ADS Lab 03 - Pipelines

Gabriel Roch Gwendoline Dössegger

10 mars 2021

1 Task1 - Exercices ON Redirection

Question 1 Run the following commands and tell where stdout and stderr are redirected to.

1. `./out > file`
   Stdout : est redirigé dans file
   Stderr : est redirigé dans le terminal
2. `./out 2> file`
   Stdout : est redirigé dans le terminal
   Stderr : est redirigé dans le fichier
3. `./out 2>&1 > file`
   Stdout : est redirigé dans file
   Stderr : est redirigé dans le terminal
4. `./out > file 2>&1`
   Stdout et Stderr sont redirigés dans file

Question 2 What do the following commands do ?

1. `cat /usr/share/doc/bash/README | grep -i posix`
   L'objectif était d'afficher toutes les lignes contenant `posix` indépendanment de la casse présent dans le fichier README. Cependant, le fichier n'existant pas, grep n'affiche rien mais cat indique que le fichier n'existe pas.
2. `./out 2>&1 | grep -i eeeee`
   Cherche `eeeee` dans la sortie de .out/ (stderr et stdout). Rien n'est trouvé car la sortie est une altérnence de OE et donc aucune suite de `eeeee`. La recherche se fait en ignorant la casse.
3. `./out 2>&1 >/dev/null | grep -i eeeee`
   Cherche `eeeee` dans la sortie d'erreur de out (stderr). La ligne est trouvée et affichée dans le terminal `EEEEE`. La recherche se fait en ignorant la casse.

Question 3 Write commands to perform the following tasks :

1. Produce a recursive listing, using `ls`, of files and directories in your home directory, including hidden files, in the file `/tmp/homefiles`.
   `ls -all -R ~ > /tmp/homefiles`

2. Produce a (non-recursive) listing of all files in your home directory whose names end in .txt, .md or .pdf, in the file /tmp/documents. The command must not display an error message if there are no corresponding files.

Nous avons trouvé les deux solutions suivantes :
- `find ~ -maxdepth 1 -name '*.txt' -or -name '*.md' -or -name '*.pdf'`
- `ls -all ~/*.{md,txt,pdf} 2> /dev/null`

2 Task2 - LOG ANALYSIS

Nous pouvons constaté qu'avec la commande `xxd`, que le séparateur des champs de la première ligne est le caractère 0x09 qui correspond à la tabulation.

1. How many log entries are in the file ?
   2781 lignes
   Commande utilisée : `wc -l ads_website.log`
2. How many accesses were successful (server sends back a status of 200) and how many had an error of "Not Found" (status 404) ?
   Nous affichons le nombre de fois que les codes de status du serveur apparaissent dans les log avec la commande suivante : `cut ads_website.log -f10 | sort | uniq -c`
   - Statut 200, 1610 occurences
   - Statut 404, 21 occurences
3. What are the URIs that generated a "Not Found" response ? Be careful in specifying the correct search criteria : avoid selecting lines that happen to have the character sequence 404 in the URI.
   `cut ads_website.log -f9-10 | grep "404$" | sort -u | cut -f1 | tr '"' ' ' | cut -d ' ' -f3`
   — /heigvd-ads ?cors
   — /heigvd-ads ?lifecycle
   — /heigvd-ads ?policy
   — /heigvd-ads ?website
4. How many different days are there in the log file on which requests were made ?
   `cut -f3 ads_website.log | cut -c2-12 | sort -u | wc -l`
   Il y a 21 jours où il y eu des requêtes.
5. How many accesses were there on 4th March 2014 ?
   Nous avons trouvé les deux commandes suivantes :
   - `cut -f3 ads_website.log | cut -c2-12 | grep "04/Mar/2014" | wc -l`
   - `cut -f3 ads_website.log | cut -c2-12 | sort | uniq -c | grep "04/Mar/2014"`
   Il y a eu 423 accès.
6. Which are the three days with the most accesses ? Hint : Create first a pipeline that produces a list of dates preceded by the count of log entries on that date.
   `cut -f3 ads_website.log | cut -c2-12 | sort | uniq -c | sort -gr | head -3`
   Les 4, 6 et 13 mars 2014
7. Which is the user agent string with the most accesses ?
   `cut ads_website.log -f17 | sort | uniq -c | sort -nr | head -1`
   `"Mozilla/5.0 (Windows NT 6.3; WOW64; rv:27.0) Gecko/20100101 Firefox/27.0"`
   avec 423 occurences.

8. If a web site is very popular and accessed by many people the user agent strings appearing in the server's log can be used to estimate the relative market share of the users' computers and operating systems. How many accesses were done from browsers that

declare that they are running on Windows, Linux and Mac OS X ?
- Windows : 1751, avec la commande `cut ads_website.log -f17 | grep -ci 'Windows'`
- Linux : 180, avec la commande `cut ads_website.log -f17 | grep -ci 'Linux'`
- Linux sans Android : 164, avec la commande `cut ads_website.log -f17 | grep -v 'Android' | grep -ci 'Linux'`
- Mac OS X : 693, avec la commande `cut ads_website.log -f17 | grep -ci 'Mac OS X'`
- Mac OS X sans Iphone & Ipad : 643, avec la commande `cut ads_website.log -f17 | grep -vEi 'iPhone|iPad' | grep -ci 'Mac OS X'`

9. Read the documentation for the tee command. Repeat the analysis of the previous question for browsers running on Windows and insert tee into the pipeline such that the user agent strings (including repeats) are written to a file for further analysis (the filename should be useragents.txt). What are the operating systems Windows NT 6.1, 6.2 and 6.3 ?
Commande : `cut ads_website.log -f17 | tee useragents.txt |grep -ci 'Windows'`
Windows NT 6.1 : Windows 7 ou Windows Server 2008R2
Windows NT 6.2 : Windows 8 ou Windows Server 2012
Windows NT 6.3 : Windows 8.1 ou Windows Server 2012R2

10. Why is the file access.log difficult to analyse, consider for example the analysis of question 7, with the commands you have seen so far ?
Le problème previent du fait que les champs sont séparés par des espaces. Ce séparateur est aussi présent à l'interieur des champs et donc cut n'arrive pas à traiter ces champs correctements.
La commande a utilisée est donc : `sed -E 's/^[0-9.]+ [^]+ [^]+ \[[^]]+\] "[^"]+" ([0-9]+).+/\1/g' access.log | sort -u`
Les code de status sont 206, 301, 304, 403, 404 et 408.

3 Task3 - Conversion to CSV

Question 1 Produce a CSV file named accesses.csv that contains for each day (given by its date) the number of accesses on that day. Transfer that file to your workstation and use spreadsheet software to import the CSV file. Plot the data in a graph and produce a file named accesses.pdf.

```
cut -f3 ads_website.log | cut -c2-12 | sort | uniq -c | grep -oiE "[0-9]+ [a-z0-9/]+" | tr ' ' ',' > access.csv
```

Le graphique est en annexe : access.pdf