

Basic Linux commands

pwd: تستخدم لمعرفة المسار (الذي يمثل المجلد) الموجود فيه حاليا

 ahed@DESKTOP-OK5G6FV: ~

```
ahed@DESKTOP-OK5G6FV:~$ pwd  
/home/ahed  
ahed@DESKTOP-OK5G6FV:~$
```

لعرض الملفات والمجلدات الموجود في المسار او المجلد الحالي :ls

```
ahed@DESKTOP-OK5G6FV: ~  
ahed@DESKTOP-OK5G6FV:~$  
ahed@DESKTOP-OK5G6FV:~$ ls  
Ahmadabusaleem_1201315.sh:Zone.Identifier  fifo_msgq  s  tc  
Diananaseer_1210363.sh:Zone.Identifier    fork_codes sari_1220982.sh:Zone.Identifier  testc  
E                                             fp          sc.c          testc.c  
ENCS3130_Projects                        grep        shc           testd  
ENCS3130_Projects1                      greptest   shmc.c        testd.c  
ENCS3130_Projects2                     head        shmc.c.save   teste  
Exp8_ToDo                               headtest   shmcl          thread  
UNIX                                     ls          shms.c        todo#2  
YousefQawwas-1220137.sh:Zone.Identifier  lspci      shmsem        todo6  
a.out                                   lstest     shmsem.c      touch  
ahed                                   main.sh:Zone.Identifier  shmsr        touchtest  
ammancom                               mkdir      shs           tr  
asmaa_1210084.sh:Zone.Identifier         mkf        soc_c.c       trtest  
c                                         mytest     soc_s.c       ts  
c_expr                                  ob          sort          uc  
cat                                     ob1.c      sorttest     udp_c.c  
cattest                               ob1.o      ss.c          udp_s.c  
chmod                                  ob2.c      ss.c.save    us  
cp                                      ob2.o      t7.c         wc  
cptest                                optimize   tail          wctest  
cut                                   project1   tailtest     who  
cuttest                             project1:Zone.Identifier  task.c       whotest  
date                                  project1_part2  task2.sh:Zone.Identifier  
datetest                             project1_part2:Zone.Identifier  task6  
ahed@DESKTOP-OK5G6FV:~$
```

- **ls folder_name:** لعرض الملفات والمجلدات الموجودة في مجلد او مسار معين

```
ahed@DESKTOP-OK5G6FV:~$  
ahed@DESKTOP-OK5G6FV:~$ ls ahed  
my_folder  test.txt  
ahed@DESKTOP-OK5G6FV:~$
```

- All commands must be written in small letters => PWD (**Wrong**)
=> pwd (**Correct**)
=> LS (**Wrong**)
=> ls (**Correct**)

- **mkdir dir(folder)_name:** لإنشاء مجلد أو مسار

```
ahed@DESKTOP-OK5G6FV:~$  
ahed@DESKTOP-OK5G6FV:~$ mkdir project  
ahed@DESKTOP-OK5G6FV:~$
```

سؤال: كيف نتأكد من إنشاء المجلد الموضح بالصورة ؟

- **rmdir dir_name:** تستخدم لحذف المسار أو المجلد الفارغ
- We can create multiple directories in the same line as the following:
 mkdir project1 project2 "test me" => create 3 directories (project1, project2 and "test me")
* Notice the double quotation with test me directory because there is a space in the name of the directory.

► cd : لتغيير المسار او المجلد الموجود فيه ويوجد منها 4 اشكال

1- `cd dir_name`: للذهاب الى مسار معين

```
ahed@DESKTOP-OK5G6FV: ~/ahed
```

```
ahed@DESKTOP-OK5G6FV:~$  
ahed@DESKTOP-OK5G6FV:~$  
ahed@DESKTOP-OK5G6FV:~$ cd ahed  
ahed@DESKTOP-OK5G6FV:~/ahed$
```

2- `cd`: للذهاب الى المسار الرئيسي المسمى بـ `home`

```
ahed@DESKTOP-OK5G6FV:~/ahed/my_folder$ pwd  
/home/ahed/ahed/my_folder  
ahed@DESKTOP-OK5G6FV:~/ahed/my_folder$ cd  
ahed@DESKTOP-OK5G6FV:~$ pwd  
/home/ahed  
ahed@DESKTOP-OK5G6FV:~$
```

How can `cd` command detect the home directory? You will know in environment variable part

- تستخدم للذهاب الى المجلد الحالي او الذي انا موجود فيه فهي تدل على المكان الموجود فيه الان : **cd .** 3-
- تستخدم للرجوع الى المجلد السابق (الأب للمجلد الحالي) : **cd ..** 4-

```
ahed@DESKTOP-OK5G6FV:~/mytest/test1$  
ahed@DESKTOP-OK5G6FV:~/mytest/test1$ pwd  
/home/ahed/mytest/test1  
ahed@DESKTOP-OK5G6FV:~/mytest/test1$ cd .  
ahed@DESKTOP-OK5G6FV:~/mytest/test1$ pwd  
/home/ahed/mytest/test1  
ahed@DESKTOP-OK5G6FV:~/mytest/test1$
```

```
ahed@DESKTOP-OK5G6FV:~/mytest/test1$  
ahed@DESKTOP-OK5G6FV:~/mytest/test1$ pwd  
/home/ahed/mytest/test1  
ahed@DESKTOP-OK5G6FV:~/mytest/test1$ cd ..  
ahed@DESKTOP-OK5G6FV:~/mytest$ pwd  
/home/ahed/mytest  
ahed@DESKTOP-OK5G6FV:~/mytest$
```

Relative vs Absolute Path

► Relative: ينتقل بناء على المكان الذي يكون عنده

► Example

```
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ pwd
/home/ahed/ahed/main_dir
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ cd ..
ahed@DESKTOP-OK5G6FV:~/ahed$ pwd
/home/ahed/ahed
ahed@DESKTOP-OK5G6FV:~/ahed$
```

نلاحظ في الصورة انه انتقل خطوة للوراء بناء على المكان الذي كان فيه
لا تأخذني دائما الى نفس المسار `cd ..` اذا لانها تعتمد على مكاني الان

► Absolute: دائما يذهب الى المسار المحدد بغض النظر عن مكانك الان

```
ahed@DESKTOP-OK5G6FV:~/testdir$ pwd
/home/ahed/testdir
ahed@DESKTOP-OK5G6FV:~/testdir$ cd /home/ahed/mkf
ahed@DESKTOP-OK5G6FV:~/mkf$ pwd
/home/ahed/mkf
ahed@DESKTOP-OK5G6FV:~/mkf$
```

نلاحظ هنا انه انتقل الى المسار الموجود وهذا لا يعتمد على مكان وجوده الحالي لانه حدد المسار

Copy, cut and rename the file

- ▶ cp source destination: نسخ الفايل من السورس الى الديرستنيشن

```
ahed@DESKTOP-OK5G6FV:~/ahed$ cp ./test.txt main_dir/  
ahed@DESKTOP-OK5G6FV:~/ahed$ ls main_dir/  
file.txt hard msg.txt soft test.txt  
ahed@DESKTOP-OK5G6FV:~/ahed$
```

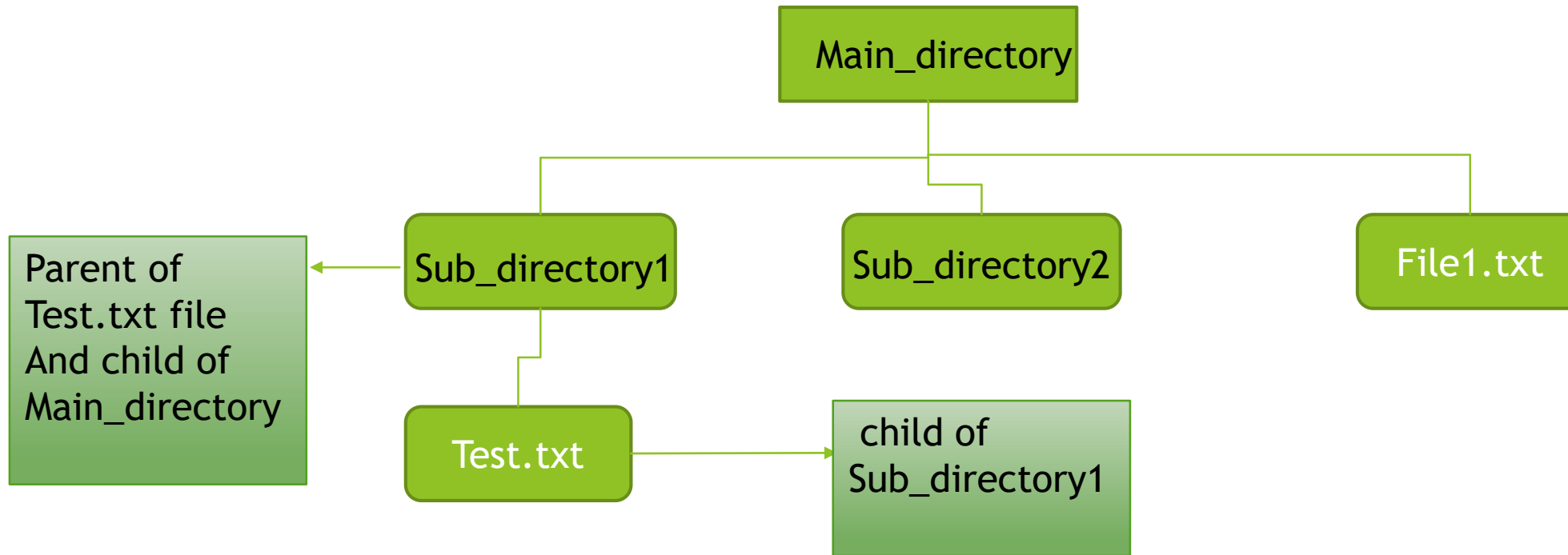
- ▶ mv src dest: cut تستخدم لاعادة التسمية و عمل
- ▶ If the src and dest was in the same directory, mv behave as **rename**
- ▶ Otherwise, mv behave as **cut**

من الافضل الاشارة الى الملف الموجود في المسار الحالي باستخدام **./file_name** وهذا يستخدم بكثرة في السيرفرات

```
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ mv ./test.txt ./new.txt  
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ls  
file.txt hard msg.txt new.txt soft  
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$
```

```
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ mv ./new.txt ../  
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ls  
file.txt hard msg.txt soft  
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ls ../  
exp3.txt file.txt fruit.txt hard main_dir msg.txt new.txt soft  
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$
```


►: Linux build the files and folders as tree :



- ▶ `touch file_name:` **تستخدم لانشاء ملف وليس مجلد**

```
ahed@DESKTOP-OK5G6FV:~/mytest$ touch file_test.txt
ahed@DESKTOP-OK5G6FV:~/mytest$ ls
file_test.txt  myhead.h  t  test1  testh.c
ahed@DESKTOP-OK5G6FV:~/mytest$
```

- ▶ Don't use `cd` command with **files**, just use it with **directories (folders)**.
- ▶ **نظام اللينوكس يعتبر كل شيء انه ملف حتى الاجهزة المتصلة به ك الماوس او الكيبورد يعتبرهم ملفات او وايضا المجلدات يعتبرهم ملفات**
- ▶ **عشان هيك ضروري نعرف كيف نتعامل مع الملفات**

files

- ▶ **touch file_name**: تستخدم لإنشاء ملف
- ▶ للدخول إلى الملفات والكتابة عليها هناك 3 طرق
 - 1- graphical user interface (GUI)
 - 2- Terminal using text editors such as **nano** (الأكثر استخدامًا), pico, vietc.

```
ahed@DESKTOP-OK5G6FV: ~/mytest
ahed@DESKTOP-OK5G6FV:~/mytest$
ahed@DESKTOP-OK5G6FV:~/mytest$ nano file_test.txt
```

- ▶ To save the content after access the file using **nano** press **ctrl+s**, then to exit press **ctrl+x**
- 3- redirection : >, >> في هذه الطريقة يتم إعادة توجيه الاوتبوت الى الملف بدلا من التيرمينال باستخدام

في المثال التالي يتم عرض الاوتبوت للكوماند على التيرمينال

```
ahed@DESKTOP-OK5G6FV:~/mytest$  
ahed@DESKTOP-OK5G6FV:~/mytest$ ls  
file_test.txt myhead.h t test1 testh.c  
ahed@DESKTOP-OK5G6FV:~/mytest$
```

> بينما في هذا المثال يتم توجيه الاوتبوت للكوماند على الملف باستخدام

```
ahed@DESKTOP-OK5G6FV:~/mytest$ ls > file_test.txt  
ahed@DESKTOP-OK5G6FV:~/mytest$
```

- ▶ > used for over write, >> used for append
- ▶ We use **cat** (🐈) command to **print** the **file content**

```
ahed@DESKTOP-OK5G6FV:~/mytest$ cat file_test.txt  
file_test.txt  
myhead.h  
t  
test1  
testh.c  
ahed@DESKTOP-OK5G6FV:~/mytest$
```

- ▶ There is other commands which is used for printing the file content such as more , less, head and tail .
- ▶ **More, less**: used with the large file content => more/less divide the large content into pages.
- ▶ **head, tail** : used to print the first 10 lines (head) or the last 10 lines (tail).

Please try them with your self 😊

- ▶ **rm file_name** : تستخدم لحذف الملف او الفايل
- ▶ **echo command**: يستخدم لطباعة سترنج على التيرمينال فهو يري كل شيء ك سترنج

```
ahed@DESKTOP-OK5G6FV:~$ echo "ahed mafarjeh"
ahed mafarjeh
ahed@DESKTOP-OK5G6FV:~$ echo "1+2+3+4"
1+2+3+4
ahed@DESKTOP-OK5G6FV:~$
```

Aliasing

- ▶ **alias**: نستخدمها لتبديل شيء بشيء آخر
- ▶ اذا كان عندي مثلا كوماندا طویل ممكن ابدله ب كوماندا اقصر بس بنفس الوظيفة
- ▶ Sytnax: alias var_name="coomand"
- ▶ لا يوجد فراغ قبل و بعد ال =
- ▶ Var_name will work as command

Example:

1)

```
ahed@DESKTOP-OK5G6FV:~$ alias ahed="echo ahed mafajreh"
ahed@DESKTOP-OK5G6FV:~$ ahed
ahed mafajreh
ahed@DESKTOP-OK5G6FV:~$
```

Aliasing

2)

```
ahed@DESKTOP-OK5G6FV:~$ alias ls="echo hello"
ahed@DESKTOP-OK5G6FV:~$ ls
hello
ahed@DESKTOP-OK5G6FV:~$
```

كما تلاحظ ان ls اصبحت تعمل كما انها echo hello اي انك تطبع كلمة hello
لنرجع ls الى وظيفتها القديمة نعمل unalias كما في الصورة ادناه

```
ahed@DESKTOP-OK5G6FV:~$ unalias ls
ahed@DESKTOP-OK5G6FV:~$ ls
Ahmadabusaleem_1201315.sh:Zone.Identifier  ls                sort
Diananaseer_1210363.sh:Zone.Identifier  lspci             sorttest
E                                           lstest            ss.c
ENCS3130_Projects                        main.sh:Zone.Identifier  ss.c.save
ENCS3130_Projects1                      mkdir             t7.c
ENCS3130_Projects2                      mkf               tail
Exp8_ToDo                               mytest            tailtest
UNIX                                     ob                task.c
YousefQawwas-1220137.sh:Zone.Identifier  ob1.c             task2.sh:Zone.Identifier
a.out                                    ob1.o             task6
ahed                                    ob2.c             tc
'ahed mafarjeh'                          ob2.o             testc
ammancom                                optimize          testc.c
asmaa_1210084.sh:Zone.Identifier          project           testd
c                                           project1          testd.c
c_expr                                  project1:Zone.Identifier  teste
cat                                       project1_part2      thread
cattest                                  project1_part2:Zone.Identifier  todo#2
chmod                                     s                 todo6
cp                                       sari_1220982.sh:Zone.Identifier  touch
```

Command option

معظم الكوماندز يوجد لها options وهذه options تمكني من جعل الكوماند يعمل بكفاءة او يعمل اضافة الى هذا الكوماند

مثلا كوماند ls يطبع الملفات والمجلدات لكن اذا استخدمنا معه option l سوف يطبع الملفات والمجلدات مع تفاصيل عنها كمثلا وقت انشاؤها وكم حجمها وهل هي ملف او مجلد...الخ

- Syntax: command -option
- Examples:

```
ahed@DESKTOP-OK5G6FV:~/ahed$ ls
exp3.txt file.txt fruit.txt test.txt
ahed@DESKTOP-OK5G6FV:~/ahed$ ls -l
total 0
-rw-r--r-- 1 ahed ahed 269 Jul 21 14:56 exp3.txt
-rw-r--r-- 1 ahed ahed 252 Jul 21 15:45 file.txt
-rw-r--r-- 1 ahed ahed 49 Jul 21 15:29 fruit.txt
-rw-r--r-- 1 ahed ahed 19 Jul 21 12:45 test.txt
ahed@DESKTOP-OK5G6FV:~/ahed$
```

```
ahed@DESKTOP-OK5G6FV:~/ahed$ cat file.txt
The Unix operating system was pioneered by Ken Unix
Thompson and Dennis Ritchie at Bell Laboratories
in the late 1960s. One of the primary goals in
the design of the Unix system was to create an
environment that promoted efficient program development.
ahed@DESKTOP-OK5G6FV:~/ahed$ cat -n file.txt
1 The Unix operating system was pioneered by Ken Unix
2 Thompson and Dennis Ritchie at Bell Laboratories
3 in the late 1960s. One of the primary goals in
4 the design of the Unix system was to create an
5 environment that promoted efficient program development.
ahed@DESKTOP-OK5G6FV:~/ahed$
```


Command option examples

► Examples :

```
ahed@DESKTOP-OK5G6FV:~/ahed$ head -2 file.txt
The Unix operating system was pioneered by Ken Unix
Thompson and Dennis Ritchie at Bell Laboratories
ahed@DESKTOP-OK5G6FV:~/ahed$ tail -2 file.txt
the design of the Unix system was to create an
environment that promoted efficient program development.
ahed@DESKTOP-OK5G6FV:~/ahed$
```

head print the **first 2 line** and tail print the **last 2 line** instead of its default behavior which is print 10 line

```
ahed@DESKTOP-OK5G6FV:~/ahed$ ls -a
.  ..  exp3.txt  file.txt  fruit.txt  test.txt
ahed@DESKTOP-OK5G6FV:~/ahed$
```

Ls -a print all files include **hidden** files

```
ahed@DESKTOP-OK5G6FV:~/ahed$ echo -n "ahed mafarjeh"
ahed mafarjehahed@DESKTOP-OK5G6FV:~/ahed$
```

echo -n print the string **without new line**

```
ahed@DESKTOP-OK5G6FV:~/ahed$ echo "ahed mafarjeh"
ahed mafarjeh
ahed@DESKTOP-OK5G6FV:~/ahed$ echo -e "ahed \nmafarjeh"
ahed
mafarjeh
ahed@DESKTOP-OK5G6FV:~/ahed$
```

echo -e **detect new line character \n or tab \t**

Command options

► Examples:

```
ahed@DESKTOP-OK5G6FV:~/ahed$ rmdir test_dir
rmdir: failed to remove 'test_dir': Directory not empty
ahed@DESKTOP-OK5G6FV:~/ahed$ rm -r test_dir/
ahed@DESKTOP-OK5G6FV:~/ahed$
```

rm -r : delete the not empty directory

► نستطيع ان نستخدم man كوماندا لعرض تفاصيل عن الكوماندا الاخرى ومعرفة كل options لها

► Syntax: man command_name, to exit from man page prees “q”

```
ahed@DESKTOP-OK5G6FV:~/ahed$ man ls
```

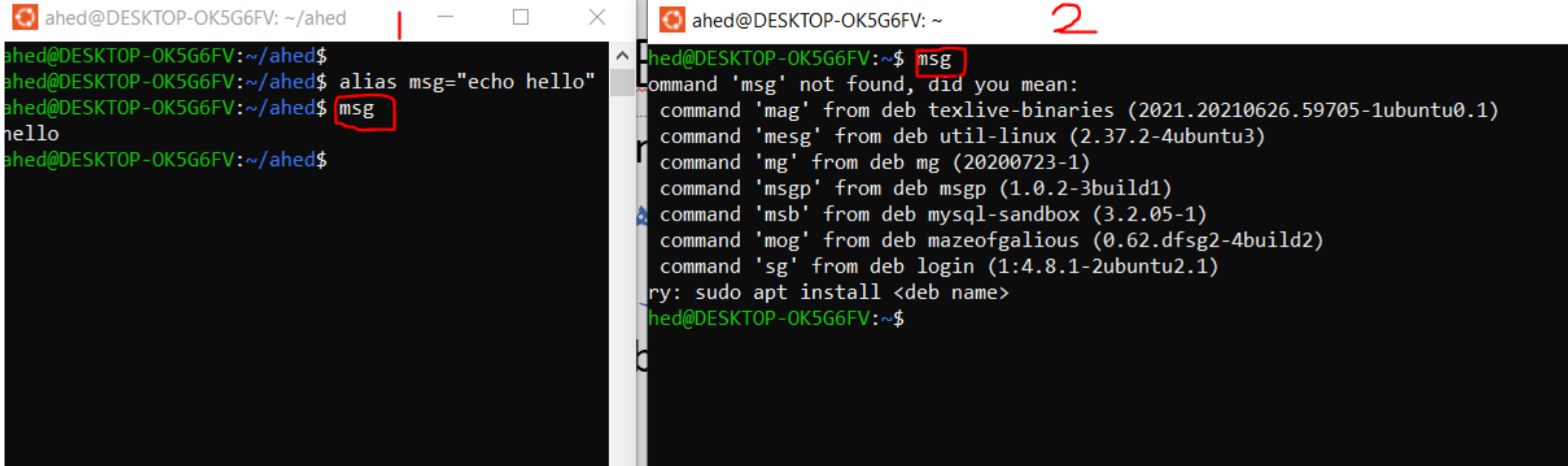
```
LS(1) User Commands LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of
  -cftuvSUX nor --sort is specified.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --all
    do not ignore entries starting with .
  -A, --almost-all
    do not list implied . and ..
  --author
    with -l, print the author of each file
  -b, --escape
    print C-style escapes for nongraphic characters
  --block-size=SIZE
    with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
Manual page ls(1) line 1 (press h for help or q to quit)
```

Enviroment Variable

► There are 2 type of environment variable:

1. Global : variables كل السستم يستطيع رؤية هذه المتغيرات
2. Local : فقط المكان المعرف فيه هذا المتغير يستطيع رؤيته

Example on local variable:



The image shows two terminal windows side-by-side. The left window shows the creation of a local alias and its successful execution. The right window shows an attempt to use the alias from a different directory, which fails because the shell's PATH does not include the current directory.

```
ahed@DESKTOP-OK5G6FV: ~/ahed | ahed@DESKTOP-OK5G6FV: ~ 2
ahed@DESKTOP-OK5G6FV:~/ahed$
ahed@DESKTOP-OK5G6FV:~/ahed$ alias msg="echo hello"
ahed@DESKTOP-OK5G6FV:~/ahed$ msg
hello
ahed@DESKTOP-OK5G6FV:~/ahed$

ahed@DESKTOP-OK5G6FV:~$ msg
command 'msg' not found, did you mean:
command 'mag' from deb texlive-binaries (2021.20210626.59705-1ubuntu0.1)
command 'mesg' from deb util-linux (2.37.2-4ubuntu3)
command 'mg' from deb mg (20200723-1)
command 'msgp' from deb msgp (1.0.2-3build1)
command 'msb' from deb mysql-sandbox (3.2.05-1)
command 'mog' from deb mazeofgalious (0.62.dfsg2-4build2)
command 'sg' from deb login (1:4.8.1-2ubuntu2.1)
ry: sudo apt install <deb name>
ahed@DESKTOP-OK5G6FV:~$
```

Enviroment variables

- في المثال السابق قمنا بتعريف متغير msg في تيرمينال 1 يقوم بطباعة رسالة hello من خلال alias ، لكن التيرمينال 2 لم تتعرف على المتغير msg لانه متعرف فقط بشكل local في تيرمينال 1
- لنعمل global var يجب ان نعرفه في ملف اسمه .bashrc وهو ملف مخفي موجود في home

```
ahed@DESKTOP-OK5G6FV:~$ ls -a
.
..
.Xauthority
.bash_history
.bash_logout
.bashrc
.cache
.config
.lesshst
.local
.motd_shown
.profile
.python_history
.sudo_as_admin_successful
Ahmadabusaleem_1201315.sh:Zone.Identifier
Diananaseer_1210363.sh:Zone.Identifier
E
FMS3170_Projects
date
datetest
fifo_msgq
fork_codes
fp
grep
greptest
head
headtest
ls
lspci
lstest
main.sh:Zone.Identifier
mkdir
mkf
mytest
ob
ob1.c
shmsr
shs
soc_c.c
soc_s.c
sort
sorttest
ss.c
ss.c.save
t7.c
tail
tailtest
task.c
task2.sh:Zone.Identifier
task6
tc
testc
testc.c
testd
```

Enviroment variables

لندخل الى الملف نستخدم nano ثم ننتقل الى اخر الملف ونضيف variable ثم نحفظ الملف ونغلق التيرمينال لبتنم اخذ التعديلا في ملف الاعدادات (.bashrc)

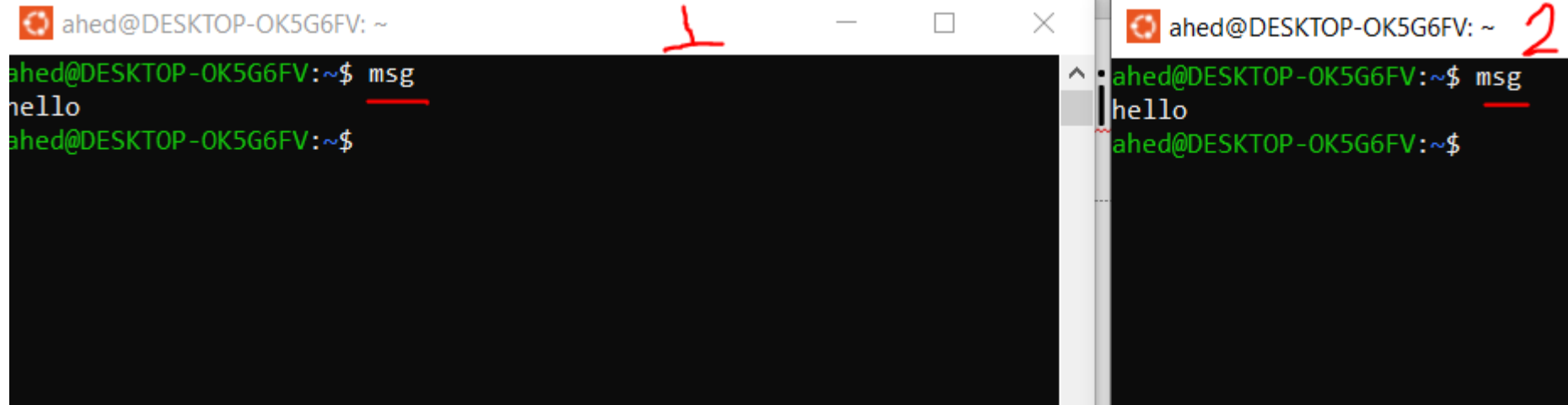
```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

alias msg="echo hello"
```

Enviroment variables

► Result



The image shows two terminal windows side-by-side. Both windows have a title bar that reads 'ahed@DESKTOP-OK5G6FV: ~'. The left window has a red '1' above it, and the right window has a red '2' above it. In both windows, the command 'msg' has been entered and executed, resulting in the output 'hello'.

```
ahed@DESKTOP-OK5G6FV: ~$ msg
hello
ahed@DESKTOP-OK5G6FV: ~$
```

DISPLAY
EDITOR
HOME
HOST
LOGNAME
PATH
SHELL
TERM
USER

global هذه المتغيرات محجوزة في النظام وتكون

File Permission الصلاحيات

لنعرض permission نحتاج الى كومانند ls -l الذي يعرض التفاصيل عن الملفات والمجلدات :

The **first char** represent the file type:
Dash (-) → ordinary file (.txt, img , video etc)
(d) → Directory
(l) → Link (short cut in windows)

```
ahed@DESKTOP-OK5G6FV:~/ahed$ ls -l
total 0
-rw-r--r-- 1 ahed ahed 269 Jul 21 14:56 exp3.txt
-rw-r--r-- 1 ahed ahed 252 Jul 21 15:45 file.txt
-rw-r--r-- 1 ahed ahed 49 Jul 21 15:29 fruit.txt
drwxr-xr-x 1 ahed ahed 4096 Jul 21 18:05 main_dir
-rw-r--r-- 1 ahed ahed 19 Jul 21 12:45 test.txt
ahed@DESKTOP-OK5G6FV:~/ahed$
```

Name of
the files

Permission
r(read), w(write),

Owner user

size

Date and
time
creation

Permissions are
divide into 3
groups:
1- owner
2- group
3- other

File Permission الصلاحيات

- يكون ترتيب permission هكذا rwx ضروري هاد الترتيب لا يتغير
- مثلا اذا بدنا نعطي write permission نعطي قيمة 1 لها والباقي 0 كالتالي



- اذا القيمة 2 تعبر عن write permission
- سؤال : ما القيمة التي تعبر عن r-x و r-- و rw-
- لتغير صلاحية الملف نستخدم chmod command كالتالي : chmod permission file_name
- وهناك 3 صلاحيات : owner , group , other اذا عملنا هكذا **chmod 741 t.txt**
- فاننا نعطي owner صلاحية rwx التي يمثلها رقم 7، و group بوخذ صلاحية r-- الذي يمثلها رقم 4 و other بوخذ -x والتي يمثلها رقم 1

▶ ناخذ الملف الاول كمثال

```
ahed@DESKTOP-OK5G6FV:~/ahed$ ls -l
total 0
-rw-r--r-- 1 ahed ahed 269 Jul 21 14:56 exp3.txt
-rw-r--r-- 1 ahed ahed 252 Jul 21 15:45 file.txt
-rw-r--r-- 1 ahed ahed 49 Jul 21 15:29 fruit.txt
drwxr-xr-x 1 ahed ahed 4096 Jul 21 18:05 main_dir
-rw-r--r-- 1 ahed ahed 19 Jul 21 12:45 test.txt
ahed@DESKTOP-OK5G6FV:~/ahed$ chmod 741 exp3.txt
ahed@DESKTOP-OK5G6FV:~/ahed$ ls -l
total 0
-rwxr---x 1 ahed ahed 269 Jul 21 14:56 exp3.txt
-rw-r--r-- 1 ahed ahed 252 Jul 21 15:45 file.txt
-rw-r--r-- 1 ahed ahed 49 Jul 21 15:29 fruit.txt
drwxr-xr-x 1 ahed ahed 4096 Jul 21 18:05 main_dir
-rw-r--r-- 1 ahed ahed 19 Jul 21 12:45 test.txt
```

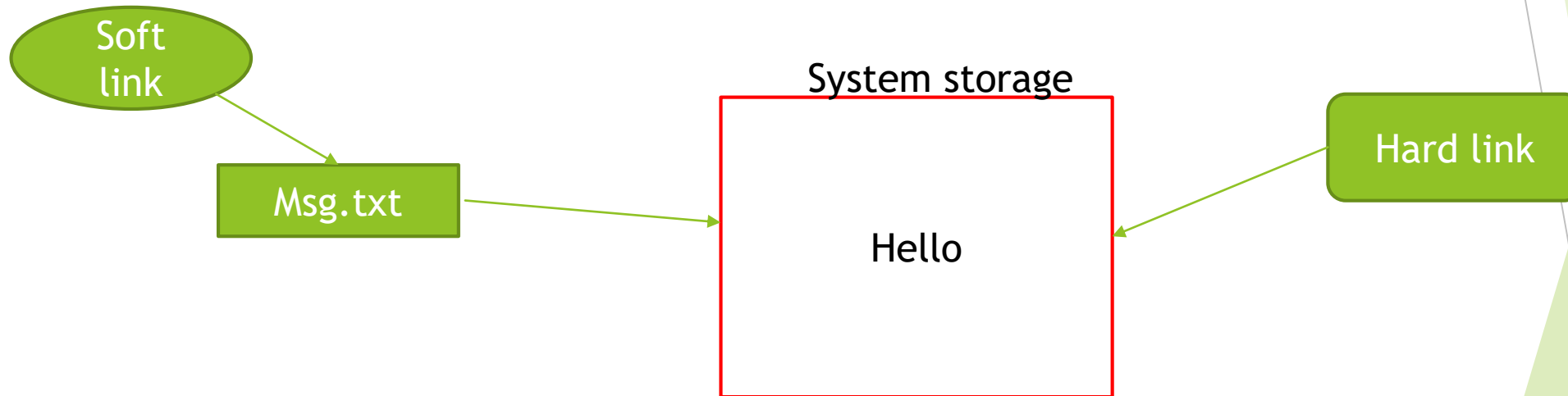
- ▶ Q) change the permission of file.txt into rwx to the owner and -w- to the group and other.

Link

تشبه short cut في windows

من خلالها نستطيع عمل مؤشر pointer على الفايل ويوجد منها نوعين : soft, hard

الفرق بينهما توضحها الرسمة التالية



يوجد ملف اسمه Msg.txt يحتوي على كلمة hello، ويقوم soft link بالتشير على الملف الاصلي بينما يقوم hard link بالتشير على content

Link

- ▶ To create the soft and hard link use the following command:
- ▶ `ln msg.txt hard` → for hard link
- ▶ `ln -s msg.txt soft` → for soft link

```
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ln msg.txt hard
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ln -s msg.txt soft
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ls -l
total 0
-rw-r--r-- 2 ahed ahed 6 Jul 21 18:53 hard
-rw-r--r-- 2 ahed ahed 6 Jul 21 18:53 msg.txt
lrwxrwxrwx 1 ahed ahed 7 Jul 21 18:54 soft -> msg.txt
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$
```

- ▶ Q1) print the value of msg.txt
- ▶ Q2) print the value of soft and hard
- ▶ Q3) edit the msg.txt file
- ▶ Q4) print the content of soft and hard
- ▶ Q5) delete msg.txt and print the content of soft and hard
- ▶ What do you notice

Error redirection

- ▶ To redirect the error value to the file content we use 2>
- ▶ If you type `ls zzzzzz > file.txt` , will print error because the zzzzzz directory not found

```
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ls zzzzzzzz > file.txt
ls: cannot access 'zzzzzzzz': No such file or directory
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$
```

- ▶ But if you want to redirect the output to the file regardless if there are errors then you must use 2> instead of > or >>

```
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ ls zzzzzzzz 2> file.txt
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$ cat file.txt
ls: cannot access 'zzzzzzzz': No such file or directory
ahed@DESKTOP-OK5G6FV:~/ahed/main_dir$
```

Created By:

Ahed Mafarjeh

Good Luck 😊