Chapter 2 - Unix Tutorial

Philipp Moritzer - 21170004

Cal

Displays a calendar on the terminal

```
$ cal # shows the calendar of current month
$ cal 2015 # shows the calendar for all months of 2015
$ cal 5 2015 # shows the calendar for May 2015
```

Output:

```
21170004@comunix:~$ cal 5 2015
   May 2015
 S  M Tu  W Th  F  S
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

```
$ whoami # displays user name
$ who am I #shows username, terminal number, login time, ip address
$ users # lists usernames of logged in users
$ who #displays who is logged in
```

Output:

```
21170004@comunix:~$ who
18101201   pts/1        Apr 13 13:07    (1.231.8.91)
20101281   pts/3        Apr 13 23:46    (120.142.100.142)
21170004   pts/4        Apr 14 00:09    (211.170.153.23)
khlee      pts/15       Apr 13 20:56    (117.17.198.87)
19100197   pts/16       Apr 13 20:02    (220.116.118.24)
21170004@comunix:~$ whoami
21170004
21170004@comunix:~$ who am I
21170004   pts/4        Apr 14 00:09    (211.170.153.23)
21170004@comunix:~$ users
18101201 19100197 20101281 21170004 khlee
21170004@comunix:~$ who
18101201   pts/1        Apr 13 13:07    (1.231.8.91)
20101281   pts/3        Apr 13 23:46    (120.142.100.142)
21170004   pts/4        Apr 14 00:09    (211.170.153.23)
khlee      pts/15       Apr 13 20:56    (117.17.198.87)
19100197   pts/16       Apr 13 20:02    (220.116.118.24)
```

ls

Lists all files or folders of directories

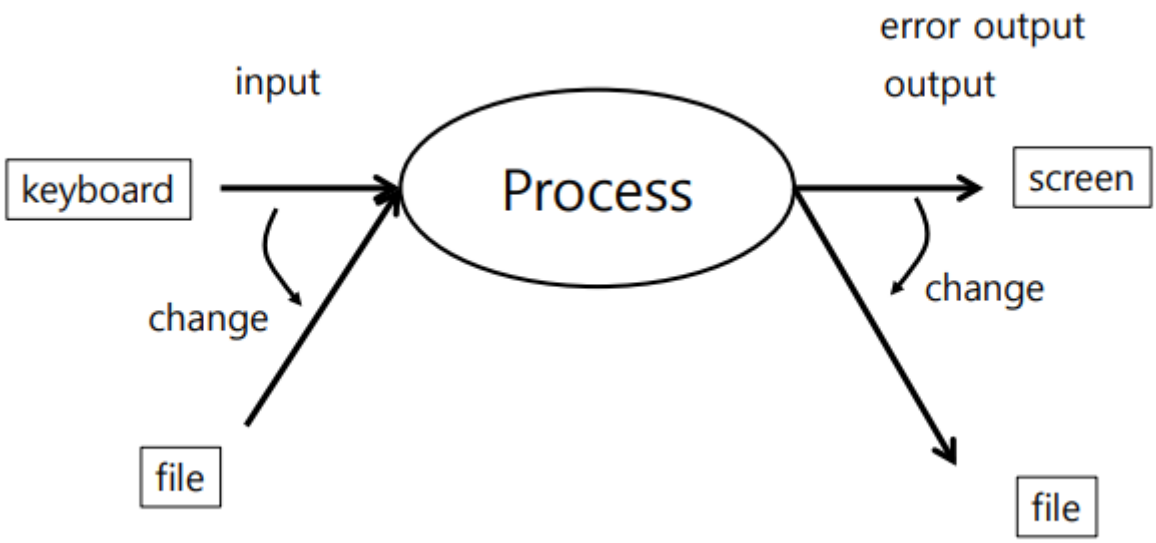
```
$ ls
$ ls -sCF #size multi-column indicates if it's file or folder
$ ls -s -C -F /usr/bin # same as above for another directory
$ ls -a # lists all files (hidden files as well)
$ ls . # current directory
$ ls -al # shows a list with permissions and other infos
$ ls -ld # lists directories only
```

Example Output:

```
21170004@comunix:~$ ls -al
total 173
drwxr-xr-x   5 21170004 studs         36 Apr 14 00:27 .
drwxr-xr-x 153 root     root        153 Mar 10 18:24 ..
-rw-------   1 21170004 studs       5743 Apr 13 17:28 .bash_history
-r--r--r--   1 21170004 studs        159 Mar 10 18:24 .bashrc
-rw-------   1 21170004 studs         35 Apr 14 00:27 .lesshst
-rw-r--r--   1 21170004 studs        568 Mar 10 18:24 .profile
-rw-------   1 21170004 studs         28 Mar 30 10:14 .sh_history
-rw-r--r--   1 21170004 studs          0 Mar 23 00:25 .txt
-rwxr-xr-x   1 21170004 studs       7416 Apr 13 14:58 a.out
-rw-r--r--   1 21170004 studs          0 Mar 23 00:25 aa
```

I/O Redirection

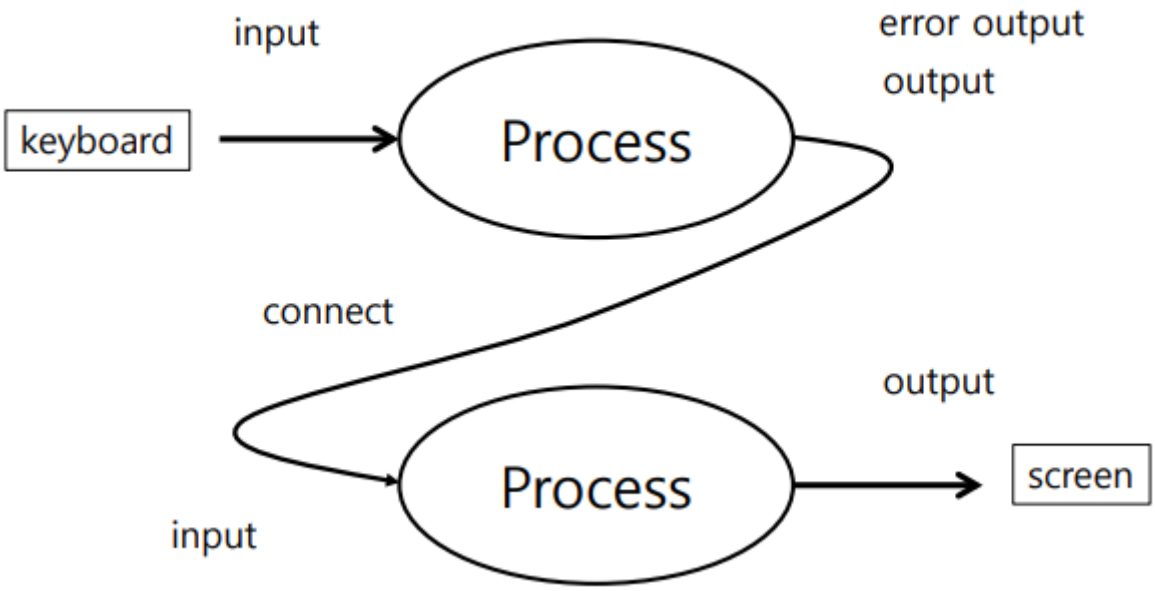
A process is a running program that consists of an input, an output and an error output. With I/O Redirection it is possible to change the input and output of a program.




```
$ 1> #change output
$ 2> #change error output
$ < #change input
$ echo hello > hi.txt # example
$ cat filedoesntexist.txt 2> error.txt # example2
```

Pipe

With a pipe it is possible to connect the output or the error output to another process:



echo

```
$ echo hi~ #prints hi~ on the console
$ echo hi > hi.txt #writes 'hi' into the file hi.txt
$ echo hello >> hi.txt #appends hello to the file hi.txt
$ echo hello | write unix300 # shows hello on the terminal of user unix300
$ echo hello | write unix300 pts/7 # same as above but specifying  the terminal
$ echo $PATH #prints the PATH-Variable
$ banner Hi~ # prints a big Hi~ on the terminal using ASCI-Characters
```

wc

Word-Count command.

```
$ cat hi.txt # shows content of hi.txt
$ cat /etc/passwd | wc # shows size, rows and word count of file hi.txt
$ wc /etc/passwd # Same as above without using a pipe
$ wc -l /etc/passwd # displays amount of lines
```

cp, mv, rm

Copy, Move, Remove

```
$ cp hi.txt hi2.txt # Copies file hi.txt with its contents to file hi2.txt
$ rm hi2.txt # removes FIle hi2.txt
$ mv hi.txt hi3.txt # Moves contents from file hi.txt to new file hi3.txt and
deletes file hi.txt
```

directory

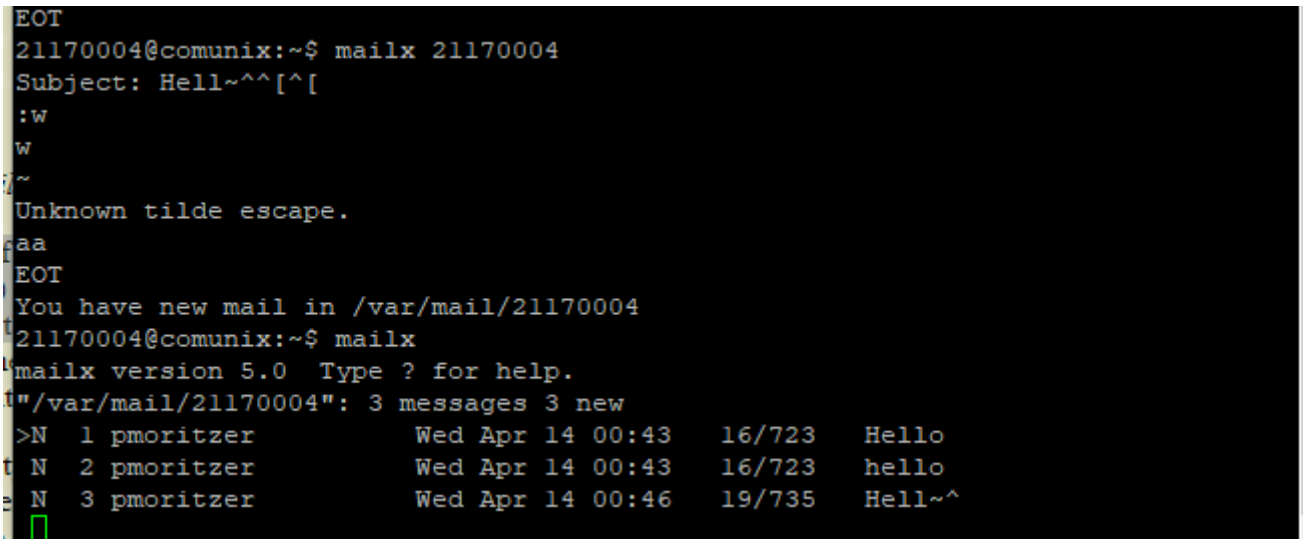
Important Directory commands

```
$ pwd # prints current working directory
$ cd .. # change working directory to upper folder
$ cd / # change working directory to specified path
$ mkdir dir1 # creates new directory with name dir1
$ rmdir dir1 # removes dir1
```

mail

```
$ mailx # Prompts mail Program, if no mails just a message, otherwise the user
will be asked for a mail
$ mailx unix300 #sends mail to user unix300
$ mailx unix300 < hi.txt # sends mail with content of file hi.txt
$ ls /var/mail # folder containing mails for users
```

Output:



grep

Grep stand for global regular expression and is used for filtering Files or Text by using regular expressions.

```
$ cat /etc/passwd | grep unix30 # only shows lines of files that include 'unix30'
$ grep unix30 /etc/passwd # same as above in /etc/passwd file without piping
```

```
$ ls -l | grep hi # lists all files that contain hi in their name
```

Viewing Text Files

```
$ cat /etc/passwd # Shows content of /etc/passwd file
$ cat /etc/passwd | more # Only shows a few lines and the file is scrollable by using the space bar
$ more /etc/passwd # same as above without piping
$ less /etc/passwd # similar to more but allows forward and backward navigation
$ pg /etc/group # Pagination for passwd file (allows the user to enter a page)
$ head -5 /etc/group # shows the first 5 lines of the file
$ tail /etc/group #prints the last lines of the file
```

Redirection

```
$ echo hi > hi.txt # saves the output 'hi' of echo into hi.txt file
$ mailx unix300 < hi.txt # sends mail to user unix300 with the contents of hi.txt
$ mkdir1
+ $mkdir dir1 2> err.txt # Prints error of folder already existing to err.txt
$ find /export/home -name 'hi.xt' 2> err.txt 1>out.txt # prints the error (permission denied e.g.) to err.txt and the finding results for 'hi.txt' into out.txt
```