

Linux Lab Unit 2

textmanipulation, pipes, output

Christian Kniep

Internation Center of Applied Technologies Bandung

5. August 2010

Table of content

- 1 Filesystem
 - searching
- 2 Handle the output
 - only last,first
 - seperate lines
 - redirect output

updatedb,root

- `updatedb` create an index of the whole directory-tree

updatedb,root

- `updatedb` create an index of the whole directory-tree
- So you have to be some superuser
 - ▶ The superuse root has no permission-restrictions

updatedb,root

- `updatedb` create an index of the whole directory-tree
- So you have to be some superuser
 - ▶ The superuse root has no permission-restrictions
 - ⇒ You can't hide from him!

updatedb,root

- `updatedb` create an index of the whole directory-tree
- So you have to be some superuser
 - ▶ The superuse root has no permission-restrictions
 - ⇒ You can't hide from him!
 - ▶ Due to security only root can change the owner of a file/directory

updatedb,root

- if a superuser has performed `updatedb` you can locate things:
`debian1:/var/linuxLab# locate Network`
`/var/linuxLab/irene/wednesday/ComputerNetwork`

updatedb,root

- if a superuser has performed `updatedb` you can locate things:
debian1:/var/linuxLab# `locate Network`
`/var/linuxLab/irene/wednesday/ComputerNetwork`
- Please search for `bandung` and discuss the problem with `updatedb`

updatedb,root

- if a superuser has performed `updatedb` you can locate things:
`debian1:/var/linuxLab# locate Network`
`/var/linuxLab/irene/wednesday/ComputerNetwork`
- Please search for `bandung` and discuss the problem with `updatedb`
 - 1 You will find this `/var/linuxLab/additional/bandung`

updatedb,root

- if a superuser has performed `updatedb` you can locate things:
`debian1:/var/linuxLab# locate Network`
`/var/linuxLab/irene/wednesday/ComputerNetwork`
- Please search for `bandung` and discuss the problem with `updatedb`
 - 1 You will find this `/var/linuxLab/additional/bandung`
 - 2 But you are not supposed to see it:
`stella@debian1:~$ ls /var/linuxLab/additional/bandung`
`ls: /var/linuxLab/additional/bandung: Permission denied`
`stella@debian1:~$ ls /var/linuxLab/additional`
`ls: /var/linuxLab/additional: Permission denied`

find

- `find` searches live

```
debian1:/var/linuxLab/additional# find ./  
./  
./bandung  
./testDir  
./testDir/testFile
```

find

- `find` searches live

```
debian1:/var/linuxLab/additional# find ./
./
./bandung
./testDir
./testDir/testFile
```

- use the manpage / help to search

- ① find only folders in `/var/linuxLab/<YOU>`

find

- `find` searches live

```
debian1:/var/linuxLab/additional# find ./
./
./bandung
./testDir
./testDir/testFile
```

- use the manpage / help to search

- 1 find only folders in `/var/linuxLab/<YOU>`
- 2 find only files in `/var/linuxLab/<YOU>`

find

- `find` searches live

```
debian1:/var/linuxLab/additional# find ./
./
./bandung
./testDir
./testDir/testFile
```

- use the manpage / help to search

- 1 find only folders in `/var/linuxLab/<YOU>`
- 2 find only files in `/var/linuxLab/<YOU>`
- 3 search for files which belongs to your user in `/var/linuxLab/`

tail,head

- if you only want to see the last 10 lines of a file: `tail file`

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`
- both provides `-n` to provide the amount of lines,e.g. `-n50`

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`
- both provides `-n` to provide the amount of lines,e.g. `-n50`
- also often used to wait for changes: `-F`

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`
- both provides `-n` to provide the amount of lines,e.g. `-n50`
- also often used to wait for changes: `-F`
- Lets play with it `/var/linuxLab/unit2/linenumbers.txt`
 - ❶ view the last 15 lines

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`
- both provides `-n` to provide the amount of lines,e.g. `-n50`
- also often used to wait for changes: `-F`
- Lets play with it `/var/linuxLab/unit2/linenumbers.txt`
 - 1 view the last 15 lines
 - 2 view the first 15 lines

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`
- both provides `-n` to provide the amount of lines,e.g. `-n50`
- also often used to wait for changes: `-F`
- Lets play with it `/var/linuxLab/unit2/linenumbers.txt`
 - 1 view the last 15 lines
 - 2 view the first 15 lines
 - 3 view the lines 20-25

tail,head

- if you only want to see the last 10 lines of a file: `tail file`
- the first 10 lines of a file: `head file`
- both provides `-n` to provide the amount of lines,e.g. `-n50`
- also often used to wait for changes: `-F`
- Lets play with it `/var/linuxLab/unit2/linenumbers.txt`
 - 1 view the last 15 lines
 - 2 view the first 15 lines
 - 3 view the lines 20-25
 - 4 use `-F` to wait for changes

grep

- to search for patterns within files you use `grep <pattern>file`

grep

- to search for patterns within files you use `grep <pattern>file`
- try to find the pattern 25 in the previous used file `linenumbers.txt`

grep

- to search for patterns within files you use `grep <pattern>file`
- try to find the pattern 25 in the previous used file `linenumbers.txt`
- try to find your username in the directory `/var/log/`,
for this use `grep -R <user> <path>*`

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?
- actually there are two different outputs:
`stdout` that's the normal output shown by the program
`stderr` warnings and errors

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?
- actually there are two different outputs:

`stdout` that's the normal output shown by the program

`stderr` warnings and errors

- `stdout` is referred to with `1>`, `stderr` with `2>`

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?
- actually there are two different outputs:

`stdout` that's the normal output shown by the program

`stderr` warnings and errors

- `stdout` is referred to with `1>`, `stderr` with `2>`
- let's try it:
 - 1 `ls /var/linuxLab/unit2/*`

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?
- actually there are two different outputs:

`stdout` that's the normal output shown by the program

`stderr` warnings and errors

- `stdout` is referred to with `1>`, `stderr` with `2>`
- let's try it:

① `ls /var/linuxLab/unit2/*`

② `ls /var/linuxLab/unit2/* 1> out`

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?
- actually there are two different outputs:

`stdout` that's the normal output shown by the program

`stderr` warnings and errors

- `stdout` is referred to with `1>`, `stderr` with `2>`
- let's try it:

- 1 `ls /var/linuxLab/unit2/*`
- 2 `ls /var/linuxLab/unit2/* 1> out`
- 3 `ls /var/linuxLab/unit2/* 2> err`

output

- To redirect the output into a file use `>`
e.g. `ls -l > file`
- what is the difference between `>` and `>>`?
- actually there are two different outputs:

`stdout` that's the normal output shown by the program

`stderr` warnings and errors

- `stdout` is referred to with `1>`, `stderr` with `2>`
- let's try it:

- 1 `ls /var/linuxLab/unit2/*`
- 2 `ls /var/linuxLab/unit2/* 1> out`
- 3 `ls /var/linuxLab/unit2/* 2> err`
- 4 `ls /var/linuxLab/unit2/* 1> out 2> err`