de sous-chaînes
 - option ':' de \${ }
 - Longueur de chaîne \${# }

```
potrero:~ coulet$ variable=helloworldhowareyoutoday
potrero:~ coulet$ echo ${variable:5:5}
world
potrero:~ coulet$ echo ${variable:10}
howareyoutoday
potrero:~ coulet$
```

```
coulet@potrero:~$ var1=helloworldhowareyoutoday
coulet@potrero:~$ var2=helloworld
coulet@potrero:~$ echo ${var1:${#var2}+1}
howareyoutoday
```


I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) grep

- **grep** : trouve les lignes d'un fichier qui contiennent une chaîne de caractère part

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ cat ./sie/liste_2a_sie.csv
1,Axel,AUPIC,,G3,GG2,SIE
2,Dhouha,AYED,,G5,GG3,SIE
3,Margaux,BERTHOU,,G2,GG3,SIE
4,Elise,BOINNOT,,G3,GG4,SIE
5,Timothee,CORDIER,,G5,GG1,SIE
6,Nicolas,EISEN,,G1,GG1,SIE
7,Sanaa,ELHAOUZI,,G3,GG4,SIE
8,Damien,FERRANT,,G1,GG1,SIE
9,Loic,GUILLEMIN,,G5,GG2,SIE
10,Marouane,HAMOUMI,,titre,GG3,SIE
11,Romain,HORY,,G2,GG4,SIE
12,Julio Raphael,KAMGAING,,G3,GG1,SIE
13,Mathieu,KLEIN,,G3,GG2,SIE
14,Youri,KOMUSSIDI,,G5,GG3,SIE
15,Anthony,LEPRETRE,,G1,GG4,SIE
16,Mounir,MEJAIT,,G1,GG1,SIE
17,Clement,RIVELLINI,,ESSTIN,GG4,SIE
18,Ghizlane,SALHI,,G1,GG2,SIE
19,Damien,SCHIRM,,G2,GG3,SIE..
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep Damien ./sie/liste_2a_sie.csv
8,Damien,FERRANT,,G1,GG1,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

• Premières **expressions régulières (regexp)**

- Une expression régulière est une chaîne de caractères qui décrit un patron de chaîne de caractères
- Si une ligne lue par grep contient un chaîne qui corresponde au patron spécifié, alors grep retourne cette ligne

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep "Damien" ./sie/liste_2a_sie.csv
8,Damien,FERRANT,,G1,GG1,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep "Damien.*GG1" ./sie/liste_2a_sie.csv
8,Damien,FERRANT,,G1,GG1,SIE
```

Symboles	Interprétation
^	Expression en début de ligne
\$	Expression en fin de ligne
\	Protèges les caractères spéciaux de leur interprétation dans une regexp ex : \^
[]	Correspond à n'importe quel caractère décrit entre les crochets ex : [aeiou], [A-Za-z][0-9]
[^]	Correspond à n'importe quel caractère non décrit entre les crochets ex : [^aeiou] [^0-9]
.	Correspond à n'importe quel caractère sauf fin de ligne
\< et \>	Début ou fin de mot
*	Cardinalité : l'expression ou caractère précédent doit être trouvée 0...n fois ex : [0-9]*
\{x\}	Cardinalité : le précédent doit être trouvé exactement x fois ex : m\{2\}

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) grep

- Exemple d'utilisation d'**expressions régulières (1/2)**

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep "SIE" ./sie/liste_2a_sie.csv
```

```
1,Axel,AUPIC,,G3,GG2,SIE
2,Dhouha,AYED,,G5,GG3,SIE
3,Margaux,BERTHOU,,G2,GG3,SIE
4,Elise,BOINNOT,,G3,GG4,SIE
5,Timothee,CORDIER,,G5,GG1,SIE
6,Nicolas,EISEN,,G1,GG1,SIE
7,Sanaa,ELHAOUZI,,G3,GG4,SIE
8,Damien,FERRANT,,G1,GG1,SIE
9,Loic,GUILLEMIN,,G5,GG2,SIE
10,Marouane,HAMOUMI,,titre,GG3,SIE
11,Romain,HORY,,G2,GG4,SIE
12,Julio Raphael,KAMGAING,,G3,GG1,SIE
13,Mathieu,KLEIN,,G3,GG2,SIE
14,Youri,KOMUSSIDI,,G5,GG3,SIE
15,Anthony,LEPRETRE,,G1,GG4,SIE
16,Mounir,MEJAIT,,G1,GG1,SIE
17,Clement,RIVELLINI,,ESSTIN,GG4,SIE
18,Ghizlane,SALHI,,G1,GG2,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
20,Laurie,TAN-LUONG-ANN,,G2,GG1,SIE
21,Mickaëlle,THOMAS,,G4,GG2,SIE
22,Clara,WIATROWSKI,,G5,GG1,SIE
23,Nicolas,SIESTE,,G20,GG40,IL
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep "SIE$" ./sie/liste_2a_sie.csv
```

```
1,Axel,AUPIC,,G3,GG2,SIE
2,Dhouha,AYED,,G5,GG3,SIE
3,Margaux,BERTHOU,,G2,GG3,SIE
4,Elise,BOINNOT,,G3,GG4,SIE
5,Timothee,CORDIER,,G5,GG1,SIE
6,Nicolas,EISEN,,G1,GG1,SIE
7,Sanaa,ELHAOUZI,,G3,GG4,SIE
8,Damien,FERRANT,,G1,GG1,SIE
9,Loic,GUILLEMIN,,G5,GG2,SIE
10,Marouane,HAMOUMI,,titre,GG3,SIE
11,Romain,HORY,,G2,GG4,SIE
12,Julio Raphael,KAMGAING,,G3,GG1,SIE
13,Mathieu,KLEIN,,G3,GG2,SIE
14,Youri,KOMUSSIDI,,G5,GG3,SIE
15,Anthony,LEPRETRE,,G1,GG4,SIE
16,Mounir,MEJAIT,,G1,GG1,SIE
17,Clement,RIVELLINI,,ESSTIN,GG4,SIE
18,Ghizlane,SALHI,,G1,GG2,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
20,Laurie,TAN-LUONG-ANN,,G2,GG1,SIE
21,Mickaëlle,THOMAS,,G4,GG2,SIE
22,Clara,WIATROWSKI,,G5,GG1,SIE
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) grep

- Exemple d'utilisation d'**expressions régulières** (2/2)

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep "G2" ./sie/liste_2a_sie.csv
1,Axel,AUPIC,,G3,G2,SIE
3,Margaux,BERTHOU,,G2,GG3,SIE
9,Loic,GUILLEMIN,,G5,G2,SIE
11,Romain,HORY,,G2,GG4,SIE
13,Mathieu,KLEIN,,G3,G2,SIE
18,Ghizlane,SALHI,,G1,G2,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
20,Laurie,TAN-LUONG-ANN,,G2,GG1,SIE
21,Mickaelle,THOMAS,,G4,G2,SIE
23,Nicolas,SIESTE,,G20,GG40,IL
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ grep "\<G2" ./sie/liste_2a_sie.csv
3,Margaux,BERTHOU,,G2,GG3,SIE
11,Romain,HORY,,G2,GG4,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
20,Laurie,TAN-LUONG-ANN,,G2,GG1,SIE
23,Nicolas,SIESTE,,G20,GG40,IL
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) grep

- Accéder au contenu d'un fichier (toujours)
 - Utilisation conjointe de grep avec cat ou find
 - `cat fichier1 | grep my_searched_string`
 - `find dir -name fichier1 | xargs grep mystring`
 - xargs permet de créer une commande à partir

```
coulet@potrero:~/enseignement/ésial/gmd/dossier_test$ cat ../sie/liste_2a_sie.csv | grep Damien
8,Damien,FERRANT,,G1,GG1,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
coulet@potrero:~/enseignement/ésial/gmd/dossier_test$ find . -name "liste_2a_sie.csv" | xargs grep Damien
8,Damien,FERRANT,,G1,GG1,SIE
19,Damien,SCHIRM,,G2,GG3,SIE
```


I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

- Accéder au contenu d'un fichier b) grep
 - Exemple : trouver les élèves de 2a du groupe

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ find . -name "liste_2a_*.csv"
./le/liste_2a_le.csv
./sie/liste_2a_sie.csv
./il/liste_2a_il.csv
./liste_2a_autre.csv
./trs/liste_2a_trs.csv
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ find . -name "liste_2a_*.csv" | xargs cat | grep GG4
1,Anthony,BARONE,,G2,GG4,LE
7,Jeremy,DOMARIN,,G3,GG4,LE
10,Jeff,GUYONNEAU,,G5,GG4,LE
13,Benoit,LEVEQUE,,G5,GG4,LE
18,Gilles,SEGHAIER,,G3,GG4,LE
20,Damien,THINUS,,G1,GG4,LE
4,Elise,BOINNOT,,G3,GG4,SIE
7,Sanaa,ELHAOUZI,,G3,GG4,SIE
11,Romain,HORY,,G2,GG4,SIE
15,Anthony,LEPRETRE,,G1,GG4,SIE
17,Clement,RIVELLINI,,ESSTIN,GG4,SIE
8,Paul-Marie,DOMINGUEZ,,G5,GG4,IL
12,Gabriel,LARROQUE,,G3,GG4,IL
1,Heline,ACHIMOFF,,G4,GG4,X
2,Houssam,ALOUY,,G4,GG4,TRS
4,Kevin,BUZEAU,,G1,GG4,TRS
11,Yoannes,GIRARDIN,,G4,GG4,TRS
15,Johanna,KELLER,,G4,GG4,TRS
18,Sebastien,PARISOT,,G5,GG4,TRS
22,Julien,WENDLING,,G1,GG4,TRS
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

- Accéder au contenu d'un fichier b) grep
 - Exemple : trouver les élèves de 2a du groupe

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ find . -name "liste_2a_*.csv"
./le/liste_2a_le.csv
./sie/liste_2a_sie.csv
./il/liste_2a_il.csv
./liste_2a_autre.csv
./trs/liste_2a_trs.csv
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ find . -name "liste_2a_*.csv" | xargs cat | grep GG4
1,Anthony,BARONE,,G2,GG4,LE
7,Jeremy,DOMARIN,,G3,GG4,LE
10,Jeff,GUYONNEAU,,G5,GG4,LE
13,Benoit,LEVEQUE,,G5,GG4,LE
18,Gilles,SEGHAIER,,G3,GG4,LE
20,Damien,THINUS,,G1,GG4,LE
4,Elise,BOINNOT,,G3,GG4,SIE
7,Sanaa,ELHAOUZI,,G3,GG4,SIE
11,Romain,HORY,,G2,GG4,SIE
15,Anthony,LEPRETRE,,G1,GG4,SIE
17,Clement,RIVELLINI,,ESSTIN,GG4,SIE
8,Paul-Marie,DOMINGUEZ,,G5,GG4,IL
12,Gabriel,LARROQUE,,G3,GG4,IL
1,Heline,ACHIMOFF,,G4,GG4,X
2,Houssam,ALOUÏ,,G4,GG4,TRS
4,Kevin,BUZEAU,,G1,GG4,TRS
11,Yoannes,GIRARDIN,,G4,GG4,TRS
15,Johanna,KELLER,,G4,GG4,TRS
18,Sebastien,PARISOT,,G5,GG4,TRS
22,Julien,WENDLING,,G1,GG4,TRS
```

Et si on ne veut que les noms de familles...

Ou

si on veut trier la liste par ordre alphabétique
du nom de famille...?

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) awk

- **awk** : est un langage de script dédié à la manipulation de fichiers textes
 - Enregistrement (record), champ (field)
 - Bcp de « built-in variables »
 - NR nbr of records, FS field separator, FNR number of the read record number, FILENAME, etc.
 - \$0, \$1, \$2, etc.
 - IGNORECASE

I. Accéder et extraire des données

- un exemple complet ^① : dans un système de gestion de fichier,
à partir de fichiers textes
b) awk
quels sont les étudiants du groupe

```
#!/bin/sh

my_file="./sie/liste_2a_sie.csv";
my_sh_param="GG4";

cat $my_file | awk -v my_awk_param="$my_sh_param" '
BEGIN{FS=",";print "start parsing..."}
{
  read_line=$0;          #ex: 15,Anthony,LEPRETRE,,G1,GG4,SIE
  student_id=$1;        # $1 contains the value of the first field : 15 in previous example
  student_first_name=$2;
  student_last_name=$3;
  group1=$5;
  group2=$6;
  app=$7;
  if(my_awk_param==group2){
    print student_first_name,student_last_name,"(record n°:",FNR,") is in group",my_awk_param;
  }
}
END{print "parsing done."}
'
```

Trois blocs dont BEGIN et END ; Passage de param avec -v ; FS ; \$0 ; \$1, etc.

I. Accéder et extraire des données

- un exemple complet : ^①

dans un système de gestion de fichier,
à partir de fichiers textes

b) awk

quels sont les étudiants du groupe

```
#!/bin/sh

my_file="./sie/liste_2a_sie.csv";
my_sh_param="GG4";

cat $my_file | awk -v my_awk_param="$my_sh_param" '
BEGIN{FS=",";print "start parsing..."}
{
  read_line=$0;          #ex: 15,Anthony,LEPRETRE,,G1,GG4,SIE
  student_id=$1;         #$1 contains the value of the first field : 15 in previous example
  student_first_name=$2;
  student_last_name=$3;
  group1=$5;
  group2=$6;
  app=$7;
  if(my_awk_param==group2){
    print student_first_name,student_last_name,"(record n°:",FNR,") is in group",my_awk_param;
  }
}
END{print "parsing done."}
'
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sh ./awk101.sh
start parsing...
Elise BOINNOT (record n°: 4 ) is in group GG4
Sanaa ELHAOUZI (record n°: 7 ) is in group GG4
Romain HORY (record n°: 11 ) is in group GG4
Anthony LEPRETRE (record n°: 15 ) is in group GG4
Clement RIVELLINI (record n°: 17 ) is in group GG4
parsing done.
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

- un exemple complet :

b) awk

Il est possible de combiner le mécanisme enregistrement/champs avec les expressions

```
#!/bin/sh
my_file="./sie/liste_2a_sie.csv";
my_sh_param="GG4";

cat $my_file | awk -v my_awk_param="$my_sh_param" '
BEGIN{FS=",";print "start parsing..."}
{
  read_line=$0;           #ex: 15,Anthony,LEPRETRE,,G1,GG4,SIE
  student_id=$1;         # $1 contains the value of the first field : 15 in previous example
  student_first_name=$2;
  student_last_name=$3;
  group1=$5;
  group2=$6;
  app=$7;
  if(group2 ~ "^"my_awk_param"$"){
    print student_first_name,student_last_name,"(record n°:",FNR,") is in group",my_awk_param;
  }
}
END{print "parsing done."}
'
```


I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) awk

- Remarque : **awk** peut aussi être appelé en ligne de commande

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ awk '/\/ { print $0 }' ./sie/liste_2a_sie.csv
4,Elise,BOINNOT,,G3,GG4,SIE
7,Sanaa,ELHAOUZI,,G3,GG4,SIE
11,Romain,HORY,,G2,GG4,SIE
15,Anthony,LEPRETRE,,G1,GG4,SIE
17,Clement,RIVELLINI,,ESSTIN,GG4,SIE
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) awk

- Avec **awk**

Il est facile de ne récupérer qu'une partie de la ligne.

Il est facile de parser :

- des fichiers systèmes, ex : /etc/fstab

```
## /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc nodev,noexec,nosuid 0 0
# / was on /dev/sda1 during installation
UUID=f6cea3ec-5b26-4427-b733-de0605a89187 / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=72ab2c2b-c75f-433e-9b59-24665c51acle none swap sw 0 0
# mount the shared dir of virtualbox
coulet /mnt/mac/coulet vboxsf defaults 0 0
```

- des fichiers de log

- Des fichiers .csv (+++)

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) awk

- Un outil puissant de **awk** : les tableaux

```
#!/bin/sh
my_dir=".";
my_regex="liste_2a_*.csv";
my_sh_param="GG4";

find "$my_dir" -type f -name "$my_regex" | xargs cat | awk -v my_awk_param="$my_sh_param" '
BEGIN{FS=",";print "start parsing...";in_gg4=0}
{
  tab[$3" "$2]=$6   #ex: tab[LEPRETRE Anthony]=GG4
}
END{
  print "parsing done.";
  print "les etudiants du "my_awk_param" sont : "
  for (x in tab){
    if(tab[x]==my_awk_param){
      print x;
      in_gg4++;
    }
  }
  print "total="in_gg4" étudiants."
}
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) awk

• Un outil puissant de **awk** : les tableaux

```
#!/bin/sh
my_dir=".";
my_regex="liste_2a_*.csv";
my_sh_param="GG4";

find "$my_dir" -type f -name "$my_regex" |
BEGIN{FS=",";print "start parsing..."}
{
  tab[$3" "$2]=$6 #ex: tab[LEPRETRE]
}
END{
  print "parsing done.";
  print "les etudiants du "my_awk_param";
  for (x in tab){
    if(tab[x]==my_awk_param){
      print x;
      in_gg4++;
    }
  }
  print "total="in_gg4" étudiants."
};
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ ./awk104_tabl.sh
start parsing...
parsing done.
les etudiants du GG4 sont :
LEVEQUE Benoit
THINUS Damien
LARROQUE Gabriel
BARONE Anthony
ACHIMOFF Helene
BOINNOT Elise
LEPRETRE Anthony
WENDLING Julien
ELHAOUZI Sanaa
DOMINGUEZ Paul-Marie
KELLER Johanna
SEGHAIER Gilles
GUYONNEAU Jeff
ALOUÏ Houssam
BUZEAU Kevin
HORY Romain
RIVELLINI Clement
PARISOT Sebastien
DOMARIN Jeremy
GIRARDIN Yoannes
total=20 étudiants.
```


I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- **Python:** est un langage de script
 - Non linux dépendent (compare to grep, sed, awk)
 - Un langage de programmation complet (Objets, GUI, etc.)
 - Facile à prendre en main, développement rapide (bon pour le prototypage)

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- Absence de délimiteur de bloc
 - i.e., pas d'accolade ou de do/done, if/fin
 - Une indentation significative
 - Début de bloc ':'
 - Continuation de bloc: position par rapport à la marge de gauche
 - Évitez de mélanger espaces et tabulations
- ';' en fin d'instruction facultatif
 - Peut servir pour 2 instructions sur une seule ligne

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- Premier exemple de script python :
quels sont les nombres premiers

```
#!/usr/bin/python

def premier(n) :
    i=n-1
    while (i>1):
        if ( (n % i) == 0 ) :
            return (0)
        i = i -1
    return (1)

j=1
while (j<500) :
    if (premier (j)) :
        print j,
    j = j+1
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ ./python101.py
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 1
93 197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 29
3 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409
419 421 431 433 439 443 449 457 461 463 467 479 487 491 499
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- Structures de contrôle

- while

```
while test :  
actions_boucles  
else :  
actions_fin
```

NB : actions_fin ne sont exécutées que si test prend un valeur fausse (ou nulle). Si un break fait sortir du while sans test=faux, alors actions_fin ne sont pas exécutées

- for

```
for variable in enumeration :  
actions_for  
else :  
actions_else
```

le for permet de parcourir les éléments d'une énumération (i.e., liste, chaîne de caractères, etc.)

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- Structures de contrôle
 - for (exemple)

```
coulet@potrero:~$ python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> for i in ["il", "le", "sie", "trs","autre"] :
...     print i
...
il
le
sie
trs
autre
>>>
>>>
coulet@potrero:~$
```

mode interactif, ctrl+D

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- Structures de contrôle

- if

```
if test :  
    actions_if  
elif autre_test :  
    actions_elif  
else :  
    action_else
```

NB : test ne nécessite pas d'être entre parenthèses sauf si test est sur plusieurs lignes

NB2 : si une seule action, on peut écrire test et action sur la même ligne

```
if test : action1  
else : action2
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

• Fonctions

- mot clé def + nom de la fct + (arg)
- déclaration
- Fin de déclaration avec return sinon None est retourné
- Voir premier exemple comme illustré
- Porté des variables : global

```
#!/usr/bin/python
x = "valeur globale"

def fonction_1() :
    print "dans fonction_1 :", x

def fonction_2() :
    x = "valeur locale"
    print "dans fonction_2 :", x

def fonction_3() :
    global x
    x = "valeur modifiée"
    print "dans fonction_3 :", x

print "Au niveau global :", x
fonction_1()
print "Au niveau global :", x
fonction_2()
print "Au niveau global :", x
fonction_3()
print "Au niveau global :", x
```



```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ python ./python102.py
Au niveau global : valeur globale
dans fonction_1 : valeur globale
Au niveau global : valeur globale
dans fonction_2 : valeur locale
Au niveau global : valeur globale
dans fonction_3 : valeur modifiée
Au niveau global : valeur modifiée
coulet@potrero:~/enseignement/esial/gmd/dossier_test$
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

- Arguments de fonctions

c) python

- Définition de mots clés et de valeurs par défaut

```
#!/usr/bin/python
x = "valeur globale"

def init_port(vitesse=9600, parite="paire", nb_bits=8, nb_stop=1) :
    print "Initialisation a :", vitesse, "bits/s"
    print " Parite :", parite
    print " ", nb_bits, "bits de donnees"
    print " ", nb_stop, "bits d'arret"

init_port()
init_port(parite="nulle", vitesse=28800) #avec mots cles, l'ordre n'importe pas
init_port(2400,"impaire",7,2)          #sans mot cle, l'ordre importe
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ python ./python103.py
Initialisation a : 9600 bits/s
Parite : paire
 8 bits de donnees
 1 bits d'arret
Initialisation a : 28800 bits/s
Parite : nulle
 8 bits de donnees
 1 bits d'arret
Initialisation a : 2400 bits/s
Parite : impaire
 7 bits de donnees
 2 bits d'arret
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

• Arguments de fonctions (suite)

c) python

– Nombre d'arg variable et arg facultatifs

```
#!/usr/bin/python

def fct1(a,b,*args): #*args est un n-uplet recevant les arg restants
    print "Fct1  :"
    print " a =  :", a
    print " b =  :", b
    for arg in args:
        print " ", arg

def fct2(g,h,i="",j="",**args):#**args est un tableau associatif
    print "Fct2  :"
    print " g =  :", g
    print " h =  :", h
    print " i =  :", i
    print " j =  :", j
    for cle in args.keys():
        print " ", cle, " = ",args[cle]

def fct3(**args):
    print "Fct3  :"
    liste = args.keys()
    liste . sort()
    for cle in args.keys():
        print " ", cle, " = ",args[cle]

fct1("alpha","bravo","charlie","delta","echo","fox-trot")
fct2("golf","hotel",j="juliet",i="india",k="kilo",l="lima")
fct3(n="november",m="mike",p="papa",o="oscar")
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ python ./python104.py
Fct1  :
a = : alpha
b = : bravo
charlie
delta
echo
fox-trot
Fct2  :
g = : golf
h = : hotel
i = : india
j = : juliet
k = : kilo
l = : lima
Fct3  :
p = : papa
m = : mike
o = : oscar
n = : november
```

Fctions prédéfinies keys, sort, e

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

• Types de données

c) python

- comme pour bash et awk, python utilise un **typage dynamique**
- Pas de déclaration explicite nécessaire
- Pendant : pas de conversion explicite

```
>a="2"
```

```
>b=a+5
```

```
TypeError: Illegal argument type
```

- Numérique

Fonctions dédiées : `abs(-5)`, `max(2,4,-10)`, et bcp d'autres dans le module `math` (ex : `cos`, `exp`, `log`, `sqrt`, etc.) Remarque : `import math` importe le module

Ex de conversion possibles :

```
>>> a="45.2"
>>> float(a)
45.200000000000003
>>> int(a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: '45.2'
>>> int(float(a))
45
```

- Chaînes : `ma_chaine[3]`, "l'apostrophe" , ""texte sur plusieurs lignes""

le module `string` contient bcp de fct utiles `import string`

```
i=string.find(chaine, souschaine)
```

```
i=string.replace(chaine, ssch1, ssch2)
```

```
l=string.split(chaine, sep)
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

• Types de données (suite)

c) python

– Listes: `l=["oui", "non", 20, 10, [12,'flan',3]]`

objets associé à des méthodes simples :

`ma_liste.sort()`

– N-uplets : listes immuables

`nup=("oui", "non", 20)`

– Dico ou tableau associatif

`dico={"lun":"Monday", "mar":"Tuesday"}`

`>print dico["lun"]`

Monday

`>len(dico)`

2

aux dictio

```
#!/usr/bin/python
import sys
liste=sys.stdin.readlines()
liste.sort()
sys.stdout.writelines(liste)
sys.stderr.write("%d lignes" % len(liste))
```

`cat ./sie/liste_2a_sie.csv | python python106.py`

```
#!/usr/bin/python
```

```
mon_fichier=open("./sie/liste_2a_sie.csv")
for line in mon_fichier:
    print line,
```

```
#!/usr/bin/python
try:
    mon_fichier=open("./sie/liste_2a_sie.csv")
    for line in mon_fichier:
        print line,
    finally: mon_fichier.close()
```

```
#!/usr/bin/python
import fileinput
for line in fileinput.input():
    print line,
```

`python python107.py ./sie/liste_2a_sie.csv`

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

- Programmation objet : exemple simple

```
#!/usr/bin/python
class Paire:
    x=0
    y=0
    add=0
    def addition (self):
        self.add=self.x+self.y
    def print_add(self):
        print self.x, " + ",self.y, " = ",self.add

p=Paire()
p.x=2
p.y=2
p.addition()
p.print_add()
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ python python108.py
2 + 2 = 4
```

I. Accéder et extraire des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

c) python

• Exemple d'extraction de données

```
#!/usr/bin/python
import sys
import os
import fnmatch
import fileinput
import re

path=sys.argv[1] #get the path to explore in arg
group=sys.argv[2] #gg4
group_list = []
for root,dirs,files in os.walk(path): # walk browse top down the dir tree
    for file in files:
        filepath=os.path.join(root, file)
        if (fnmatch.fnmatch(filepath,"*liste_2a_*.csv")):
            try:
                readfile=open(filepath)
                for line in readfile:
                    fields=re.split(",",line)
                    if (fields[5]==group):
                        group_list.append(fields[2]+" "+fields[1])
            finally:readfile.close()
group_list.sort()
for student in group_list:
    print student
```

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ python python109.py
/home/coulet/enseignement/esial/gmd/dossier_test/ GG4
ACHIMOFF Heline
ALOUI Houssam
BARONE Anthony
BOINNOT Elise
BUZEAU Kevin
DOMARIN Jeremy
DOMINGUEZ Paul-Marie
ELHAOUZI Sanaa
GIRARDIN Yoannes
GUYONNEAU Jeff
HORY Romain
KELLER Johanna
LARROQUE Gabriel
LEPRETRE Anthony
LEVEQUE Benoit
PARISOT Sebastien
RIVELLINI Clement
SEGHAIER Gilles
THINUS Damien
WENDLING Julien
```

I. Accéder et extraire des données

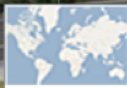
①

dans un système de gestion de fichier,
à partir de fichiers textes

• Info Pratiques

c) python

- la dernière version stable est 3.4 (le 05/01/14)
- doc python : <http://docs.python.org/library/>
- éditeurs (du plus léger au plus lourd)
 - emacs, vim
 - iPython, Spyder
 - Eclipse



Map

Traffic

100 ft
50 m



II. Transformer des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

• Recherche de motifs

a) bash

- `${variable#motif}`: élimination du plus court préfix
- `${variable##motif}` : élimination du plus long préfix
- `${variable%motif}` : élimination du plus court suffixe
- `${variable%%motif}` : élimination du plus long suffixe

```
coulet@potrero:~$ var=haveaniceone
coulet@potrero:~$ echo ${var#have}
aniceone
coulet@potrero:~$ echo ${var#*a}
veaniceone
coulet@potrero:~$ echo ${var##*a}
niceone
```

- `${variable/motif/remplacement}` : remplace la première occurrence du motif par le remplacement
- `${variable//motif/remplacement}` : remplace toutes les occurrences du motif par le remplacement

```
coulet@potrero:~$ echo ${var/one/day}
haveaniceday
coulet@potrero:~$ echo ${var//e/\'}
hav'anic'on'
```

II. Transformer des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

b) sed

– sed est

- une command Unix puissante

```
sed -e 's/Ancien/Nouveau/g' nomFichierEntrée > nomFichierSortie
```

- un langage de script

```
sed -f subst.sed nomFichierEntrée > nomFichierSortie
```

– ...qui permet de transformer/filtrer le contenu de fichiers textes efficacement

II. Transformer des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

– Exemples (1/2)

b) sed

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sed -e s/SIE/"Système d'Information d'Entreprises"/g ./sie/liste_2a_sie.csv
1,Axel,AUPIC,,G3,GG2,Système d'Information d'Entreprises
2,Dhouha,AYED,,G5,GG3,Système d'Information d'Entreprises
3,Margaux,BERTHOU,,G2,GG3,Système d'Information d'Entreprises
4,Elise,BOINNOT,,G3,GG4,Système d'Information d'Entreprises
5,Timothee,CORDIER,,G5,GG1,Système d'Information d'Entreprises
6,Nicolas,EISEN,,G1,GG1,Système d'Information d'Entreprises
7,Sanaa,ELHAOUZI,,G3,GG4,Système d'Information d'Entreprises
8,Damien,FERRANT,,G1,GG1,Système d'Information d'Entreprises
9,Loic,GUILLEMIN,,G5,GG2,Système d'Information d'Entreprises
10,Marouane,HAMOUMI,,titre,GG3,Système d'Information d'Entreprises
11,Romain,HORY,,G2,GG4,Système d'Information d'Entreprises
12,Julio Raphael,KAMGAING,,G3,GG1,Système d'Information d'Entreprises
13,Mathieu,KLEIN,,G3,GG2,Système d'Information d'Entreprises
14,Youri,KOMUSSIDI,,G5,GG3,Système d'Information d'Entreprises
15,Anthony,LEPRETRE,,G1,GG4,Système d'Information d'Entreprises
16,Mounir,MEJAÏT,,G1,GG1,Système d'Information d'Entreprises
17,Clement,RIVELLINI,,ESSTIN,GG4,Système d'Information d'Entreprises
18,Ghizlane,SALHI,,G1,GG2,Système d'Information d'Entreprises
19,Jamien,SCHIRM,,G2,GG3,Système d'Information d'Entreprises
20,Laurie,TAN-LUONG-ANN,,G2,GG1,Système d'Information d'Entreprises
21,Mickaëlle,THOMAS,,G4,GG2,Système d'Information d'Entreprises
22,Clara,WIATROWSKI,,G5,GG1,Système d'Information d'Entreprises
23,Nicolas,Système d'Information d'EntreprisesSTE,,G20,GG4,IL
```


II. Transformer des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

– Exemples (1/2)

b) sed

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sed -e s/SIE/"Système d'Information d'Entreprises"/g ./sie/liste_2a_sie.csv
1,Axel,AUPIC,,G3,GG2,Système d'Information d'Entreprises
2,Dhouha,AYED,,G5,GG3,Système d'Information d'Entreprises
3,Margaux,BERTHOU,,G2,GG3,Système d'Information d'Entreprises
4,Elise,BOINNOT,,G3,GG4,Système d'Information d'Entreprises
5,Timothee,CORDIER,,G5,GG1,Système d'Information d'Entreprises
6,Nicolas,EISEN,,G1,GG1,Système d'Information d'Entreprises
7,Sanaa,ELHAOUZI,,G3,GG4,Système d'Information d'Entreprises
8,Damien,FERRANT,,G1,GG1,Système d'Information d'Entreprises
9,Loic,GUILLEMIN,,G5,GG2,Système d'Information d'Entreprises
10,Marouane,HAMOUMI,,titre,GG3,Système d'Information d'Entreprises
11,Romain,HORY,,G2,GG4,Système d'Information d'Entreprises
12,Julio Raphael,KAMGAING,,G3,GG1,Système d'Information d'Entreprises
13,Mathieu,KLEIN,coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sed -e s/SIE$/"Système d'Information d'Entreprises"/g ./sie/liste_2a_sie.csv
14,Youri,KOMUSSIDI,1,Axel,AUPIC,,G3,GG2,Système d'Information d'Entreprises
15,Anthony,LEPRETRE,2,Dhouha,AYED,,G5,GG3,Système d'Information d'Entreprises
16,Mounir,MEJAIT,3,Margaux,BERTHOU,,G2,GG3,Système d'Information d'Entreprises
17,Clement,RIVELLINI,4,Elise,BOINNOT,,G3,GG4,Système d'Information d'Entreprises
18,Ghizlane,SALHI,5,Timothee,CORDIER,,G5,GG1,Système d'Information d'Entreprises
19,Damien,SCHIRM,6,Nicolas,EISEN,,G1,GG1,Système d'Information d'Entreprises
20,Laurie,TAN-LUONG-ANN,7,Sanaa,ELHAOUZI,,G3,GG4,Système d'Information d'Entreprises
21,Mickaëlle,THOMAS,8,Damien,FERRANT,,G1,GG1,Système d'Information d'Entreprises
22,Clara,WIATROWSKI,9,Loic,GUILLEMIN,,G5,GG2,Système d'Information d'Entreprises
23,Nicolas,SIESTE,10,Marouane,HAMOUMI,,titre,GG3,Système d'Information d'Entreprises
11,Romain,HORY,,G2,GG4,Système d'Information d'Entreprises
12,Julio Raphael,KAMGAING,,G3,GG1,Système d'Information d'Entreprises
13,Mathieu,KLEIN,,G3,GG2,Système d'Information d'Entreprises
14,Youri,KOMUSSIDI,,G5,GG3,Système d'Information d'Entreprises
15,Anthony,LEPRETRE,,G1,GG4,Système d'Information d'Entreprises
16,Mounir,MEJAIT,,G1,GG1,Système d'Information d'Entreprises
17,Clement,RIVELLINI,,ESSTIN,GG4,Système d'Information d'Entreprises
18,Ghizlane,SALHI,,G1,GG2,Système d'Information d'Entreprises
19,Damien,SCHIRM,,G2,GG3,Système d'Information d'Entreprises
20,Laurie,TAN-LUONG-ANN,,G2,GG1,Système d'Information d'Entreprises
21,Mickaëlle,THOMAS,,G4,GG2,Système d'Information d'Entreprises
22,Clara,WIATROWSKI,,G5,GG1,Système d'Information d'Entreprises
23,Nicolas,SIESTE,,G20,GG40,IL
```

II. Transformer des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

– Exemples (1/2)

b) sed

```
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sed -e s/SIE/"Système d'Information d'Entreprises"/g ./sie/liste_2a_sie.csv
1,Axel,AUPIC,,G3,GG2,Système d'Information d'Entreprises
2,Dhouha,AYED,,G5,GG3,Système d'Information d'Entreprises
3,Margaux,BERTHOU,,G2,GG3,Système d'Information d'Entreprises
4,Elise,BOINNOT,,G3,GG4,Système d'Information d'Entreprises
5,Timothee,CORDIER,,G5,GG1,Système d'Information d'Entreprises
6,Nicolas,EISEN,,G1,GG1,Système d'Information d'Entreprises
7,Sanaa,ELHAOUZI,,G3,GG4,Système d'Information d'Entreprises
8,Damien,FERRANT,,G1,GG1,Système d'Information d'Entreprises
9,Loic,GUILLEMIN,,G5,GG2,Système d'Information d'Entreprises
10,Marouane,HAMOUMI,,titre,GG3,Système d'Information d'Entreprises
11,Romain,HORY,,G2,GG4,Système d'Information d'Entreprises
12,Julio Raphael,KAMGAING,,G3,GG1,Système d'Information d'Entreprises
13,Mathieu,KLEIN,coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sed -e s/SIE$/"Système d'Information d'Entreprises"/g ./sie/liste_2a_sie.csv
14,Youri,KOMUSSIDI1,Axel,AUPIC,,G3,GG2,Système d'Information d'Entreprises
15,Anthony,LEPRETRE2,Dhouha,AYED,,G5,GG3,Système d'Information d'Entreprises
16,Mounir,MEJAIT3,Margaux,BERTHOU,,G2,GG3,Système d'Information d'Entreprises
17,Clement,RIVELLINI4,Elise,BOINNOT,,G3,GG4,Système d'Information d'Entreprises
18,Ghizlane,SALHI5,Timothee,CORDIER,,G5,GG1,Système d'Information d'Entreprises
19,Damien,SCHIRM6,Nicolas,EISEN,,G1,GG1,Système d'Information d'Entreprises
20,Laurie,TAN-LUONG-ANN7,Sanaa,ELHAOUZI,,G3,GG4,Système d'Information d'Entreprises
21,Mickaelle,THOMAS8,Damien,FERRANT,,G1,GG1,Système d'Information d'Entreprises
22,Clara,WIA9,Loic,GUILLEMIN,,G5,GG2,Système d'Information d'Entreprises
23,Nicolas,SY10,Marouane,HAMOUMI,,titre,GG3,Système d'Information d'Entreprises
11,Romain,HORY,,G2,GG4,Système d'Information d'Entreprises
12,Julio Raphael,KAMGAING,,G3,GG1,Système d'Information d'Entreprises
13,Mathieu,KLEIN,,G3,GG2,Système d'Information d'Entreprises
14,Youri,KOMUSSIDI,,G5,GG3,Système d'Information d'Entreprises
15,Anthony,LEPRETRE,,G1,GG4,Système d'Information d'Entreprises
16,Mounir,MEJAIT,,G1,GG1,Système d'Information d'Entreprises
17,Clement,RIVELLINI,,ESSTIN,GG4,Système d'Information d'Entreprises
18,Ghizlane,SALHI,,G1,GG2,Système d'Information d'Entreprises
19,Damien,SCHIRM,,G2,GG3,Système d'Information d'Entreprises
20,Laurie,TAN-LUONG-ANN,,G2,GG1,Système d'Information d'Entreprises
21,Mickaelle,THOMAS,,G4,GG2,Système d'Information d'Entreprises
coulet@potrero:~/enseignement/esial/gmd/dossier_test$ sed -e s/SIE$/"Système d'Information d'Entreprises"/g ./sie/liste_2a_sie.csv >./sie/liste_2a_sie_bis.csv
```

II. Transformer des données

①

dans un système de gestion de fichier,
à partir de fichiers textes

– Exemples (2/2)

b) sed

Identification de groupes

- Permuter deux éléments d'un champ séparés par un tiret

```
s/\(.*\) - \(.*\) / \2 - \1 /
```

- Afficher les *n* derniers caractères

```
s/\(.*\) \(. \{n\} \) / \2 /
```

Gestion de plusieurs lignes

- Joindre à la ligne précédentes les lignes qui commencent par un espace

```
sed 'N;s/\n / /g;P;D;'
```

; sépare les commandes

N ajoute une ligne au tampon

P affiche le résultat en mémoire

D initialise le tampon et passe à la ligne suivante

```
Voici mon chat,  
il s'appelle Poupougne.  
Voici mon chien,  
il s'appelle Cinoke.
```

—————>

```
Voici mon chat, il s'appelle Poupougne.  
Voici mon chien, il s'appelle Cinoke.
```

- Rérérences

- Christophe Blaess, **Script Sous Linux**, Eyrolles, 2004