

INFO 2603 – Lab 5

Linux Terminal

Prepared by Naresh Seegobin
naresh.seegobin@sta.uwi.edu

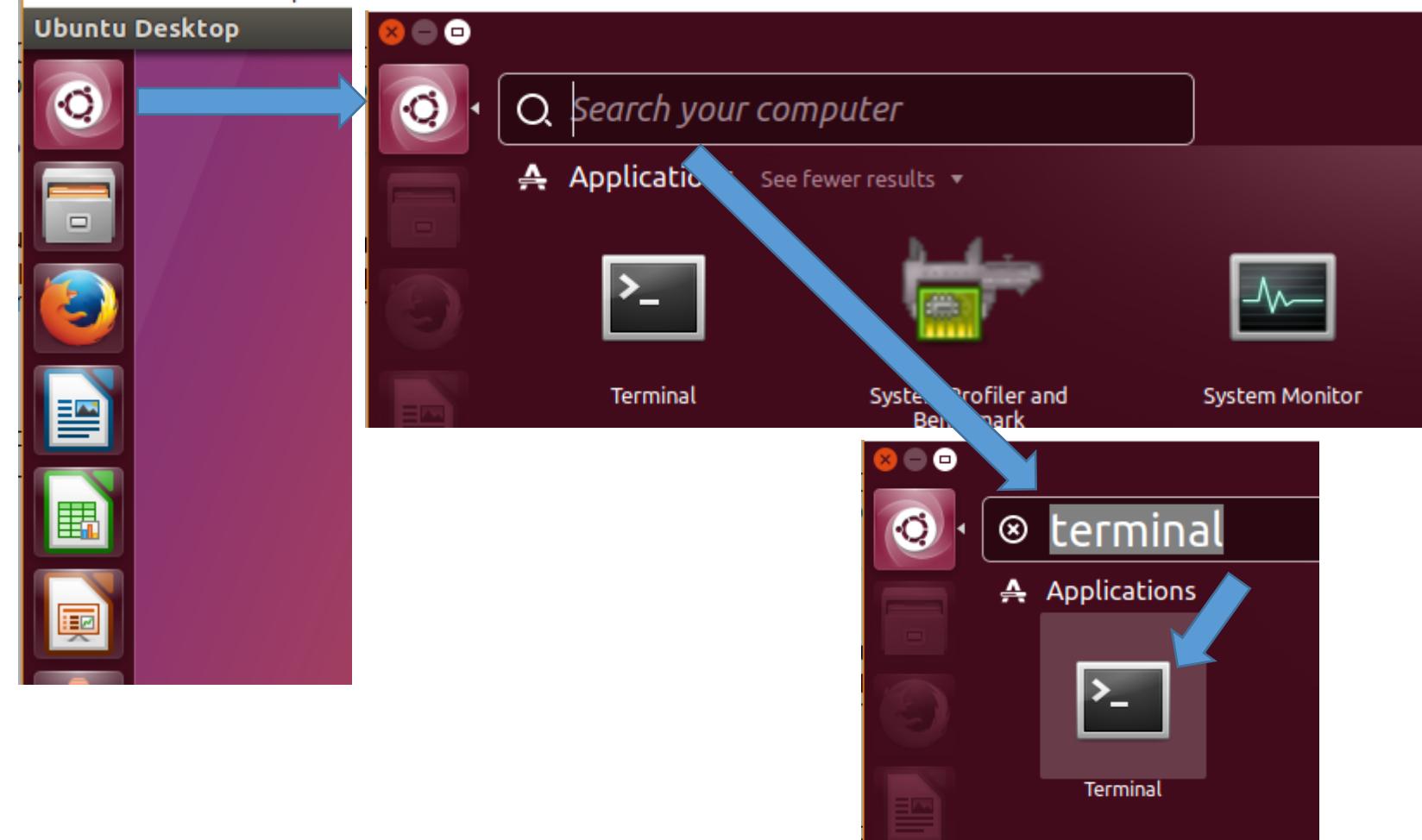
Outline - Linux Terminal Introduction

- Navigating directories
 - bash help
 - cd, ls (dir)
 - Relative and absolute paths
 - mkdir, rm, mv
- Linux Directories
 - /
 - /boot
 - /home
 - /etc
 - /bin
 - /sbin
 - /usr
 - /var

Outline - Linux Terminal Introduction

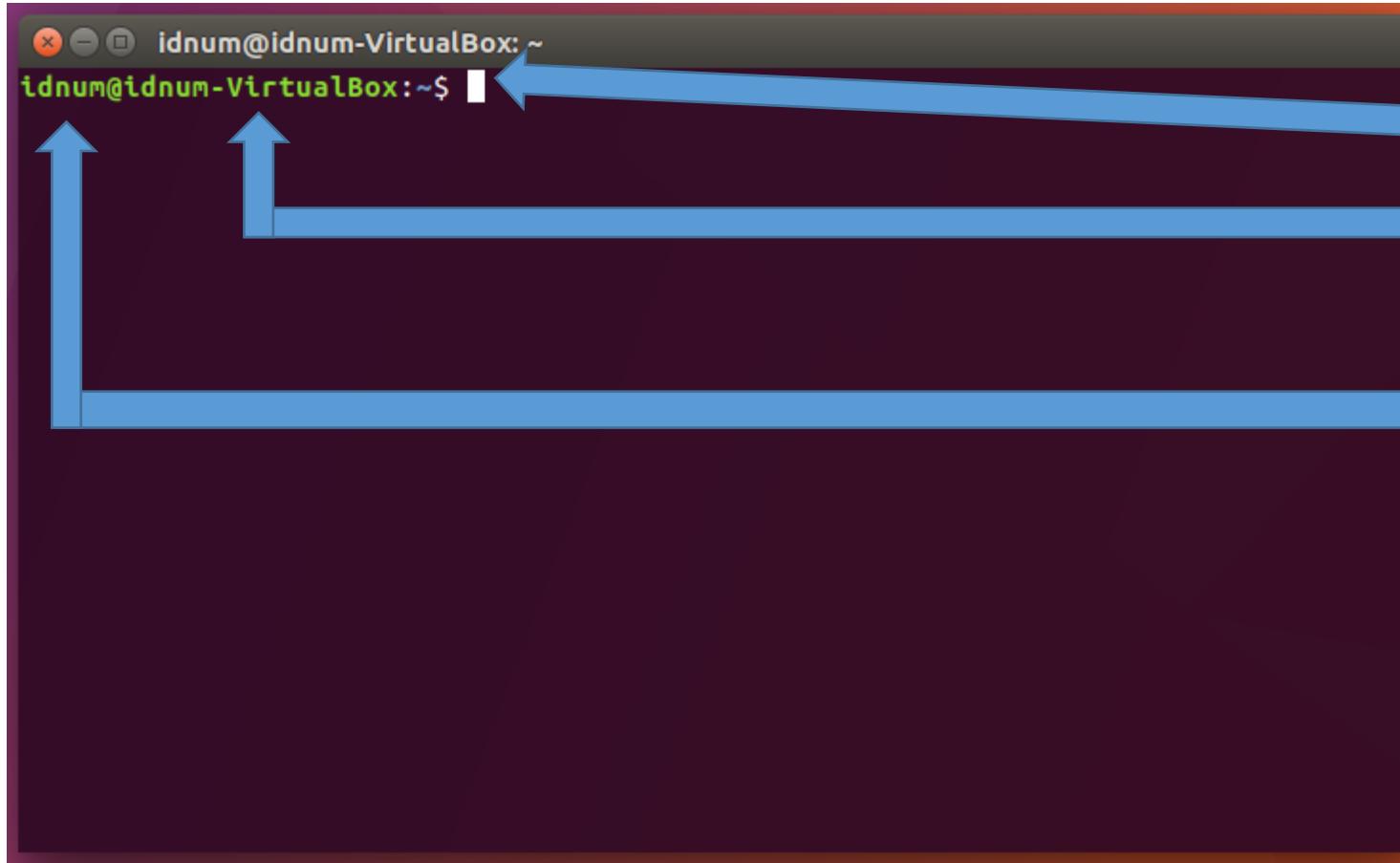
- Bash Commands
 - echo
 - #
 - Find
 - grep
 - pipe (|)
 - cat
 - more
 - sleep
 - touch
- Output Redirection
 - >
 - >>
 - <
 - 2>&1
 - /dev/null
 - /dev/zero

Launching the Linux Terminal



- Click on Ubuntu Search your computer icon.
- Type “terminal”.
- Click on the Terminal icon.

Launching the Linux Terminal



- Shell Prompt
- Computer host name
- User

Terminal – Help Command

```
idnum@idnum-VirtualBox:~$ help
GNU bash, version 4.4.7(1)-release (x86_64-pc-linux-gnu)
These shell commands are defined internally. Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.

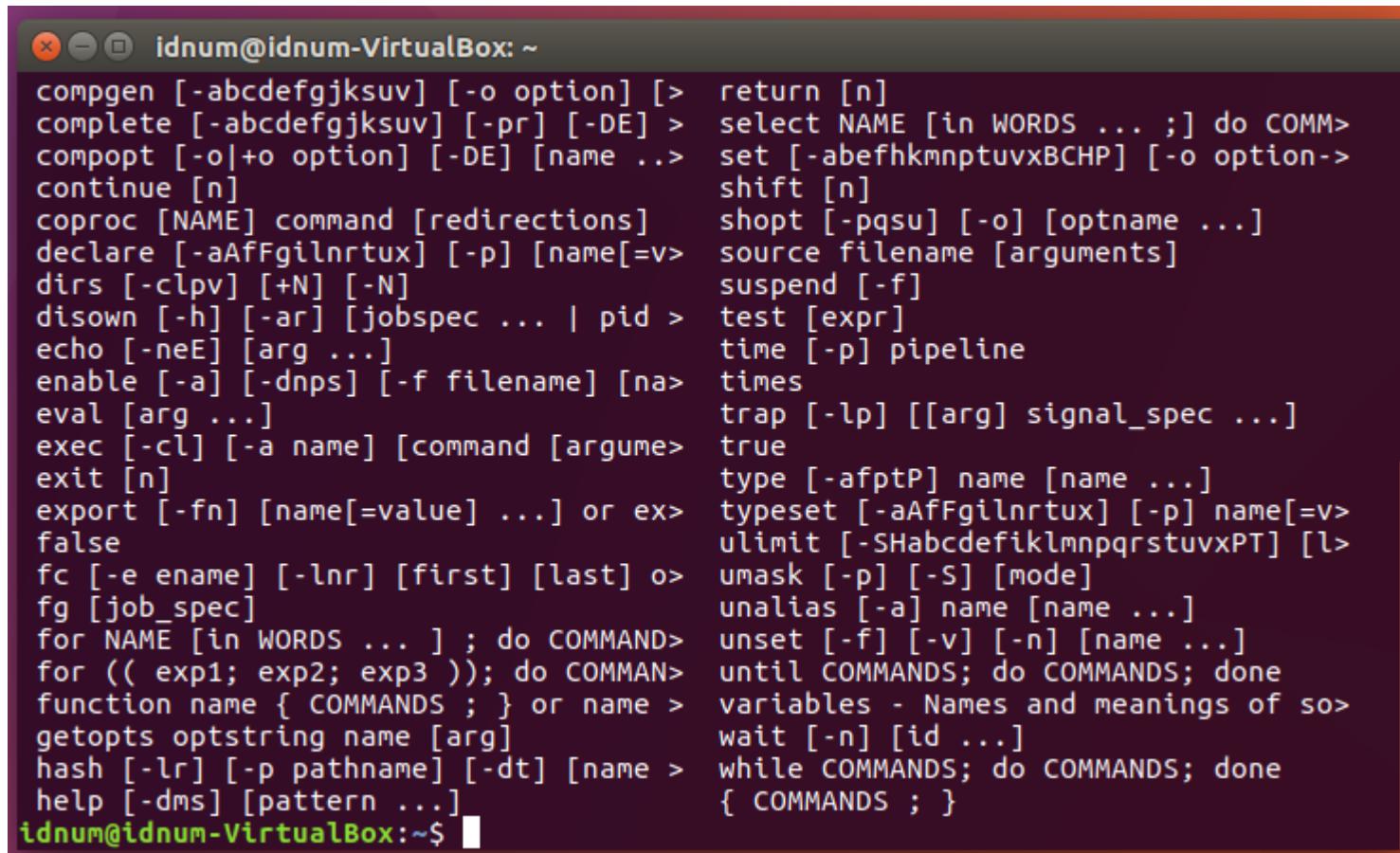
A star (*) next to a name means that the command is disabled.

job_spec [&]
(( expression ))
. filename [arguments]
:
[ arg... ]
[[ expression ]]
alias [-p] [name[=value] ... ]
bg [job_spec ...]
bind [-lpsvPSVX] [-m keymap] [-f file]
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN [| PATTERN]...)...
cd [-L|[-P [-e]] [-@]] [dir]
command [-pVv] command [arg ...]

history [-c] [-d offset] [n] or hist>
if COMMANDS; then COMMANDS; [ elif C>
jobs [-lnprs] [jobspec ...] or jobs >
kill [-s sigspec | -n signum | -sigs>
let arg [arg ...]
local [option] name[=value] ...
logout [n]
mapfile [-d delim] [-n count] [-O or>
popd [-n] [+N | -N]
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [->
readarray [-n count] [-O origin] [-s>
readonly [-aAf] [name[=value] ...] o>
```

- At the ~\$ type help
- Over 60 commands will be listed.
- In most cases you will only use about 5-10 of them.

Terminal – Help Command

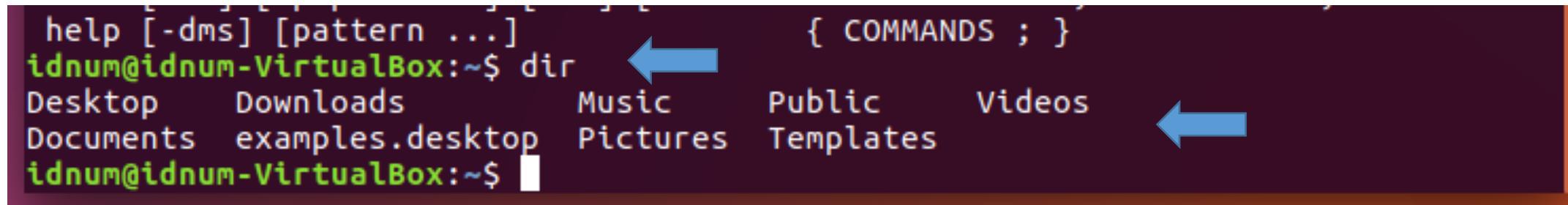


A screenshot of a terminal window titled "idnum@idnum-VirtualBox: ~". The window displays a large list of shell commands, starting with "compgen" and ending with "variables - Names and meanings of so>". The text is white on a dark background. The terminal prompt "idnum@idnum-VirtualBox:~\$" is visible at the bottom.

```
compgen [-abcdefgjksuv] [-o option] [> return [n]
complete [-abcdefgjksuv] [-pr] [-DE] > select NAME [in WORDS ... ;] do COMMAND
compopt [-o]+o option] [-DE] [name ..> set [-abefhkmnptuvwxyzBCHP] [-o option->
continue [n] shift [n]
coproc [NAME] command [redirections] shopt [-pqsu] [-o] [optname ...]
declare [-aAfFgilnrtux] [-p] [name[=v]> source filename [arguments]
dirs [-clpv] [+N] [-N] suspend [-f]
disown [-h] [-ar] [jobspec ... | pid > test [expr]
echo [-neE] [arg ...] time [-p] pipeline
enable [-a] [-dnps] [-f filename] [na> times
eval [arg ...] trap [-lp] [[arg] signal_spec ...]
exec [-cl] [-a name] [command [argume> true
exit [n] type [-afptP] name [name ...]
export [-fn] [name[=value] ...] or ex> typeset [-aAfFgilnrtux] [-p] name[=v>
false ulimit [-SHabcdefiklmnpqrstuvwxyzPT] [l>
fc [-e ename] [-lnr] [first] [last] o> umask [-p] [-S] [mode]
fg [job_spec] unalias [-a] name [name ...]
for NAME [in WORDS ... ] ; do COMMAND> unset [-f] [-v] [-n] [name ...]
for (( exp1; exp2; exp3 )); do COMMAND> until COMMANDS; do COMMANDS; done
function name { COMMANDS ; } or name > variables - Names and meanings of so>
getopts optstring name [arg] wait [-n] [id ...]
hash [-lr] [-p pathname] [-dt] [name > while COMMANDS; do COMMANDS; done
help [-dms] [pattern ...] { COMMANDS ; }
```

- At the ~\$ type help
- Over 60 commands will be listed.
- In most cases you will only use about 5-10 of them.

Linux Terminal – DIR command



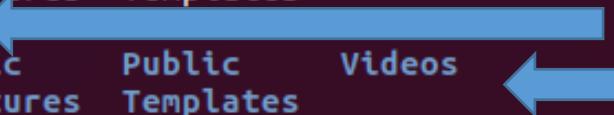
```
help [-dms] [pattern ...] { COMMANDS ; ; }
idnum@idnum-VirtualBox:~$ dir
Desktop   Downloads   Music   Public   Videos
Documents examples.desktop Pictures Templates
idnum@idnum-VirtualBox:~$
```

- Use the `dir` command to list the contents of a directory.
- At the `~$` prompt, type “`dir`”.
- What is listed here?
- Any folders that make sense?
- How many folders are there?

Linux Terminal – ls command

```
function name { COMMANDS ; } or name > variables - Names and meanings of so>
getopts optstring name [arg]           wait [-n] [id ...]
hash [-lr] [-p pathname] [-dt] [name > while COMMANDS; do COMMANDS; done
help [-dms] [pattern ...]             { COMMANDS ; }

idnum@idnum-VirtualBox:~$ dir
Desktop   Downloads      Music      Public      Videos
Documents examples.desktop Pictures  Templates
idnum@idnum-VirtualBox:~$ ls
Desktop   Downloads      Music      Public      Videos
Documents examples.desktop Pictures  Templates
idnum@idnum-VirtualBox:~$
```



- Use the **ls** command to list the contents of a directory.
- At the **~\$** prompt, type “**ls**”.
- What is listed here?
- Any folders that make sense?
- What is different here?
- Blue means folders, white are files.

Linux Terminal – ls command

```
idnum@idnum-VirtualBox:~$ ls -alh
total 104K
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 .
drwxr-xr-x  3 root  root 4.0K Sep 17 12:27 ..
-rw-----  1 idnum idnum 275 Sep 25 09:59 .bash_history
-rw-r--r--  1 idnum idnum 220 Sep 17 12:27 .bash_logout
-rw-r--r--  1 idnum idnum 3.7K Sep 17 12:27 .bashrc
drwx----- 14 idnum idnum 4.0K Sep 17 13:51 .cache
drwx----- 13 idnum idnum 4.0K Sep 17 12:42 .config
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Desktop
-rw-r--r--  1 idnum idnum 25 Sep 17 12:41 .dmrc
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Documents
drwxr-xr-x  2 idnum idnum 4.0K Sep 24 16:46 Downloads
-rw-r--r--  1 idnum idnum 8.8K Sep 17 12:27 examples.desktop
drwxr--r--  2 idnum idnum 4.0K Sep 17 15:00 .hardinfo
-rw-----  1 idnum idnum 358 Sep 17 12:41 .ICEauthority
drwxrwxr-x  3 idnum idnum 4.0K Sep 17 12:41 .local
drwx-----  4 idnum idnum 4.0K Sep 17 12:43 .mozilla
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Music
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Pictures
-rw-r--r--  1 idnum idnum 675 Sep 17 12:27 .profile
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Public
-rw-r--r--  1 idnum idnum    0 Sep 17 14:53 .sudo_as_admin_successful
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Templates
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Videos
-rw-----  1 idnum idnum   61 Sep 17 12:41 .Xauthority
-rw-----  1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
idnum@idnum-VirtualBox:~$
```

- Use the **ls** command to list the contents of a directory.
- At the **~\$** prompt,
 - type “**ls -alh**”.
- What is listed here?
- We see more folders and files as well as sizes.

Linux Terminal – CD command

```
-rw----- 1 idnum idnum 61 Sep 17 12:41 .Xauthority  
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors  
idnum@idnum-VirtualBox:~$ cd Documents  
idnum@idnum-VirtualBox:~/Documents$ █
```

- Use the CD command to navigate into folders.
- cd into the “Documents” folder.
- What has changed?
- What can be done here?

Windows CLI – CD command

```
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Music
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Pictures
-rw-r--r-- 1 idnum idnum 675 Sep 17 12:27 .profile
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Public
-rw-r--r-- 1 idnum idnum 0 Sep 17 14:53 .sudo_as_admin_successful
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Templates
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Videos
-rw----- 1 idnum idnum 61 Sep 17 12:41 .Xauthority
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
idnum@idnum-VirtualBox:~$ cd Documents
idnum@idnum-VirtualBox:~/Documents$ ls -alh ←
total 8.0K
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
idnum@idnum-VirtualBox:~/Documents$ █
```

- Run the **ls -alh** command in this folder.
- What can be seen?
- How many files are there?
- How many folders are there?

Linux Terminal – [.] – Current Directory

```
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 Videos
-rw-------  1 idnum idnum   61 Sep 17 12:41 .Xauthority
-rw-------  1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
idnum@idnum-VirtualBox:~$ cd Documents
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 8.0K
drwxr-xr-x  2 idnum idnum 4.0K Sep 17 12:41 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
idnum@idnum-VirtualBox:~/Documents$ cd .
idnum@idnum-VirtualBox:~/Documents$ █
```

- It seems that the “.” is a directory.
- Lets cd into it.
- Type “cd .”
- Any change?
- What is the purpose of the “.” directory then?
- Will see this later on.

Linux Terminal – [..] – Parent Directory

```
cd /root/Desktop & cd Documents  
idnum@idnum-VirtualBox:~/Documents$ ls -alh  
total 8.0K  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 .  
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..  
idnum@idnum-VirtualBox:~/Documents$ cd .  
idnum@idnum-VirtualBox:~/Documents$ cd ..  
idnum@idnum-VirtualBox:~$ █
```

```
idnum@idnum-VirtualBox:~/Documents$ cd .  
idnum@idnum-VirtualBox:~/Documents$ cd ..  
idnum@idnum-VirtualBox:~/Documents$ ls -alh  
total 104K  
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 .  
drwxr-xr-x 3 root root 4.0K Sep 17 12:27 ..  
-rw----- 1 idnum idnum 275 Sep 25 09:59 .bash_history  
-rw-r--r-- 1 idnum idnum 220 Sep 17 12:27 .bash_logout  
-rw-r--r-- 1 idnum idnum 3.7K Sep 17 12:27 .bashrc  
drwx----- 14 idnum idnum 4.0K Sep 17 13:51 .cache  
drwx----- 13 idnum idnum 4.0K Sep 17 12:42 .config  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Desktop  
-rw-r--r-- 1 idnum idnum 25 Sep 17 12:41 .dmrc  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Documents  
drwxr-xr-x 2 idnum idnum 4.0K Sep 24 16:46 Downloads  
-rw-r--r-- 1 idnum idnum 8.8K Sep 17 12:27 examples.desktop  
drwxr--r-- 2 idnum idnum 4.0K Sep 17 15:00 .hardinfo  
-rw----- 1 idnum idnum 358 Sep 17 12:41 .ICEauthority  
drwxrwxr-x 3 idnum idnum 4.0K Sep 17 12:41 .local  
drwx----- 4 idnum idnum 4.0K Sep 17 12:43 .mozilla  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Music  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Pictures  
-rw-r--r-- 1 idnum idnum 675 Sep 17 12:27 .profile  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Public  
-rw-r--r-- 1 idnum idnum 0 Sep 17 14:53 .sudo_as_admin_successful  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Templates  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Videos  
-rw----- 1 idnum idnum 61 Sep 17 12:41 .Xauthority  
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
```

- There is another directory called “..”.
- Type “..” At the command prompt.
- What happened?
- Did the directory changed?
- Type “ls -alh” at the ~\$ prompt.
- Anything familiar?
- The “..” is the parent directory.
- Changing into this directory will take you to the parent of the current directory.

CD .. To the root

```
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Public
-rw-r--r-- 1 idnum idnum    0 Sep 17 14:53 .sudo_as_admin_successful
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Templates
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Videos
-rw----- 1 idnum idnum   61 Sep 17 12:41 .Xauthority
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
idnum@idnum-VirtualBox:~$ cd ..
idnum@idnum-VirtualBox:/home$ cd ..
idnum@idnum-VirtualBox:/$ cd ..
idnum@idnum-VirtualBox:/$
```



- Keep typing “cd ..” till the path doesn’t change anymore.
- We are now at the root of the drive.
- What is located here?
- We will find out later.

Linux Terminal – Absolute Paths

```
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 .local/share/naresh/seegobin/.Xauthority  
-rw-r--r-- 1 idnum idnum 675 Sep 17 12:27 .profile  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Public  
-rw-r--r-- 1 idnum idnum 0 Sep 17 14:53 .sudo_as_admin_successful  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Templates  
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Videos  
-rw----- 1 idnum idnum 61 Sep 17 12:41 .Xauthority  
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors  
idnum@idnum-VirtualBox:~$ cd ..  
idnum@idnum-VirtualBox:/home$ cd ..  
idnum@idnum-VirtualBox:/$ cd ..  
idnum@idnum-VirtualBox:/$ cd /home/idnum/Documents  
idnum@idnum-VirtualBox:~/Documents$ █
```



- Lets go straight to the Documents folder.
- Type “**cd /home/idnum/Documents**”
- We are back to the Documents folder.
- We used an absolute path to get there.
- No matter where we are in the directory structure, once we enter an absolute path (once it exists, and access is permitted) we will reach there.

Linux Terminal – Absolute Paths

```
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Videos
-rw----- 1 idnum idnum    61 Sep 17 12:41 .Xauthority
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
idnum@idnum-VirtualBox:~$ cd ..
idnum@idnum-VirtualBox:/home$ cd ..
idnum@idnum-VirtualBox:/$ cd ..
idnum@idnum-VirtualBox:/$ cd /home/idnum/Documents
idnum@idnum-VirtualBox:~/Documents$ cd /usr/bin
idnum@idnum-VirtualBox:/usr/bin$ █
```



- Try the absolute paths for other folders, type “`cd /usr/bin`”.
- Did it work?
- What does this mean?
 - There is a folder `usr` with a `bin` folder in it.

Linux Terminal – Relative Paths

```
-rw----- 1 idnum idnum 61 Sep 17 12:41 .Xauthority
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors
idnum@idnum-VirtualBox:~$ cd ..
idnum@idnum-VirtualBox:/home$ cd ..
idnum@idnum-VirtualBox:$ cd ..
idnum@idnum-VirtualBox:$ cd /home/idnum/Documents
idnum@idnum-VirtualBox:~/Documents$ cd /usr/bin
idnum@idnum-VirtualBox:/usr/bin$ cd ../../home/idnum/Do
Documents/ Downloads/
idnum@idnum-VirtualBox:/usr/bin$ cd ../../home/idnum/Documents/
idnum@idnum-VirtualBox:~/Documents$ █
```



- Navigate back to the idnum's Documents folder using:
 - “cd ../../home/idnum/Documents”
- Why did this work?
- Two things happened here:
 1. Each “cd ..” is one parent folder up. Another “cd ..” is another parent folder up, etc.
 2. /bin and /home are the same level. Once you reach there, it is a matter of navigating in the sub folder idnum then the subfolder Documents.
- This is using relative paths.
- To use relative paths it is good to have an idea of your folder structure.

Linux Terminal – Tree command

```
drwxr-xr-x 2 idnum idnum 4.0K Sep 17 12:41 Videos  
-rw----- 1 idnum idnum   61 Sep 17 12:41 .Xauthority  
-rw----- 1 idnum idnum 3.2K Sep 17 12:41 .xsession-errors  
idnum@idnum-VirtualBox:~$ cd ..  
idnum@idnum-VirtualBox:/home$ cd ..  
idnum@idnum-VirtualBox:/$ cd ..  
idnum@idnum-VirtualBox:/$ cd /home/idnum/Documents  
idnum@idnum-VirtualBox:~/Documents$ cd /usr/bin  
idnum@idnum-VirtualBox:/usr/bin$ cd ../../home/idnum/Do  
Documents/ Downloads/  
idnum@idnum-VirtualBox:/usr/bin$ cd ../../home/idnum/Documents/  
idnum@idnum-VirtualBox:~/Documents$ tree  
The program 'tree' is currently not installed. You can install it by typing:  
sudo apt install tree  
idnum@idnum-VirtualBox:~/Documents$ █
```

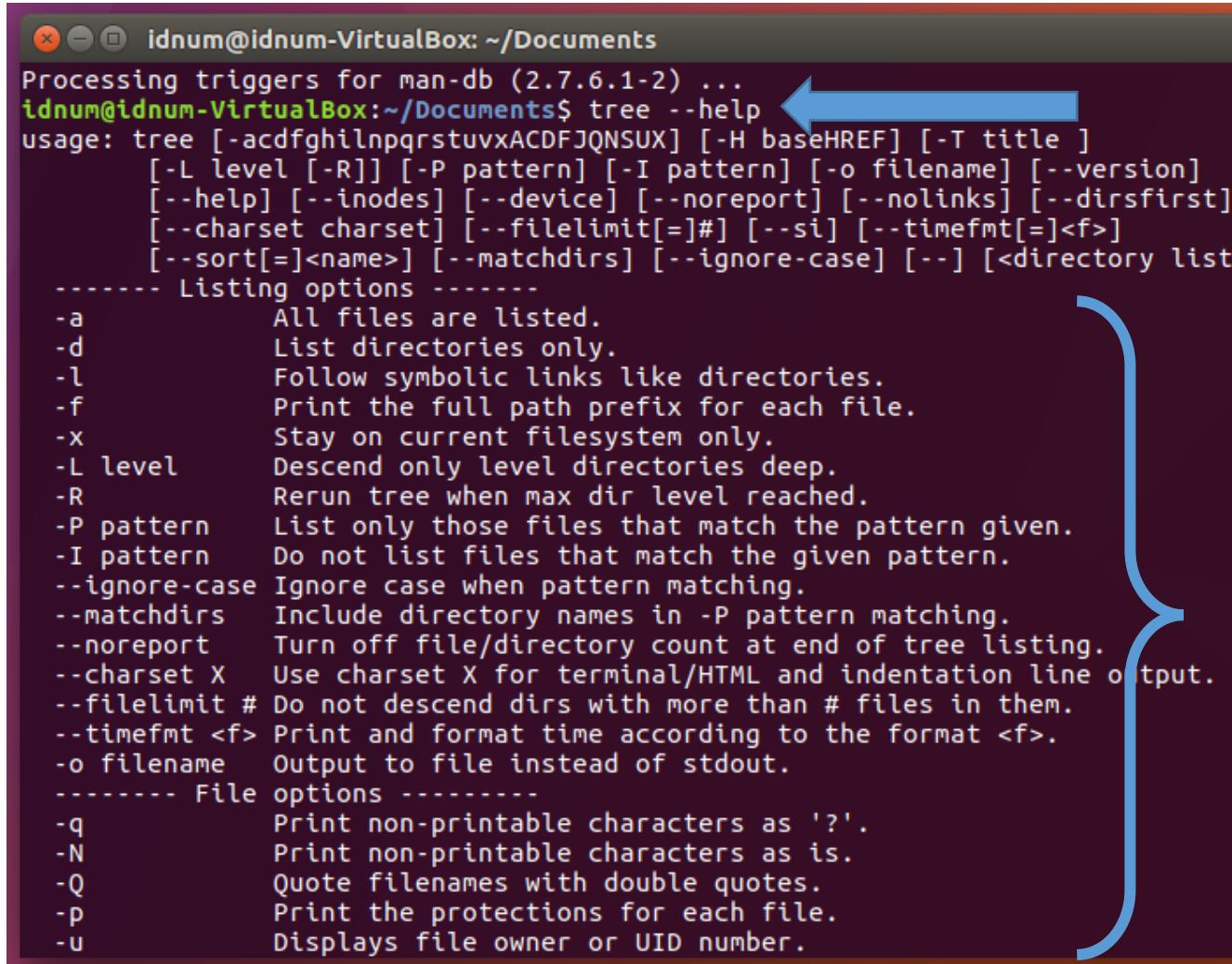
- How can we get an idea of our folder structure?
- Can either **cd** and **ls -alh** each folder and write it out or
- Use the **tree** command.
- Execute the command “**tree**”.
- May need to install it, Execute the command, “**sudo apt install tree**”.

Linux Terminal – Tree command

```
idnum@idnum-VirtualBox:~/Documents$ tree
The program 'tree' is currently not installed. You can install it by typing:
sudo apt install tree
idnum@idnum-VirtualBox:~/Documents$ sudo apt install tree ←
[sudo] password for idnum: ←
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 114 not upgraded.
Need to get 40.7 kB of archives.
After this operation, 105 kB of additional disk space will be used.
Get:1 http://tt.archive.ubuntu.com/ubuntu zesty/universe amd64 tree amd64 1.7.0-
5 [40.7 kB]
Fetched 40.7 kB in 0s (46.0 kB/s)
Selecting previously unselected package tree.
(Reading database ... 203012 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-5_amd64.deb ...
Unpacking tree (1.7.0-5) ...
Setting up tree (1.7.0-5) ...
Processing triggers for man-db (2.7.6.1-2) ...
idnum@idnum-VirtualBox:~/Documents$
```

- May need to install it, Execute the command, “**sudo apt install tree**”.
- Enter your password again.
- The install should be completed successfully.

Linux Terminal – Tree command



```
idnum@idnum-VirtualBox: ~/Documents
Processing triggers for man-db (2.7.6.1-2) ...
idnum@idnum-VirtualBox:~/Documents$ tree --help
usage: tree [-acdfghilnpqrstuvxACDFJQNSUX] [-H baseHREF] [-T title]
            [-L level [-R]] [-P pattern] [-I pattern] [-o filename] [--version]
            [--help] [--inodes] [--device] [--noreport] [--nolinks] [--dirsfirst]
            [--charset charset] [--filelimit[=]#] [--si] [--timefmt[=]<f>]
            [--sort[=]<name>] [--matchdirs] [--ignore-case] [--] [<directory list>
----- Listing options -----
-a          All files are listed.
-d          List directories only.
-l          Follow symbolic links like directories.
-f          Print the full path prefix for each file.
-x          Stay on current filesystem only.
-L level    Descend only level directories deep.
-R          Rerun tree when max dir level reached.
-P pattern   List only those files that match the pattern given.
-I pattern   Do not list files that match the given pattern.
--ignore-case Ignore case when pattern matching.
--matchdirs  Include directory names in -P pattern matching.
--noreport   Turn off file/directory count at end of tree listing.
--charset X  Use charset X for terminal/HTML and indentation line output.
--filelimit # Do not descend dirs with more than # files in them.
--timefmt <f> Print and format time according to the format <f>.
-o filename  Output to file instead of stdout.
----- File options -----
-q          Print non-printable characters as '?'.
-N          Print non-printable characters as is.
-Q          Quote filenames with double quotes.
-p          Print the protections for each file.
-u          Displays file owner or UID number.
```

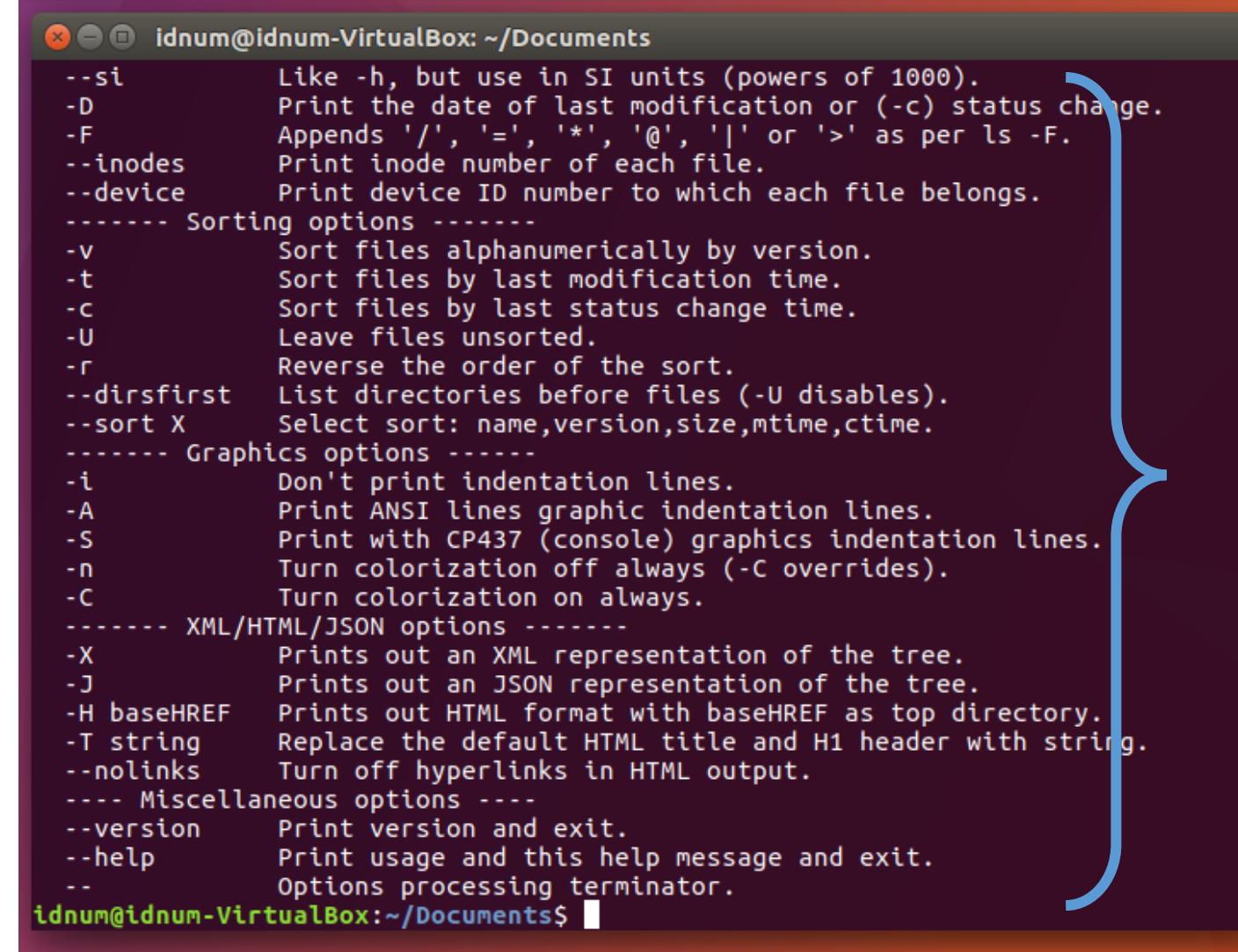
- Execute the command “**tree --help**”.
- There are a lot of options shown.

Linux Terminal – Tree command

```
idnum@idnum-VirtualBox: ~/Documents
--filelimit # Do not descend dirs with more than # files in them.
--timefmt <f> Print and format time according to the format <f>.
-o filename Output to file instead of stdout.
----- File options -----
-q          Print non-printable characters as '?'.
-N          Print non-printable characters as is.
-Q          Quote filenames with double quotes.
-p          Print the protections for each file.
-u          Displays file owner or UID number.
-g          Displays file group owner or GID number.
-s          Print the size in bytes of each file.
-h          Print the size in a more human readable way.
--si        Like -h, but use in SI units (powers of 1000).
-D          Print the date of last modification or (-c) status change.
-F          Appends '/', '=', '*', '@', ']' or '>' as per ls -F.
--inodes    Print inode number of each file.
--device    Print device ID number to which each file belongs.
----- Sorting options -----
-v          Sort files alphanumerically by version.
-t          Sort files by last modification time.
-c          Sort files by last status change time.
-U          Leave files unsorted.
-r          Reverse the order of the sort.
--dirsfirst List directories before files (-U disables).
--sort X    Select sort: name,version,size,mtime,ctime.
----- Graphics options -----
-i          Don't print indentation lines.
-A          Print ANSI lines graphic indentation lines.
-S          Print with CP437 (console) graphics indentation lines.
-n          Turn colorization off always (-C overrides).
```

- There are a lot of options shown.

Linux Terminal – Tree command



```
idnum@idnum-VirtualBox: ~/Documents
--si      Like -h, but use in SI units (powers of 1000).
-D       Print the date of last modification or (-c) status change.
-F       Appends '/', '=', '*', '@', '||' or '>' as per ls -F.
--inodes  Print inode number of each file.
--device  Print device ID number to which each file belongs.
----- Sorting options -----
-v       Sort files alphanumerically by version.
-t       Sort files by last modification time.
-c       Sort files by last status change time.
-U       Leave files unsorted.
-r       Reverse the order of the sort.
--dirsfirst List directories before files (-U disables).
--sort X  Select sort: name,version,size,mtime,ctime.
----- Graphics options -----
-i       Don't print indentation lines.
-A       Print ANSI lines graphic indentation lines.
-S       Print with CP437 (console) graphics indentation lines.
-n       Turn colorization off always (-C overrides).
-C       Turn colorization on always.
----- XML/HTML/JSON options -----
-X      Prints out an XML representation of the tree.
-J      Prints out an JSON representation of the tree.
-H baseHref Prints out HTML format with baseHref as top directory.
-T string Replace the default HTML title and H1 header with string.
--nolinks Turn off hyperlinks in HTML output.
----- Miscellaneous options -----
--version Print version and exit.
--help    Print usage and this help message and exit.
--      Options processing terminator.
idnum@idnum-VirtualBox:~/Documents$
```

- There are a lot of options shown.

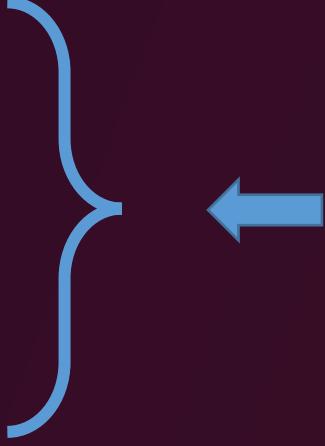
Linux Terminal – Tree command

```
--string      Replace the default HTML title and hi header with STRING.  
--nolinks    Turn off hyperlinks in HTML output.  
---- Miscellaneous options ----  
--version    Print version and exit.  
--help       Print usage and this help message and exit.  
--          Options processing terminator.  
idnum@idnum-VirtualBox:~/Documents$ tree -d ←  
. ←  
0 directories  
idnum@idnum-VirtualBox:~/Documents$ ┌─←
```

- Execute the command “**tree -d**”.
- No subfolders exist.
 - Why is this the case?
 - It is correct.

Linux Terminal – Tree command

```
idnum@idnum-VirtualBox:~/Documents$ tree -d  
.  
0 directories  
idnum@idnum-VirtualBox:~/Documents$ tree -d /home ←  
/home ←  
└── idnum  
    ├── Desktop  
    ├── Documents  
    ├── Downloads  
    ├── Music  
    ├── Pictures  
    ├── Public  
    ├── Templates  
    └── Videos  
9 directories  
idnum@idnum-VirtualBox:~/Documents$ █
```



- Try the command “tree -d / **home**”.
- What do we get?
- Anything familiar?
- We can see the tree structure of the **/home** folder.
- Try this with other folders.

Linux Terminal – Tree command

```
idnum@idnum-VirtualBox: ~/Documents
└── upstart
    ├── mail
    └── metrics
    ├── opt
    ├── run -> /run
    ├── snap
    └── spool
        ├── anacron
        ├── cron
        │   ├── crontabs [error opening dir]
        │   ├── cups [error opening dir]
        │   └── libreoffice
        │       ├── uno_packages
        │       │   └── cache
        │       └── mail -> ../mail
        └── syslog [error opening dir]
    └── tmp
        ├── systemd-private-6508d7ef59ad499c9e1b116a69a4a9f0-colord.service-99cf
        ├── Ir [error opening dir]
        ├── systemd-private-6508d7ef59ad499c9e1b116a69a4a9f0-fwupd.service-0m2E9
        └── Z [error opening dir]
    └── e-5CuXi2 [error opening dir]
        ├── systemd-private-6508d7ef59ad499c9e1b116a69a4a9f0-rtkit-daemon.service
        └── systemd-private-6508d7ef59ad499c9e1b116a69a4a9f0-systemd-resolved.se
    rvice-e3aH0d [error opening dir]
        └── systemd-private-6508d7ef59ad499c9e1b116a69a4a9f0-systemd-timesyncd.s
    ervice-uKTEa0 [error opening dir]

35180 directories
idnum@idnum-VirtualBox:~/Documents$
```

Try this with other folders.

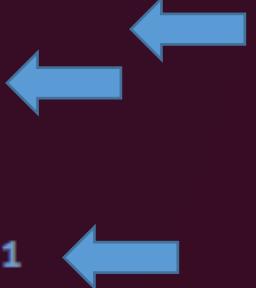
- This will take a while;
- Execute the command “`tree -d /`”.
- What happened here?

The “`tree -d /`” command showed the directory structure of the entire file system.

- It is a lot of folders and would not make sense to use it now.
- Keep it simple.

Linux Terminal – mkdir

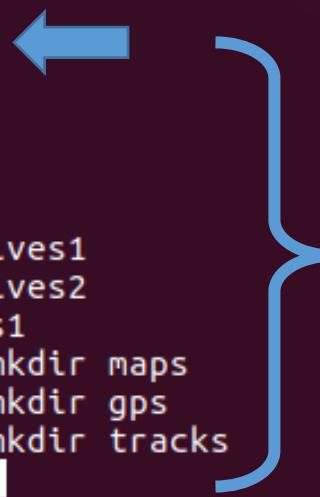
```
35180 directories
idnum@idnum-VirtualBox:~/Documents$ mkdir folder1
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 12K
drwxr-xr-x 3 idnum idnum 4.0K Sep 25 13:40 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
drwxr-xr-x 2 idnum idnum 4.0K Sep 25 13:40 folder1
idnum@idnum-VirtualBox:~/Documents$
```



- To create a directory use the **mkdir** command with parameters.
- Execute the command “**mkdir folder1**”.
- To see if it was created, enter the “**ls -alh**” command.
 - As you can see the folder was created.

Linux Terminal – mkdir

```
idnum@idnum-VirtualBox:~/Documents$ mkdir folder1
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 12K
drwxr-xr-x 3 idnum idnum 4.0K Sep 25 13:40 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
drwxr-xr-x 2 idnum idnum 4.0K Sep 25 13:40 folder1
idnum@idnum-VirtualBox:~/Documents$ mkdir folder2
idnum@idnum-VirtualBox:~/Documents$ mkdir groceries
idnum@idnum-VirtualBox:~/Documents$ mkdir car
idnum@idnum-VirtualBox:~/Documents$ mkdir games
idnum@idnum-VirtualBox:~/Documents$ cd folder2
idnum@idnum-VirtualBox:~/Documents/folder2$ mkdir archives1
idnum@idnum-VirtualBox:~/Documents/folder2$ mkdir archives2
idnum@idnum-VirtualBox:~/Documents/folder2$ cd archives1
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ mkdir maps
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ mkdir gps
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ mkdir tracks
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ █
```



- Create a few more folders:
 - “**mkdir folder2**”
 - “**mkdir groceries**”
 - “**mkdir car**”
 - “**mkdir games**”
- Create some sub folders
 - “**cd folder2**”
 - “**mkdir archives1**”
 - “**mkdir archives2**”
 - “**cd archives1**”
 - “**mkdir maps**”
 - “**mkdir gps**”
 - “**mkdir tracks**”

Linux Terminal – mkdir

```
idnum@idnum-VirtualBox:~/Documents$ cd folder2
idnum@idnum-VirtualBox:~/Documents/folder2$ mkdir archives1
idnum@idnum-VirtualBox:~/Documents/folder2$ mkdir archives2
idnum@idnum-VirtualBox:~/Documents/folder2$ cd archives1
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ mkdir maps
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ mkdir gps
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ mkdir tracks
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$ tree -d /home/idnum
/home/idnum
├── Desktop
├── Documents
│   ├── car
│   ├── folder1
│   └── folder2
│       ├── archives1
│       │   ├── gps
│       │   ├── maps
│       │   └── tracks
│       └── archives2
│           └── games
└── groceries
└── Downloads
└── Music
└── Pictures
└── Public
└── Templates
└── Videos
18 directories
idnum@idnum-VirtualBox:~/Documents/folder2/archives1$
```



- What does this look like?
- Execute the command “**tree -d /home/idnum**”.
- Did it look as it was expected to be created?

Linux Terminal – rm

```
idnum@idnum-VirtualBox: ~/Documents/folder2
idnum@idnum-VirtualBox:~/Documents/folder2$ rm archives2
rm: cannot remove 'archives2': Is a directory
idnum@idnum-VirtualBox:~/Documents/folder2$ rm -r archives2
idnum@idnum-VirtualBox:~/Documents/folder2$ ls -alh
total 12K
drwxr-xr-x 3 idnum idnum 4.0K Sep 25 14:04 .
drwxr-xr-x 7 idnum idnum 4.0K Sep 25 13:43 ..
drwxr-xr-x 5 idnum idnum 4.0K Sep 25 13:44 archives1
idnum@idnum-VirtualBox:~/Documents/folder2$ tree -d /home/idnum
/home/idnum
    Desktop
    Documents
        car
        folder1
        folder2
            archives1
                gps
                maps
                tracks
        games
        groceries
    Downloads
    Music
    Pictures
    Public
    Templates
    Videos
17 directories
idnum@idnum-VirtualBox:~/Documents/folder2$
```

```
idnum@idnum-VirtualBox:~/Documents/folder2$ tree -d /home/idnum
/home/idnum
    Desktop
    Documents
        car
        folder1
        folder2
            archives1
                gps
                maps
                tracks
        games
        groceries
    Downloads
```

- Move to the parent directory, “cd ..”.
- Remove the **archives2** folder using the command, “rm **archives2**”.
 - Use “rm -r **archives2**” instead.
- Perform the “ls -alh” and “tree -d /home/idnum” commands to see if the folder was removed.

Linux Terminal – rm

```
x idnum@idnum-VirtualBox: ~/Documents/folder2
idnum@idnum-VirtualBox:~/Documents/folder2$ clear
idnum@idnum-VirtualBox:~/Documents/folder2$ rm -r archives1
idnum@idnum-VirtualBox:~/Documents/folder2$ rm --help
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).

-f, --force          ignore nonexistent files and arguments, never prompt
-i                  prompt before every removal
-I                  prompt once before removing more than three files, or
                   when removing recursively; less intrusive than -i,
                   while still giving protection against most mistakes
--interactive[=WHEN]  prompt according to WHEN: never, once (-I), or
                   always (-i); without WHEN, prompt always
--one-file-system    when removing a hierarchy recursively, skip any
                   directory that is on a file system different from
                   that of the corresponding command line argument
--no-preserve-root   do not treat '/' specially
--preserve-root      do not remove '/' (default)
-r, -R, --recursive  remove directories and their contents recursively
-d, --dir            remove empty directories
-v, --verbose         explain what is being done
--help              display this help and exit
--version           output version information and exit

By default, rm does not remove directories. Use the --recursive (-r or -R)
option to remove each listed directory, too, along with all of its contents.

To remove a file whose name starts with a '-', for example '-foo',
use one of these commands:
```

- Remove the **archives1** folder by issuing the “**rm -r archives1**” command.
- Did it work?
- Execute the command “**rm --help**” to see possible options.
 - The **-r** option, which was used, removed the directories and their contents recursively.

Linux Terminal – mv

The screenshot shows a terminal window with the following session:

```
idnum@idnum-VirtualBox: ~/Documents
Full documentation at: <http://www.gnu.org/software/coreutils/rm>
or available locally via: info '(coreutils) rm invocation'
idnum@idnum-VirtualBox:~/Documents/folder2$ cd ..
idnum@idnum-VirtualBox:~/Documents$ mv car folder1
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 24K
drwxr-xr-x  6 idnum idnum 4.0K Sep 25 14:20 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
idnum@idnum-VirtualBox:~/Documents$ tree -d /home/idnum
/home/idnum
└── Desktop
    └── Documents
        ├── folder1
        │   └── car
        ├── folder2
        ├── games
        └── groceries
        └── Downloads
        └── Music
        └── Pictures
        └── Public
        └── Templates
        └── Videos
13 directories
idnum@idnum-VirtualBox:~/Documents$
```

A blue curly brace on the left side groups the "Documents" directory and its contents (folder1, folder2, games, groceries). Four blue arrows point from the right side of the terminal window towards the brace, indicating the scope of the mv command.

- Move to the parent directory “cd ..”.
- Move the **car** folder into **folder1** by issuing the command “**mv car folder1**”.
- Did it work?
- Perform the “**ls -alh**” and “**tree -d /home/idnum**” commands to see if the folder was moved.

Linux Directories

```
.o 19 generated  
idnum@idnum-VirtualBox:~/Documents$ ls /  
bin  dev  initrd.img      lib64      mnt      root    snap      sys      var  
boot  etc  initrd.img.old  lost+found  opt      run     srv      tmp      vmlinuz  
cdrom  home  lib          media      proc     sbin    swapfile  usr      vmlinuz.old  
idnum@idnum-VirtualBox:~/Documents$ █
```

- Execute the command “ls /”
- What are the main folders listed?

Linux Directories

```
idnum@idnum-VirtualBox: ~/Documents
idnum@idnum-VirtualBox:~/Documents$ ls -alh /
total 473M
drwxr-xr-x 24 root root 4.0K Sep 18 14:27 .
drwxr-xr-x 24 root root 4.0K Sep 18 14:27 ..
drwxr-xr-x 2 root root 4.0K Sep 18 14:26 bin
drwxr-xr-x 3 root root 4.0K Sep 18 14:28 boot
drwxrwxr-x 2 root root 4.0K Sep 17 12:26 cdrom
drwxr-xr-x 18 root root 3.8K Sep 17 12:38 dev
drwxr-xr-x 129 root root 12K Sep 26 03:37 etc
drwxr-xr-x 3 root root 4.0K Sep 17 12:27 home
lrwxrwxrwx 1 root root 33 Sep 18 14:27 initrd.img -> boot/initrd.img-4.16
35-generic
lrwxrwxrwx 1 root root 33 Sep 17 12:28 initrd.img.old -> boot/initrd.img-
0.0-19-generic
drwxr-xr-x 22 root root 4.0K Sep 17 12:28 lib
drwxr-xr-x 2 root root 4.0K Sep 18 14:26 lib64
drwx----- 2 root root 16K Sep 17 12:25 lost+found
drwxr-xr-x 2 root root 4.0K Apr 11 23:07 media
drwxr-xr-x 2 root root 4.0K Apr 11 23:07 mnt
drwxr-xr-x 2 root root 4.0K Apr 11 23:07 opt
dr-xr-xr-x 210 root root 0 Sep 17 12:38 proc
drwx----- 3 root root 4.0K Apr 11 23:20 root
drwxr-xr-x 28 root root 1000 Sep 18 14:28 run
drwxr-xr-x 2 root root 12K Sep 18 14:26 sbin
drwxr-xr-x 2 root root 4.0K Apr 6 04:32 snap
drwxr-xr-x 2 root root 4.0K Apr 11 23:07 srv
-rw----- 1 root root 473M Sep 17 12:25 swapfile
dr-xr-xr-x 13 root root 0 Sep 26 06:14 sys
drwxrwxrwt 14 root root 4.0K Sep 26 06:17 tmp
drwxr-xr-x 11 root root 4.0K Apr 11 23:13 usr
```

- Execute the command “ls -alh /”
- /
 - Root folder (filesystem), contains the entire filesystem for the Linux OS.
- /boot
 - Contains the Linux boot kernel and bootloaders.
- /home
 - The user profiles.
- /etc
 - Contains OS and applications configurations. Can be considered the Linux equivalent of the windows registry.

Linux Directories

```
idnum@idnum-VirtualBox: ~/Documents$ ls -l
drwxrwxr-x  2 root root 4.0K Sep 17 12:26 cdrom
drwxr-xr-x 18 root root 3.8K Sep 17 12:38 dev
drwxr-xr-x 129 root root 12K Sep 26 03:37 etc
drwxr-xr-x  3 root root 4.0K Sep 17 12:27 home
lrwxrwxrwx  1 root root   33 Sep 18 14:27 initrd.img -> boot/initrd.img-4.15.0-generic
lrwxrwxrwx  1 root root   33 Sep 17 12:28 initrd.img.old -> boot/initrd.i
0.0-19-generic
drwxr-xr-x 22 root root 4.0K Sep 17 12:28 lib
drwxr-xr-x  2 root root 4.0K Sep 18 14:26 lib64
drwx----- 2 root root 16K Sep 17 12:25 lost+found
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 media
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 mnt
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 opt
dr-xr-xr-x 210 root root    0 Sep 17 12:38 proc
drwx----- 3 root root 4.0K Apr 11 23:20 root
drwxr-xr-x 28 root root 1000 Sep 18 14:28 run
drwxr-xr-x  2 root root 12K Sep 18 14:26 sbin
drwxr-xr-x  2 root root 4.0K Apr  6 04:32 snap
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 srv
-rw-----  1 root root 473M Sep 17 12:25 swapfile
dr-xr-xr-x 13 root root    0 Sep 26 06:14 sys
drwxrwxrwt 14 root root 4.0K Sep 26 06:17 tmp
drwxr-xr-x 11 root root 4.0K Apr 11 23:13 usr
drwxr-xr-x 14 root root 4.0K Apr 11 23:21 var
lrwxrwxrwx  1 root root   30 Sep 18 14:27 vmlinuz -> boot/vmlinuz-4.10.0-eric
lrwxrwxrwx  1 root root   30 Sep 17 12:28 vmlinuz.old -> boot/vmlinuz-4.1
-generic
idnum@idnum-VirtualBox:~/Documents$ 
```

- **/bin**
 - All executable binary programs for the Linux OS.
- **/sbin**
 - Binary executable programs for system administration.
- **/usr**
 - User executable binaries such as firefox and other compiled applications.
- **/var**
 - Contains log data and temp files.
- **/media**
 - Location of where external devices are mounted like CD/DVD ROMs, USB storage devices.
- **/mnt**
 - Legacy location for mounting filesystems.

Linux Directories

```
idnum@idnum-VirtualBox: ~/Documents
drwxrwxr-x  2 root root 4.0K Sep 17 12:26 cdrom
drwxr-xr-x 18 root root 3.8K Sep 17 12:38 dev
drwxr-xr-x 129 root root 12K Sep 26 03:37 etc
drwxr-xr-x  3 root root 4.0K Sep 17 12:27 home
lrwxrwxrwx  1 root root   33 Sep 18 14:27 initrd.img -> boot/initrd.img-4
35-generic
lrwxrwxrwx  1 root root   33 Sep 17 12:28 initrd.img.old -> boot/initrd.i
0.0-19-generic
drwxr-xr-x 22 root root 4.0K Sep 17 12:28 lib
drwxr-xr-x  2 root root 4.0K Sep 18 14:26 lib64
drwx----- 2 root root 16K Sep 17 12:25 lost+found
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 media
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 mnt
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 opt
dr-xr-xr-x 210 root root    0 Sep 17 12:38 proc
drwx----- 3 root root 4.0K Apr 11 23:20 root
drwxr-xr-x  28 root root 1000 Sep 18 14:28 run
drwxr-xr-x  2 root root 12K Sep 18 14:26 sbin
drwxr-xr-x  2 root root 4.0K Apr  6 04:32 snap
drwxr-xr-x  2 root root 4.0K Apr 11 23:07 srv
-rw-----  1 root root 473M Sep 17 12:25 swapfile
dr-xr-xr-x 13 root root    0 Sep 26 06:14 sys
drwxrwxrwt 14 root root 4.0K Sep 26 06:17 tmp
drwxr-xr-x 11 root root 4.0K Apr 11 23:13 usr
drwxr-xr-x 14 root root 4.0K Apr 11 23:21 var
lrwxrwxrwx  1 root root   30 Sep 18 14:27 vmlinuz -> boot/vmlinuz-4.10.0-
eric
lrwxrwxrwx  1 root root   30 Sep 17 12:28 vmlinuz.old -> boot/vmlinuz-4.1
-generic
idnum@idnum-VirtualBox:~/Documents$
```

- /root

- The home directory of the root user. Totally different from the root filesystem or root folder.

- /opt

- Contains third party applications like java.

- /dev

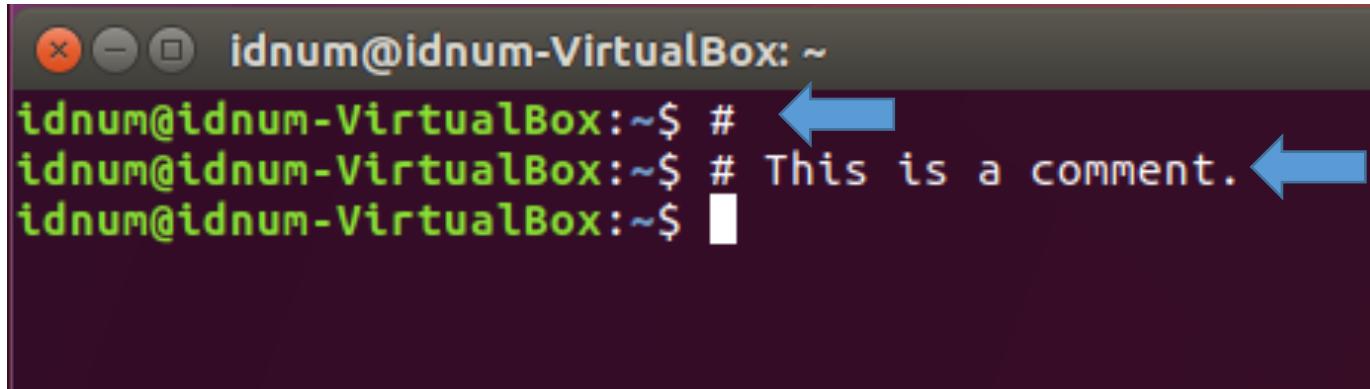
- Contains device files for all hardware devices including CPU, disk drives.

Linux Terminal Introduction - echo

```
-generic  
idnum@idnum-VirtualBox:~/Documents$ echo ←  
  
idnum@idnum-VirtualBox:~/Documents$ echo This is a comment to the console. ←  
This is a comment to the console.  
idnum@idnum-VirtualBox:~/Documents$ █
```

- Execute the command “echo”.
- What can be observed?
- Execute the command “echo **This is a comment to the console.**”.
- As in windows the **echo** command output contents to the console.
Good for debugging purposes.

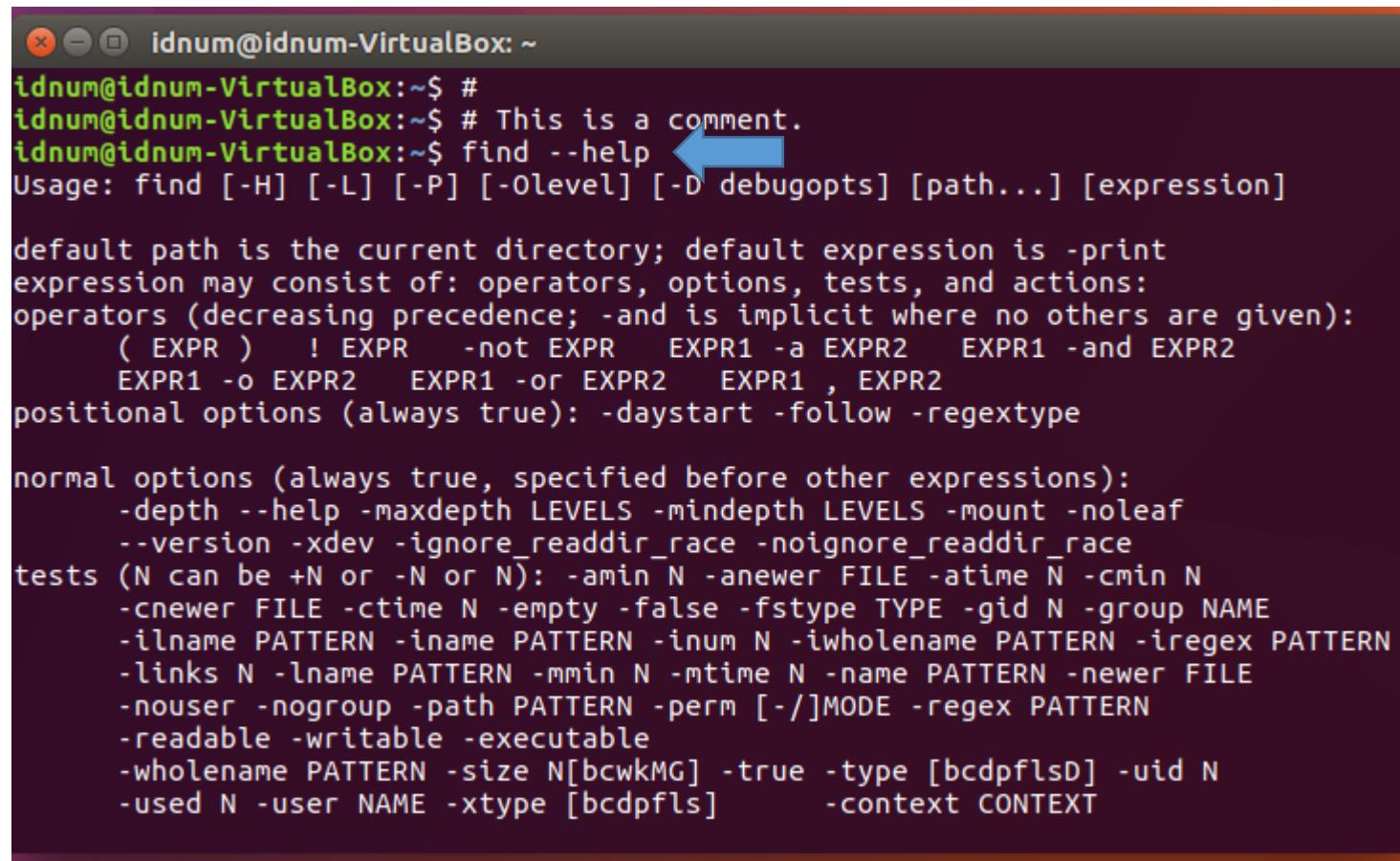
Linux Terminal Introduction -



A screenshot of a Linux terminal window titled "idnum@idnum-VirtualBox: ~". The terminal shows the following text:
idnum@idnum-VirtualBox:~\$ # ←
idnum@idnum-VirtualBox:~\$ # This is a comment. ←
idnum@idnum-VirtualBox:~\$ █

- Execute the command “#”.
- Anything was observed?
- Execute the command “# This is a comment.”.
- You should get the same result as before.
- The # command is for comments in script files. Anything after the # is ignored.

Linux Terminal Introduction - find



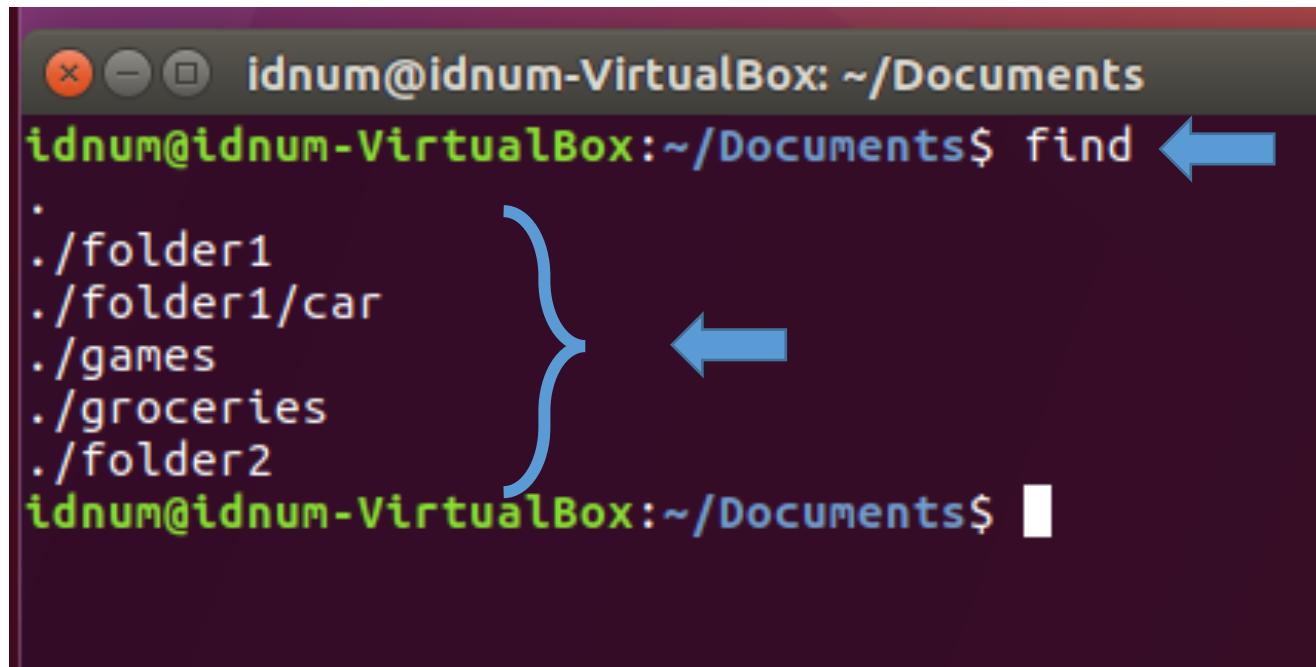
```
idnum@idnum-VirtualBox:~$ #
idnum@idnum-VirtualBox:~$ # This is a comment.
idnum@idnum-VirtualBox:~$ find --help ←
Usage: find [-H] [-L] [-P] [-Olevel] [-D debugopts] [path...] [expression]

default path is the current directory; default expression is -print
expression may consist of: operators, options, tests, and actions:
operators (decreasing precedence; -and is implicit where no others are given):
  ( EXPR ) ! EXPR -not EXPR EXPR1 -a EXPR2 EXPR1 -and EXPR2
  EXPR1 -o EXPR2 EXPR1 -or EXPR2 EXPR1 , EXPR2
positional options (always true): -daystart -follow -regextype

normal options (always true, specified before other expressions):
  -depth --help -maxdepth LEVELS -mindepth LEVELS -mount -noleaf
  --version -xdev -ignore_readdir_race -noignore_readdir_race
tests (N can be +N or -N or N): -amin N -anewer FILE -atime N -cmin N
  -cnewer FILE -ctime N -empty -false -fstype TYPE -gid N -group NAME
  -ilname PATTERN -iname PATTERN -inum N -iwholename PATTERN -iregex PATTERN
  -links N -lname PATTERN -mmin N -mtime N -name PATTERN -newer FILE
  -nouser -nogroup -path PATTERN -perm [/-]MODE -regex PATTERN
  -readable -writable -executable
  -wholename PATTERN -size N[bckwMG] -true -type [bcdpflsD] -uid N
  -used N -user NAME -xtype [bcdpfls]      -context CONTEXT
```

- Execute the command “**find --help**”.
- You will see a list of parameters on how to use the **find** command.

Linux Terminal Introduction - find

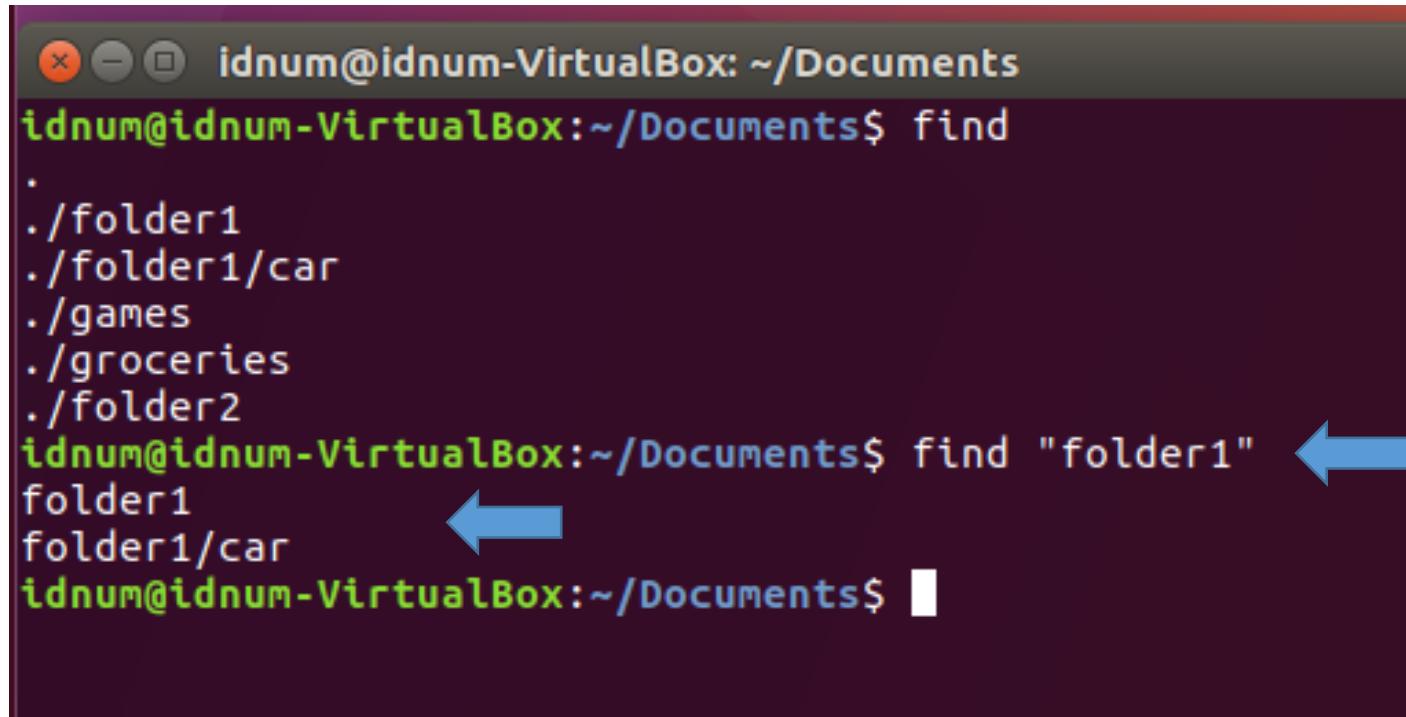


A screenshot of a Linux terminal window titled "idnum@idnum-VirtualBox: ~/Documents". The terminal shows the command "find" being run at the prompt. The output lists several files and folders in the current directory: '.', './folder1', './folder1/car', './games', './groceries', and './folder2'. A blue curly brace is placed over the first four items ('.', './folder1', './folder1/car', './games') to group them together. A blue arrow points from the brace to the right, indicating the scope of the search command.

```
idnum@idnum-VirtualBox:~/Documents$ find
.
./folder1
./folder1/car
./games
./groceries
./folder2
idnum@idnum-VirtualBox:~/Documents$
```

- Execute the command “**find**”.
- It shows the list of folders and files in the current directory.

Linux Terminal Introduction - find



A screenshot of a Linux terminal window titled 'idnum@idnum-VirtualBox: ~/Documents'. The terminal shows the following command execution:

```
idnum@idnum-VirtualBox:~/Documents$ find  
.  
./folder1  
./folder1/car  
./games  
./groceries  
./folder2  
idnum@idnum-VirtualBox:~/Documents$ find "folder1"  
folder1  
folder1/car  
idnum@idnum-VirtualBox:~/Documents$
```

Two blue arrows point from the text 'folder1' in the second 'find' command to the first 'folder1' in the output, highlighting the search results.

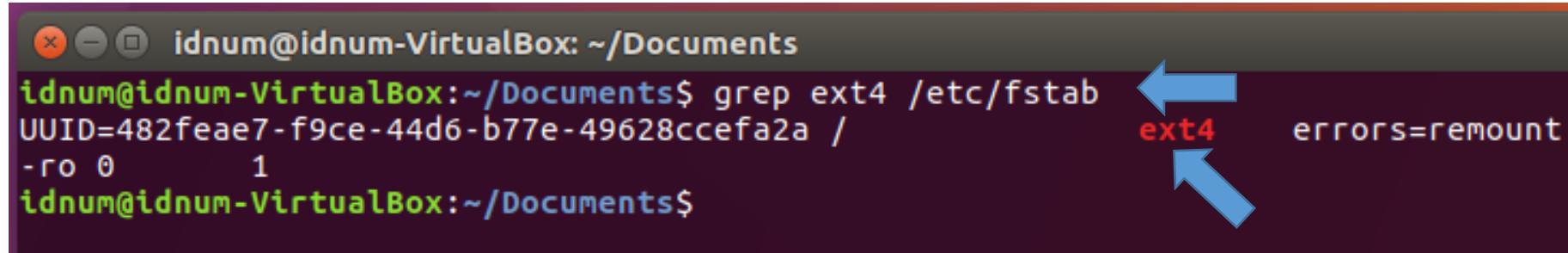
- Execute the command ‘**find “folder1”**’.
- It shows the list of folders and files in the current directory and subdirectories that matches the string **folder1**.

Linux Terminal Introduction - find

```
folder1/car  
idnum@idnum-VirtualBox:~/Documents$ find ./ -name "folder1"  
./folder1  
idnum@idnum-VirtualBox:~/Documents$ find ./ -name "folder*" ←  
./folder1 ←  
./folder2 ←  
idnum@idnum-VirtualBox:~/Documents$
```

- Execute the command ‘**find ./ -name “folder1”**’.
- What is the result?
- It returns only the folder name alone.
- Execute the command ‘**find ./ -name “folder*”**’.
- The * matches any folder named folder1, folder2 etc.

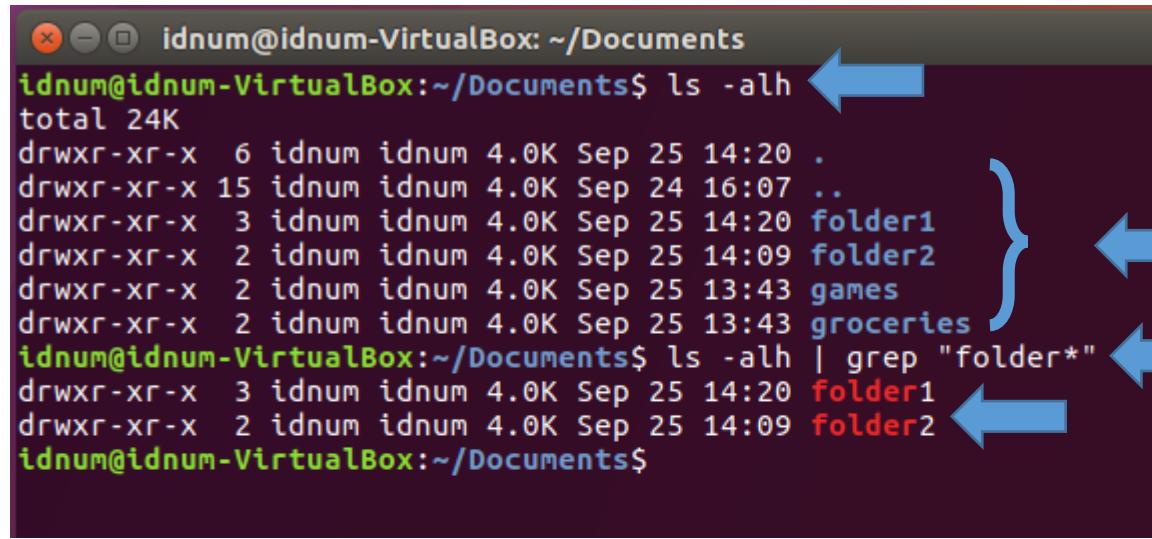
Linux Terminal Introduction - grep



```
idnum@idnum-VirtualBox: ~/Documents
idnum@idnum-VirtualBox:~/Documents$ grep ext4 /etc/fstab
UUID=482feae7-f9ce-44d6-b77e-49628ccefa2a /
-ro 0      1
idnum@idnum-VirtualBox:~/Documents$
```

- While find is more for **files** and file systems, the **grep** utility is much more powerful when searching for strings in files.
- Execute the command “**grep ext4 /etc/fstab**”.
- What happened here?
- Essentially, **grep** read the file **/etc/fstab** and with the filter showing lines containing “**ext4**”.
- We will meet up with this later.

Linux Terminal Introduction – pipe (|)



The screenshot shows a terminal window titled 'idnum@idnum-VirtualBox: ~/Documents'. The user runs the command 'ls -alh' to list all files and folders in the current directory. The output includes a folder named 'groceries' and two other folders named 'folder1' and 'folder2'. Then, the user runs the command 'ls -alh | grep "folder*"', which pipes the output of the first command to the 'grep' command. The 'grep' command filters the output to show only lines containing 'folder*' in them, resulting in the output 'folder1' and 'folder2' being highlighted in red.

```
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 24K
drwxr-xr-x  6 idnum idnum 4.0K Sep 25 14:20 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
idnum@idnum-VirtualBox:~/Documents$ ls -alh | grep "folder*"
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
idnum@idnum-VirtualBox:~/Documents$
```

- A pipe is a form of redirection to send output from one program to another for further processing.
- In the **Documents** folder, execute the command ‘**ls -alh**’.
- Then, execute the command ‘**ls -alh | grep “folder*”**’.
- What can be seen here?
- The output of the **ls** command is redirected via a **pipe** to the **grep** which filtered the output to show only lines containing **folder*** in them.

Linux Terminal Introduction – pipe (|)

```
idnum@idnum-VirtualBox:~/Documents$ ls -alh /
```

```
total 473M
drwxr-xr-x 24 root root 4.0K Sep 18 14:27 .
drwxr-xr-x 24 root root 4.0K Sep 18 14:27 ..
drwxr-xr-x 2 root root 4.0K Sep 18 14:26 bin
drwxr-xr-x 3 root root 4.0K Sep 18 14:28 boot
drwxrwxr-x 2 root root 4.0K Sep 17 12:26 cdrom
drwxr-xr-x 18 root root 3.8K Sep 17 12:38 dev
```

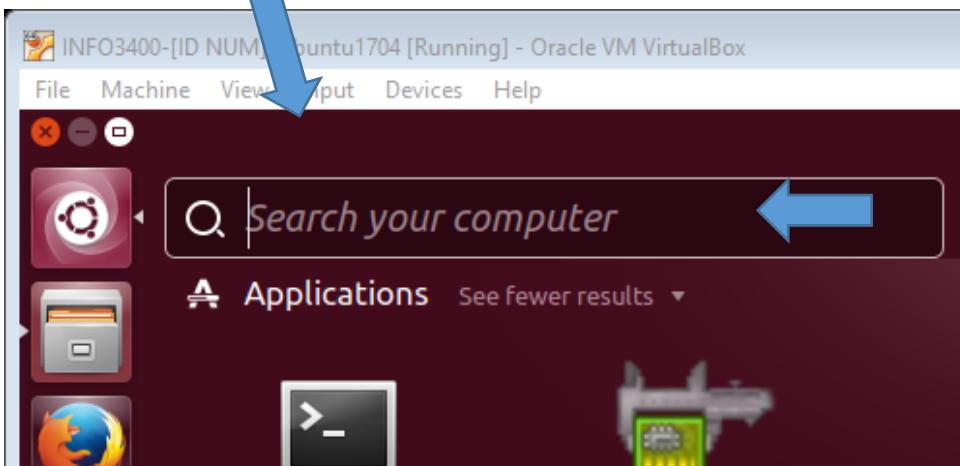
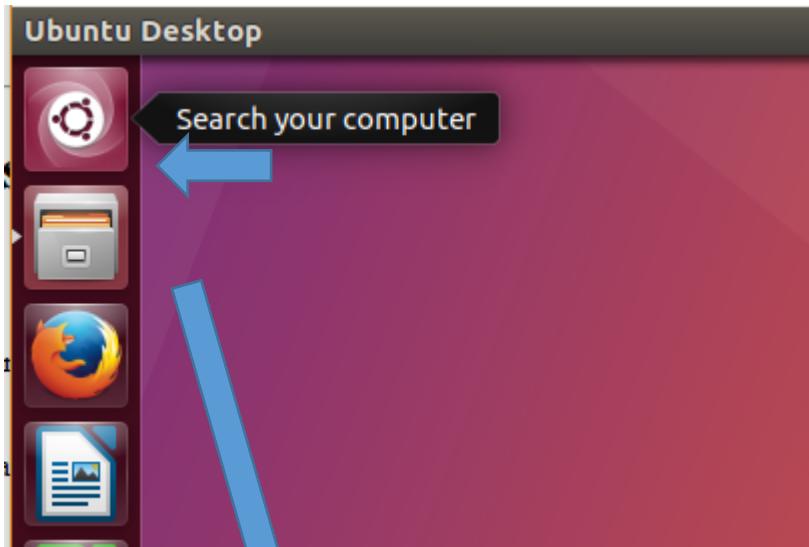


```
idnum@idnum-VirtualBox:~/Documents$ ls -alh / | grep "lrw"
```

```
lrwxrwxrwx 1 root root 33 Sep 18 14:27 initrd.img -> boot/initrd.img-4.10.0-35-generic
lrwxrwxrwx 1 root root 33 Sep 17 12:28 initrd.img.old -> boot/initrd.img-4.10.0-19-generic
lrwxrwxrwx 1 root root 30 Sep 18 14:27 vmlinuz -> boot/vmlinuz-4.10.0-35-generic
lrwxrwxrwx 1 root root 30 Sep 17 12:28 vmlinuz.old -> boot/vmlinuz-4.10.0-19-generic
idnum@idnum-VirtualBox:~/Documents$
```

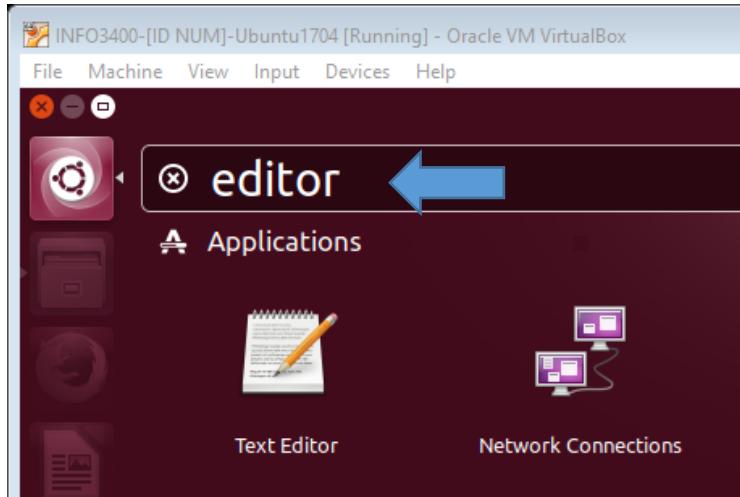
- Execute the command “ls - alh /”
- Then, execute the command ‘ls -alh / | grep “lrw”’.
- What can be seen?
- Only lines with “lrw” are filtered.
- The output of the ls command is “piped” to the grep command for processing.

Linux Terminal Introduction – cat

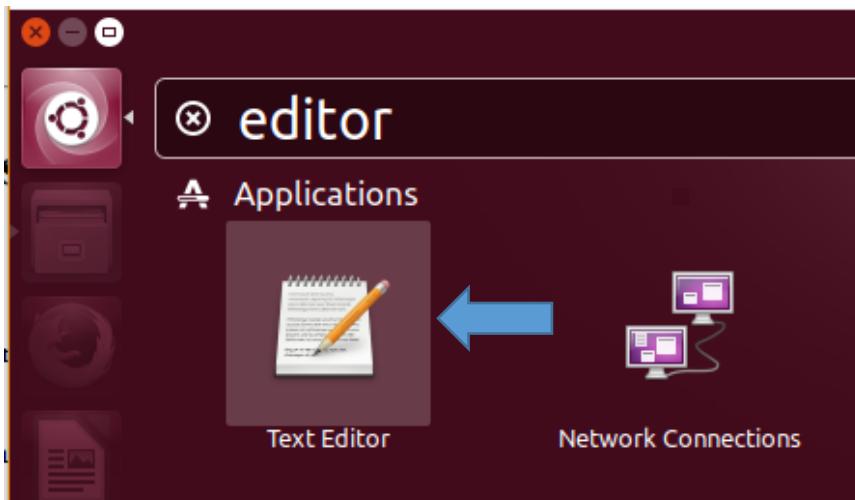


- The Linux **cat** command is similar to the windows **type** command.
- Open the Text editor,
- Click the Ubuntu “**Search your computer**” icon.

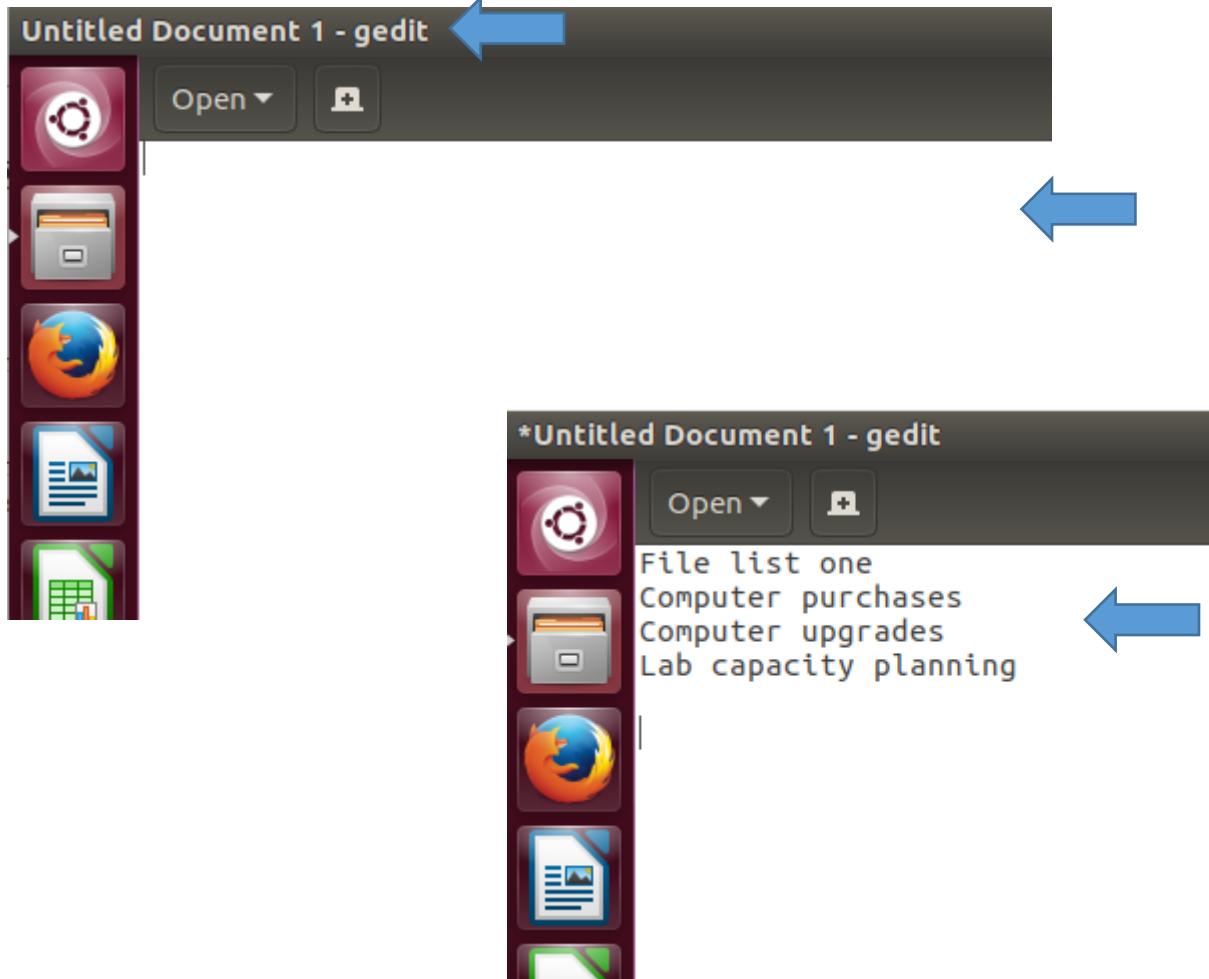
Linux Terminal Introduction – cat



- Type “editor” → click on “Text Editor” icon.

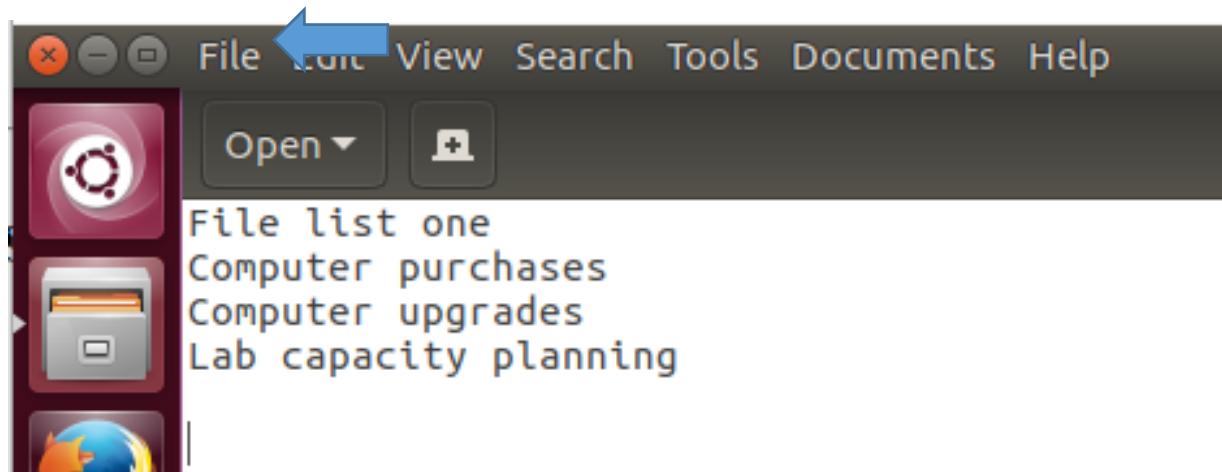
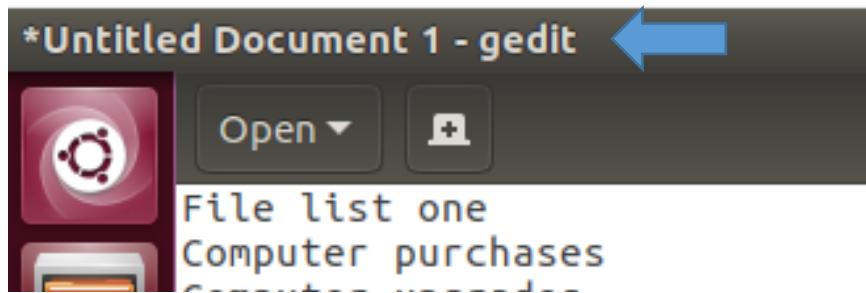


Linux Terminal Introduction – cat



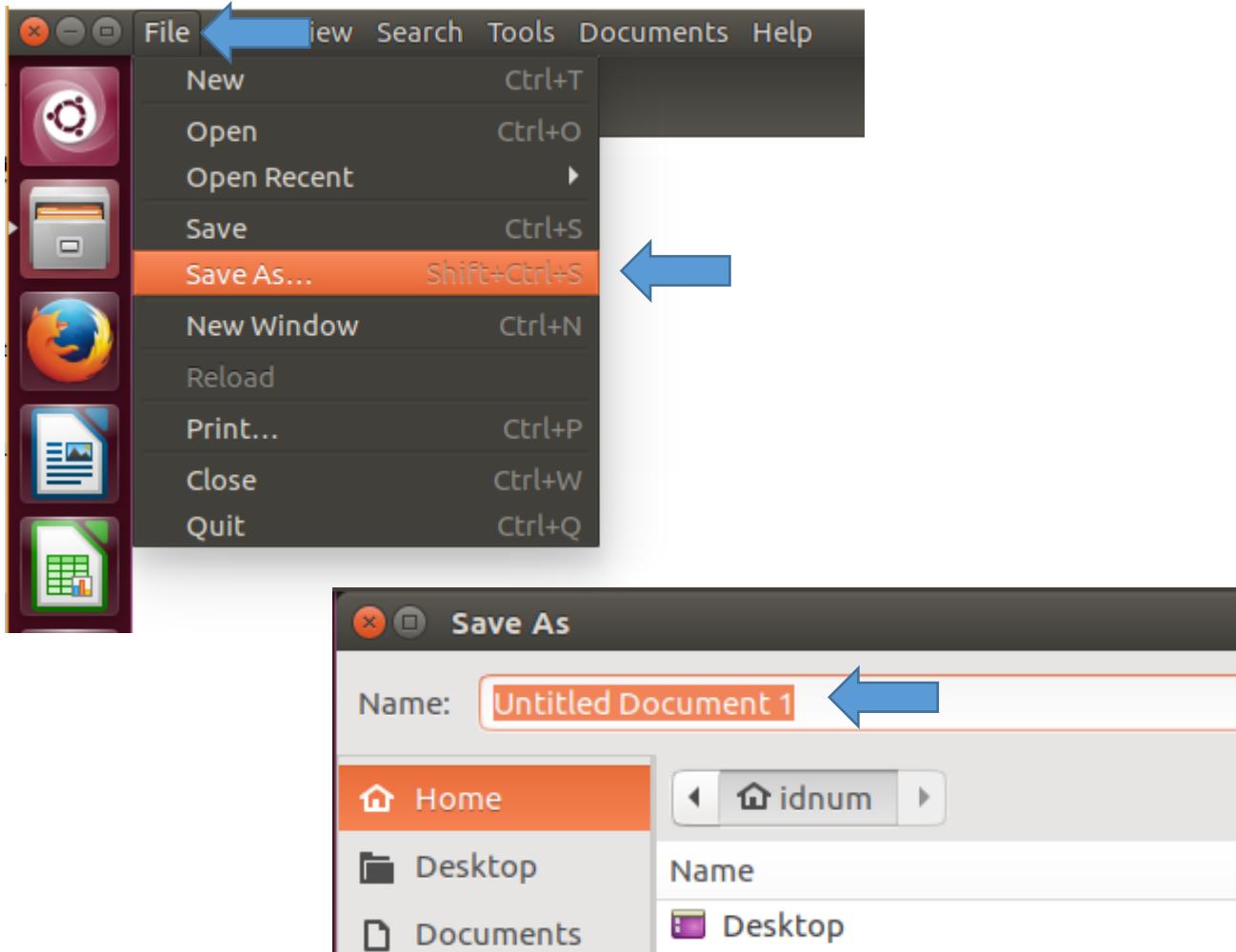
- Enter the following text:
 - File list one
 - Computer purchases
 - Computer upgrades
 - Lab capacity planning

Linux Terminal Introduction – cat



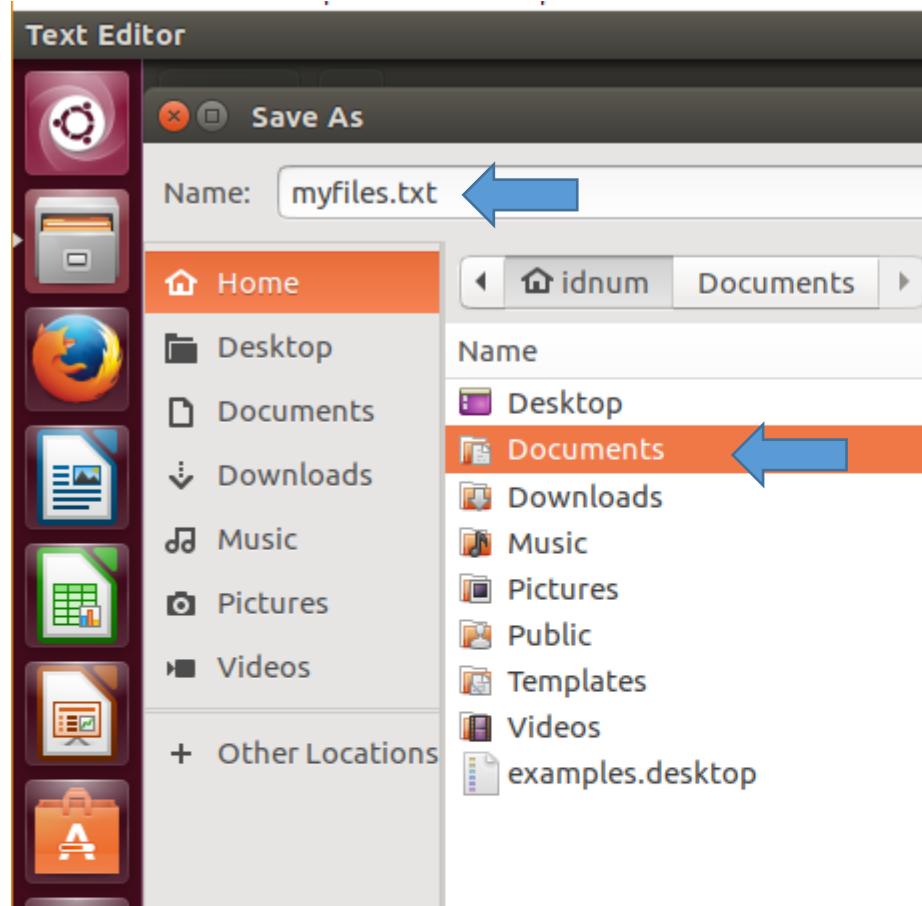
- Save the document as “**myfiles.txt**” in the **Documents** folder.
- Move the mouse over the “***Untitled Document 1 - gedit**” title bar.
- Click “File”.

Linux Terminal Introduction – cat



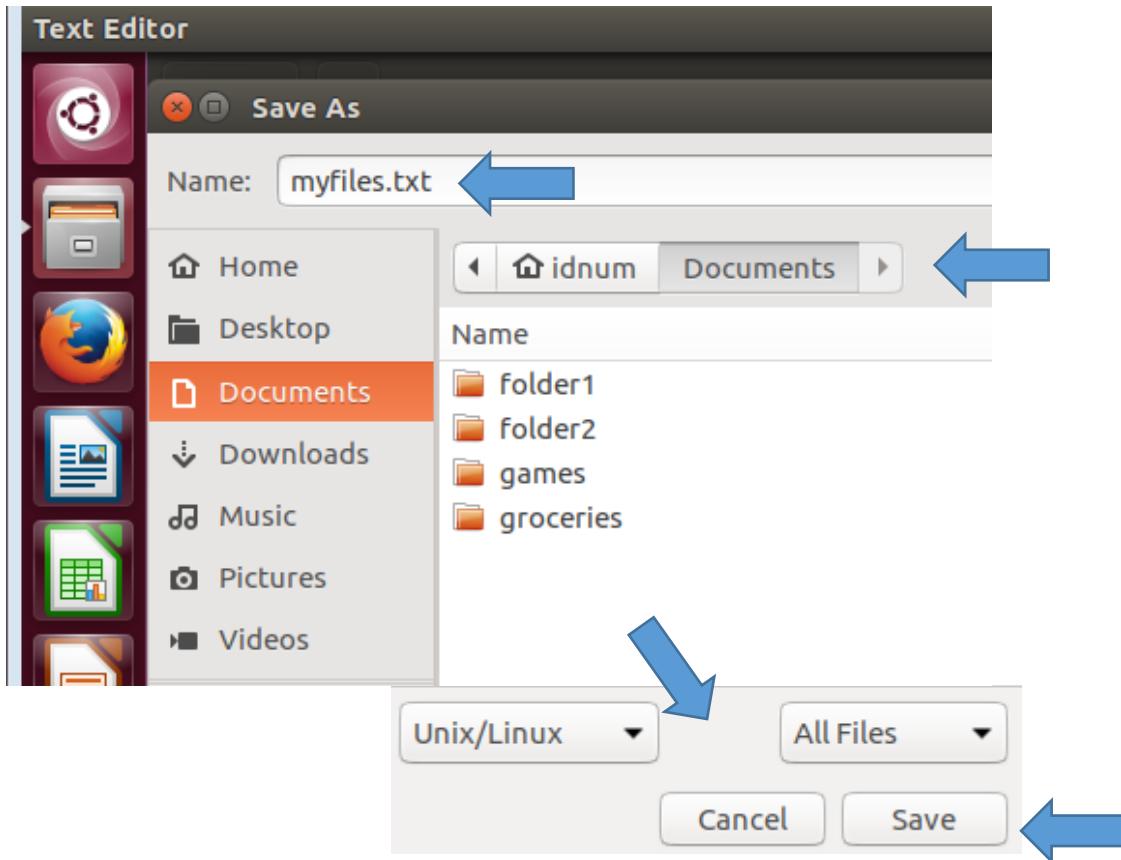
- Click “Save As...”.
- In the “Name:” textbox, change “Untitled Document 1” to “myfile.txt”.

Linux Terminal Introduction – cat



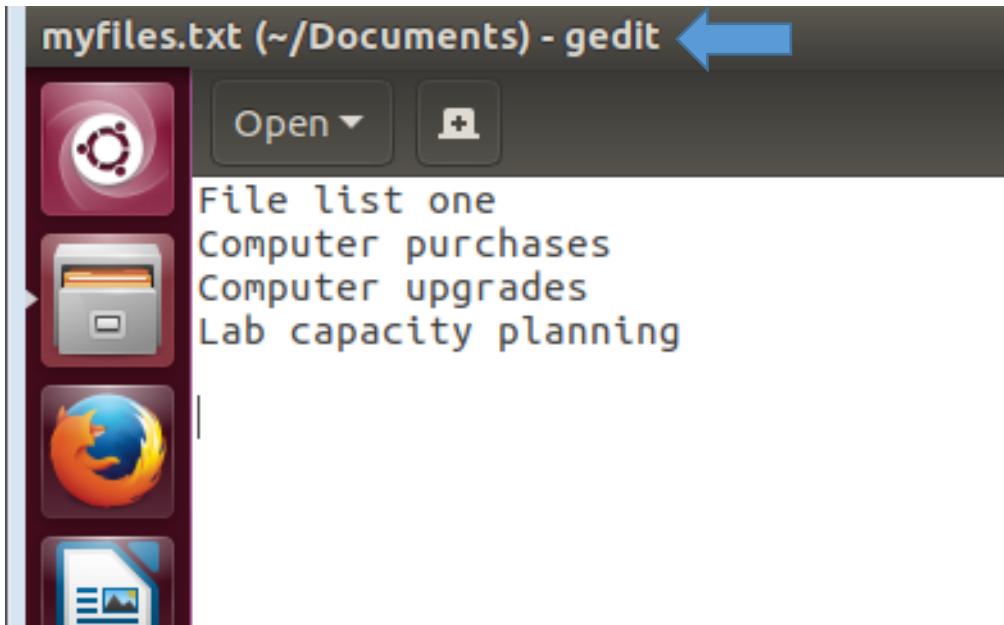
- Once the document is labelled to **myfiles.txt**, double click the “Documents” folder.

Linux Terminal Introduction – cat



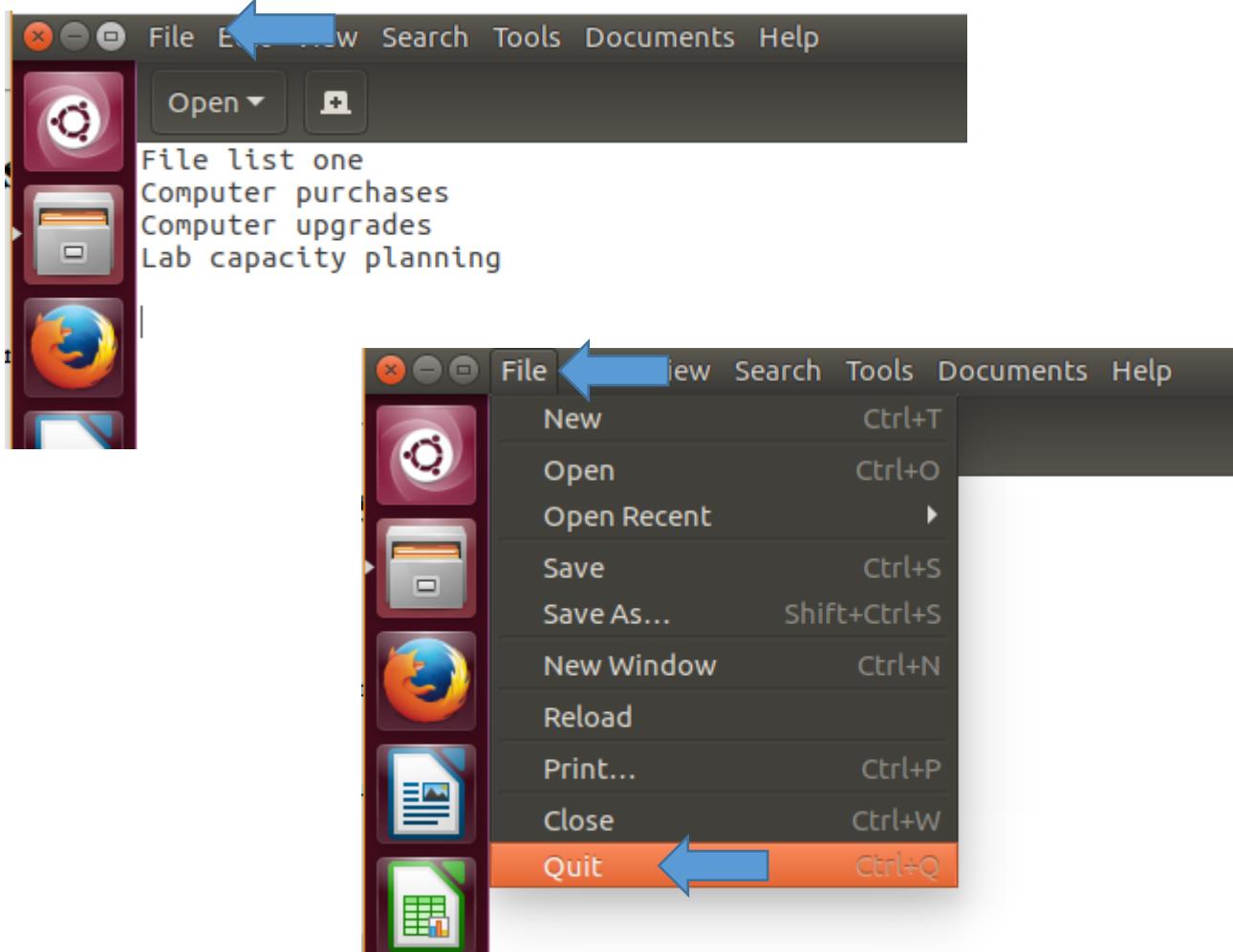
- Once in the **Documents** folder click the “Save” button at the lower right-hand corner of the window.

Linux Terminal Introduction – cat



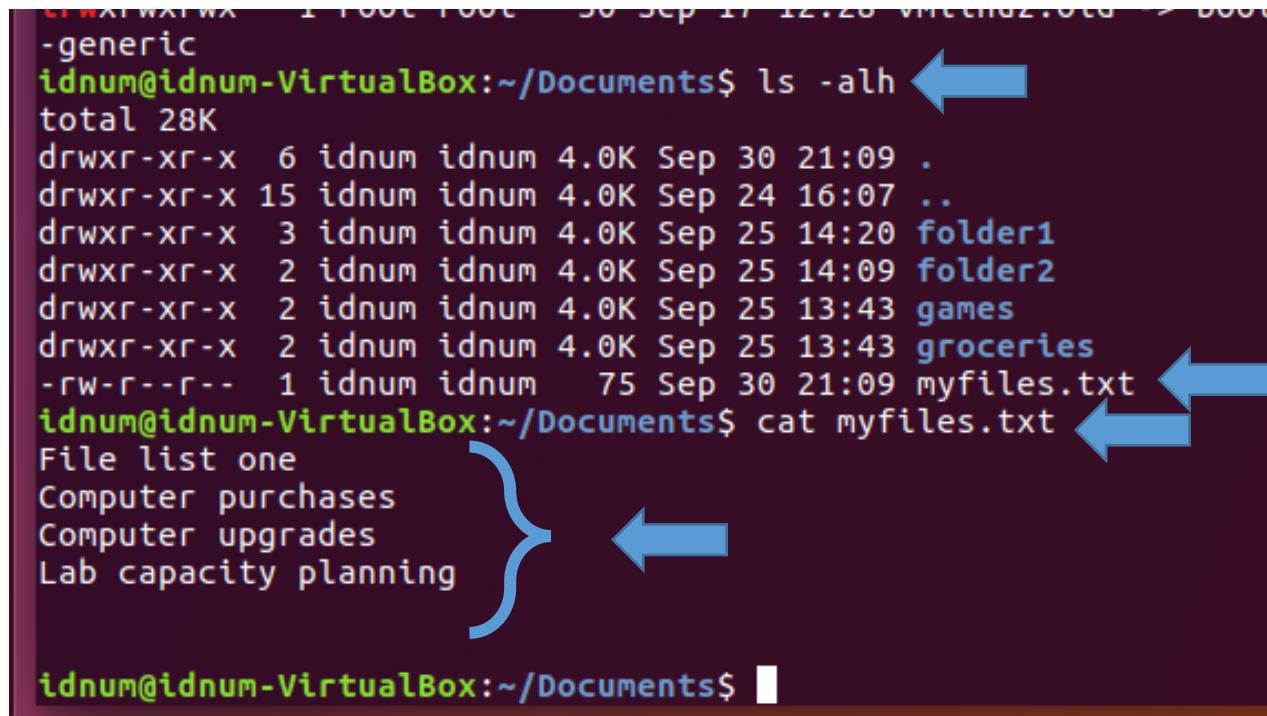
- The document should be saved as **myfiles.txt** in the **Documents** folder.
- Represented by “**myfiles.txt (~/Documents) - gedit**” in the title bar.

Linux Terminal Introduction – cat



- Close the application by moving the mouse over the title bar → click on “File” → then click on “Quit”.

Linux Terminal Introduction – cat



A screenshot of a Linux terminal window titled "idnum@idnum-VirtualBox:~/Documents\$". The terminal displays the output of the "ls -alh" command, showing a directory structure with files like ".", "..", "folder1", "folder2", "games", "groceries", and "myfiles.txt". Below this, the command "cat myfiles.txt" is run, and its output is shown: "File list one Computer purchases Computer upgrades Lab capacity planning". A blue curly brace on the left side groups the lines "Computer purchases", "Computer upgrades", and "Lab capacity planning". Three blue arrows point from the right side of the slide towards the terminal window: one arrow points to the "ls -alh" command, another points to the "cat myfiles.txt" command, and a third points to the grouped lines.

```
total 28K
drwxr-xr-x  6 idnum idnum 4.0K Sep 30 21:09 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum   75 Sep 30 21:09 myfiles.txt
File list one
Computer purchases
Computer upgrades
Lab capacity planning
```

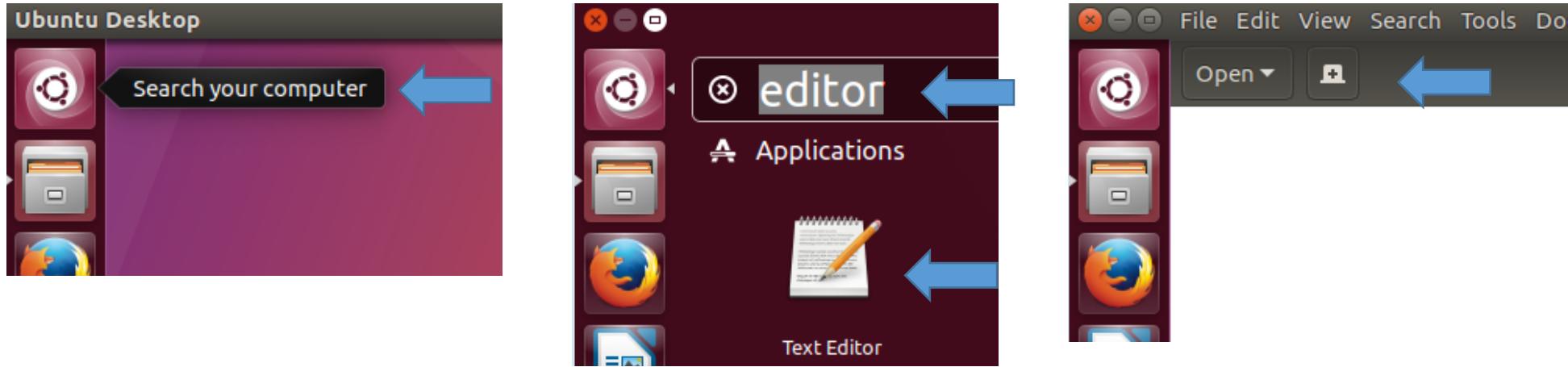
- Back at the terminal, execute the “ls -alh” command to make sure the file is there.
- Then execute the command “cat myfiles.txt”.
- What can be observed?
- How can this be useful?
- It is an easy way to see the contents of a relatively small text file without opening it in a GUI text editor.

Linux Terminal Introduction – cat

```
-rw-r--r-- 1 idnum idnum 75 Sep 30 21:09 myfiles.txt  
idnum@idnum-VirtualBox:~/Documents$ cat myfiles.txt  
File list one  
Computer purchases  
Computer upgrades  
Lab capacity planning  
  
idnum@idnum-VirtualBox:~/Documents$ cat myfiles.txt | grep "Computer" ←  
Computer purchases ←  
Computer upgrades  
idnum@idnum-VirtualBox:~/Documents$ █
```

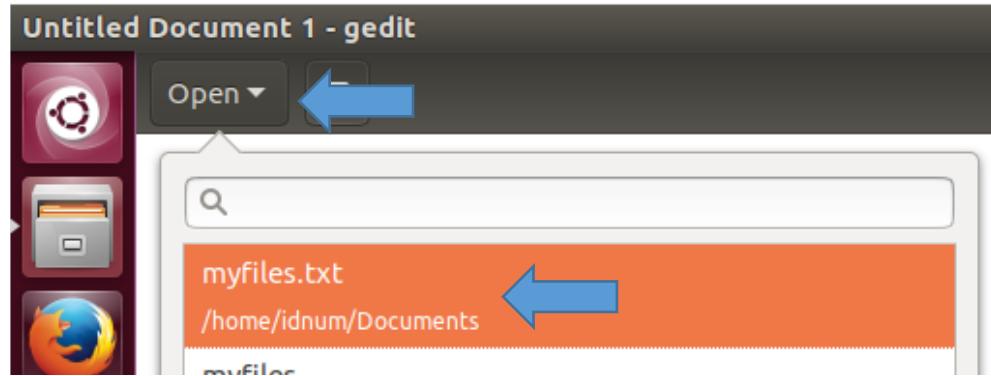
- Extending the functionality of the type command, it can be piped to find.
- Execute the command ‘**cat myfiles.txt | grep “Computer”**’.
- What happens here?

Linux Terminal Introduction – more

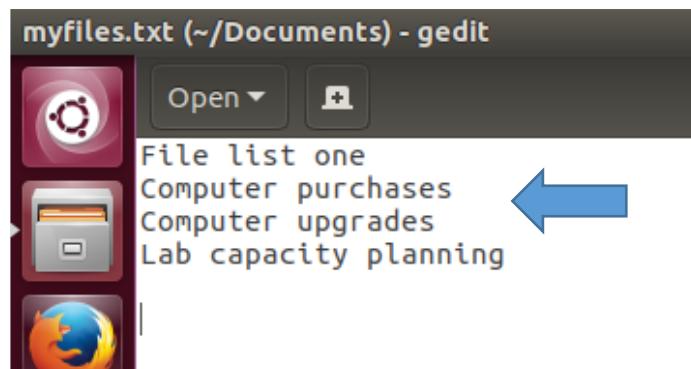


- Open the “myfiles.txt” file in the Text Editor.
- Click the Ubuntu “Search your computer” → type editor in the search field if it hasn't been done → click “Text Editor” icon.

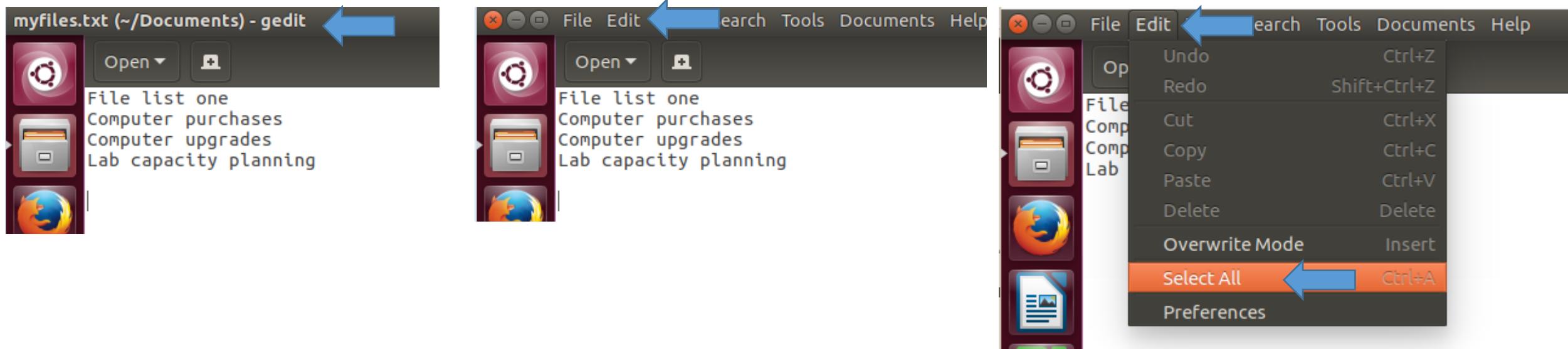
Linux Terminal Introduction – more



- Click “Open”.
- Double click on “**myfiles.txt**”.
- The file should be opened.

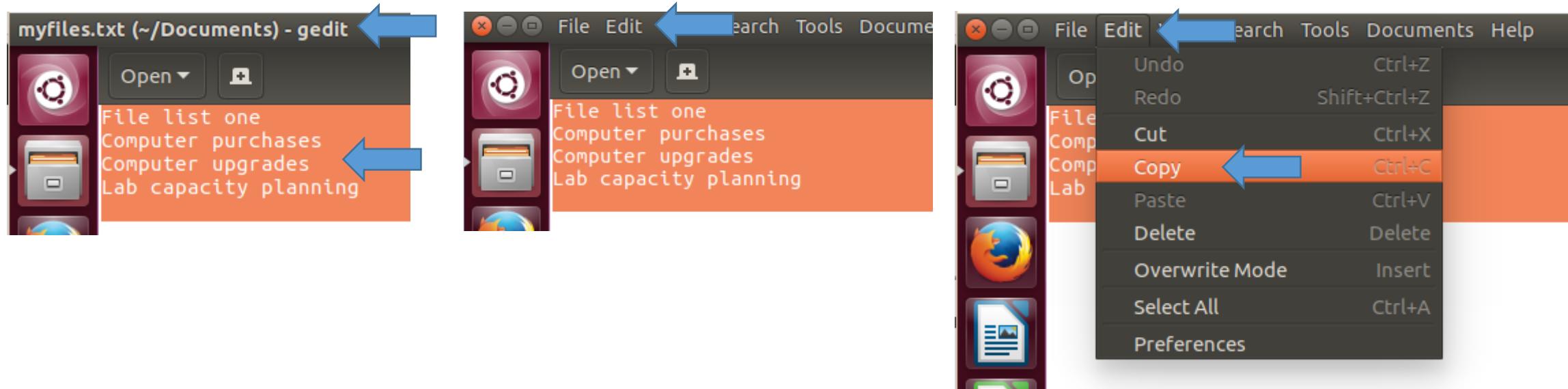


Linux Terminal Introduction – more



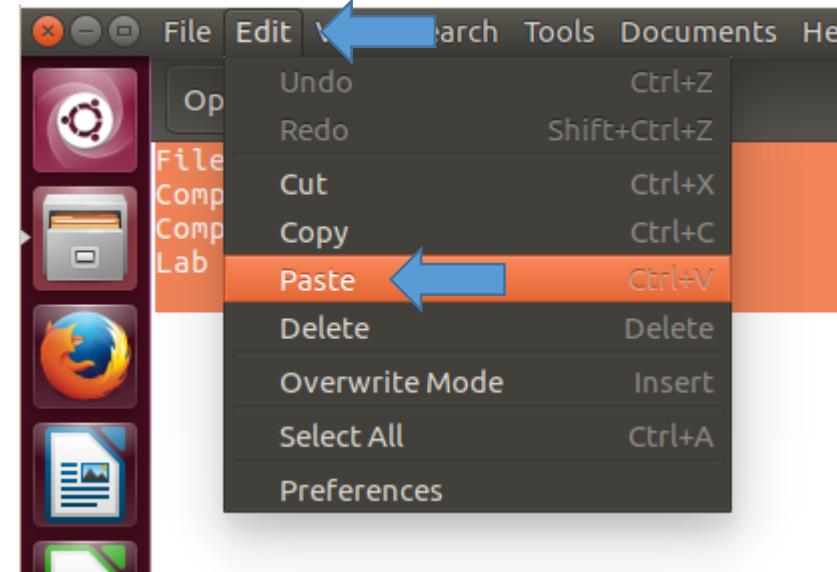
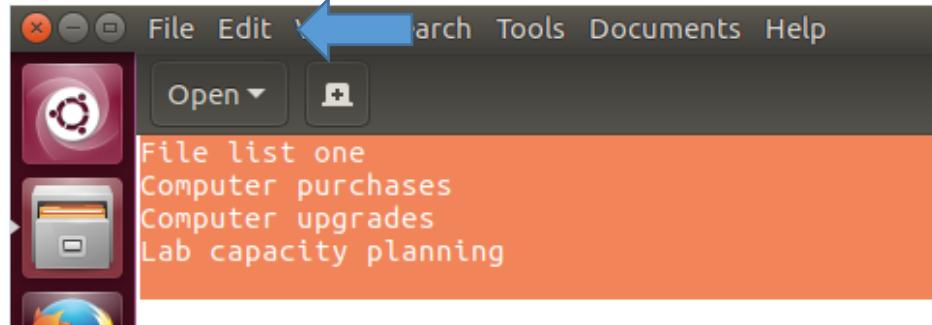
- Move your mouse over the title bar.
- Click “Edit”.
- Click “Select All”.

Linux Terminal Introduction – more



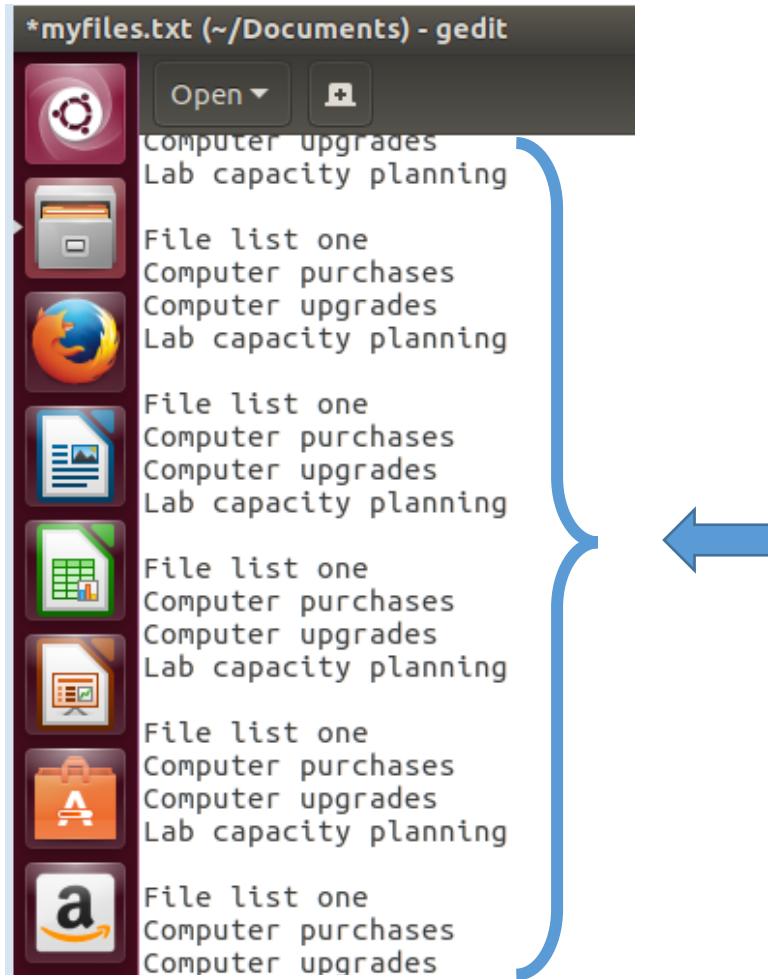
- Move your mouse over to the title bar.
- Click “Edit”.
- Click “Copy”.

Linux Terminal Introduction – more

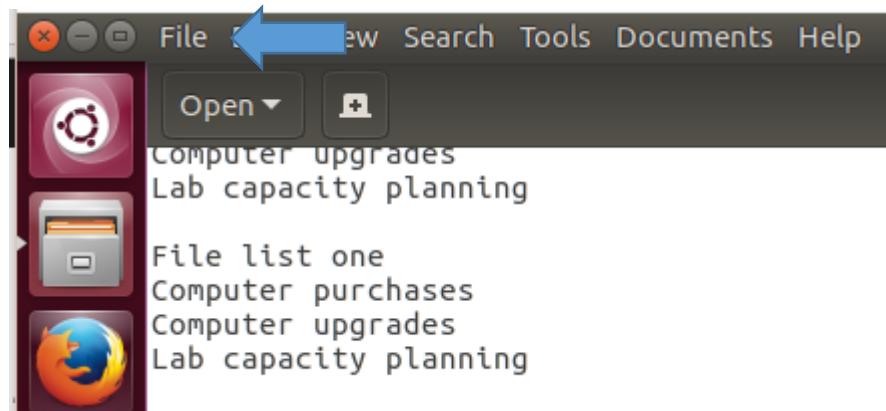


- Move your mouse over the title bar.
- Click “Edit”.
- Click “Paste”.

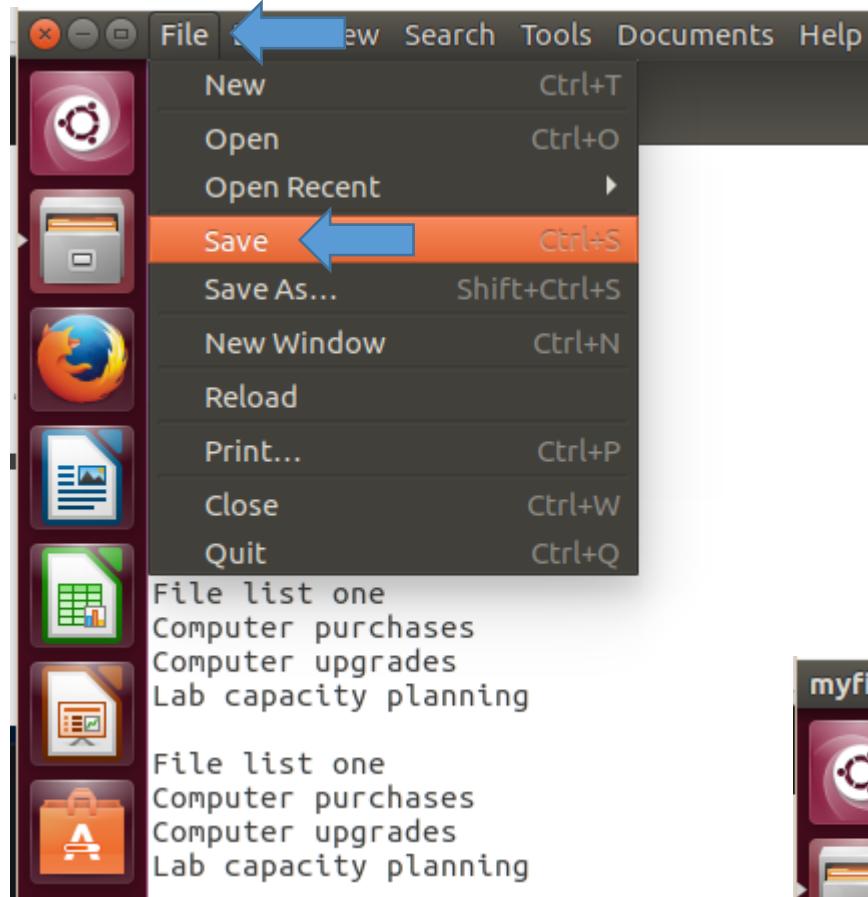
Linux Terminal Introduction – more



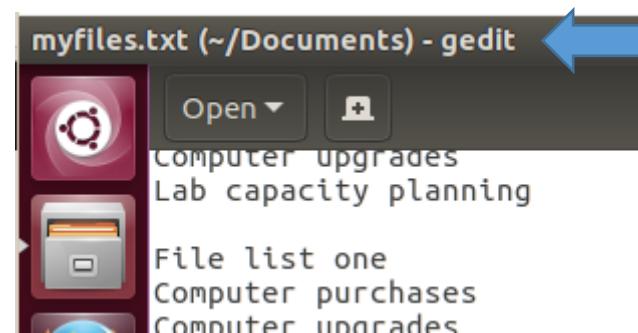
- Repeat the **Edit/Paste** operation at least 10 times.
- Save the file:
 - Move your mouse over the title bar.
 - Click “File”.



Linux Terminal Introduction – more

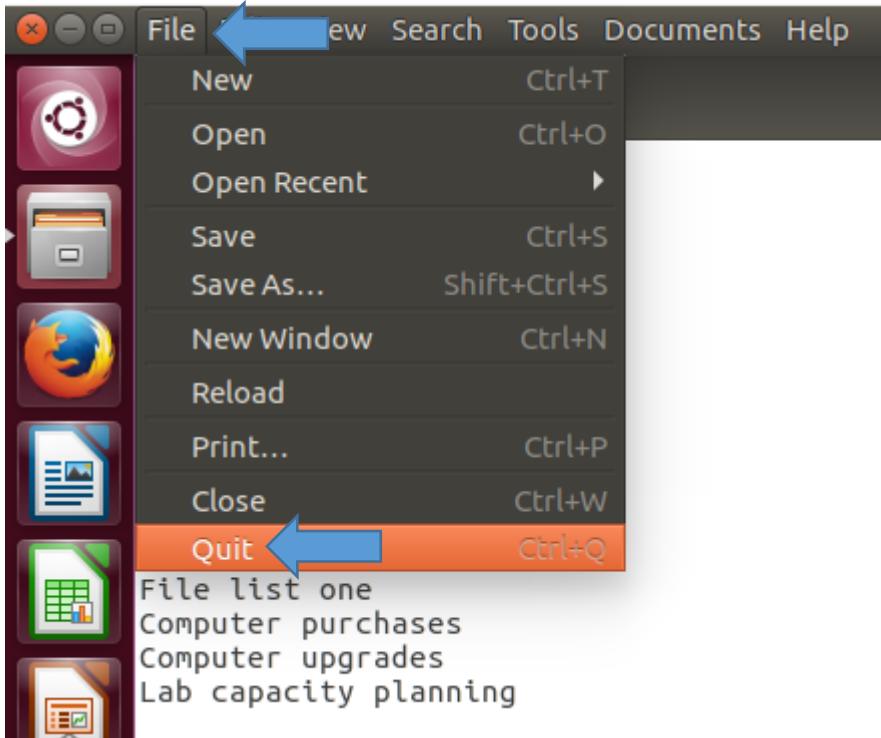


- Click “Save”.
- The updated document should be saved.
- Exit the application:
 - Move your mouse over the title bar.



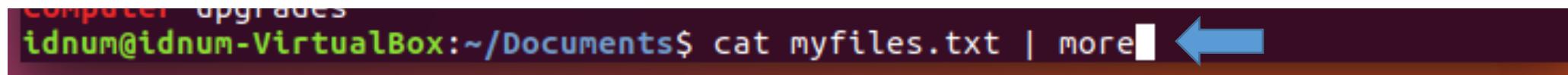
Linux Terminal Introduction – more

- Click “File”.
- Click “Quit”.

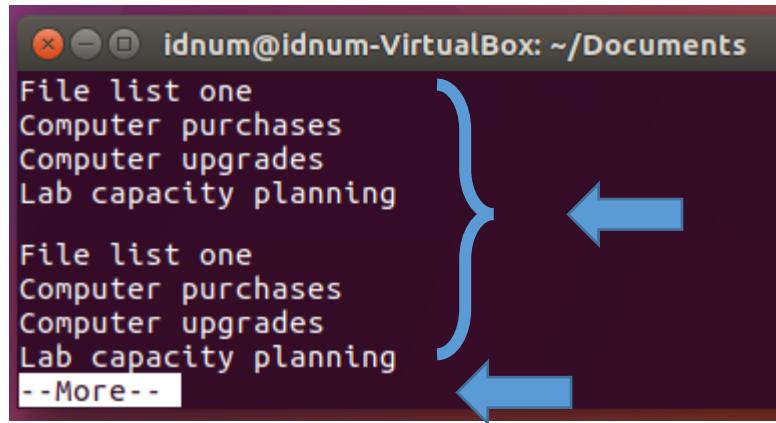


Linux Terminal Introduction – more

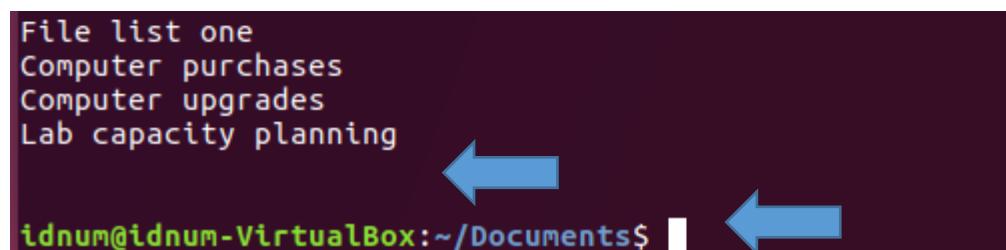
```
computer upgrades
idnum@idnum-VirtualBox:~/Documents$ cat myfiles.txt | more
```



```
idnum@idnum-VirtualBox:~/Documents
File list one
Computer purchases
Computer upgrades
Lab capacity planning
File list one
Computer purchases
Computer upgrades
Lab capacity planning
-- More --
```



```
File list one
Computer purchases
Computer upgrades
Lab capacity planning
idnum@idnum-VirtualBox:~/Documents$
```



- Back at the terminal, execute the command “**cat myfiles.txt | more**”.
- What happens?
- Press the **enter** key a few times.
 - What happens?
 - Scroll one line at a time.
- Press the **space bar**.
 - What happens?
 - Scroll one page at a time.

Linux Terminal Introduction – more

```
Computer upgrades  
Lab capacity planning
```

```
idnum@idnum-VirtualBox:~/Documents$ cat myfiles.txt | grep "o" | more
```

```
File list one  
Computer purchases  
Computer upgrades  
File list one  
Computer purchases  
- -More--
```

```
File list one  
Computer purchases  
Computer upgrades
```

2017-10-01 8:48:20 AM

- Once there is output to the console or even a file, you can perform “nested” pipe operations.
- Execute the command “**cat myfiles.txt | grep “o” | more**”.
- What happens?
- Press the **enter key** or **space bar** to complete listing all the text.
- Notice anything different?
- The lines containing “**Lab capacity planning**” is not shown.
 - Grep filtered all lines with “o”.

Linux Terminal Introduction – sleep

```
Computer purchases
Computer upgrades
idnum@idnum-VirtualBox:~/Documents$ sleep 5 ←
idnum@idnum-VirtualBox:~/Documents$ █
```

- Similar to the **timeout** command in windows the sleep command causes the terminal or script to “sleep” for a duration specified.
- Execute the command “**sleep 5**”.
- The console output will be blank or sleep for 5 seconds.

Linux Terminal Introduction – sleep

```
idnum@idnum-VirtualBox:~/Documents$ sleep 5
idnum@idnum-VirtualBox:~/Documents$ sleep --help ←
Usage: sleep NUMBER[SUFFIX]...
      or: sleep OPTION
Pause for NUMBER seconds. SUFFIX may be 's' for seconds (the default),
'm' for minutes, 'h' for hours or 'd' for days. Unlike most implementations
that require NUMBER be an integer, here NUMBER may be an arbitrary floating
point number. Given two or more arguments, pause for the amount of time
specified by the sum of their values.

--help      display this help and exit
--version   output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/sleep>
or available locally via: info '(coreutils) sleep invocation'
idnum@idnum-VirtualBox:~/Documents$
```

```
GNU coreutils online help: <http://www.gnu.org/software/
Full documentation at: <http://www.gnu.org/software/core
or available locally via: info '(coreutils) sleep invoca
idnum@idnum-VirtualBox:~/Documents$ sleep 5.6s ←
idnum@idnum-VirtualBox:~/Documents$
```

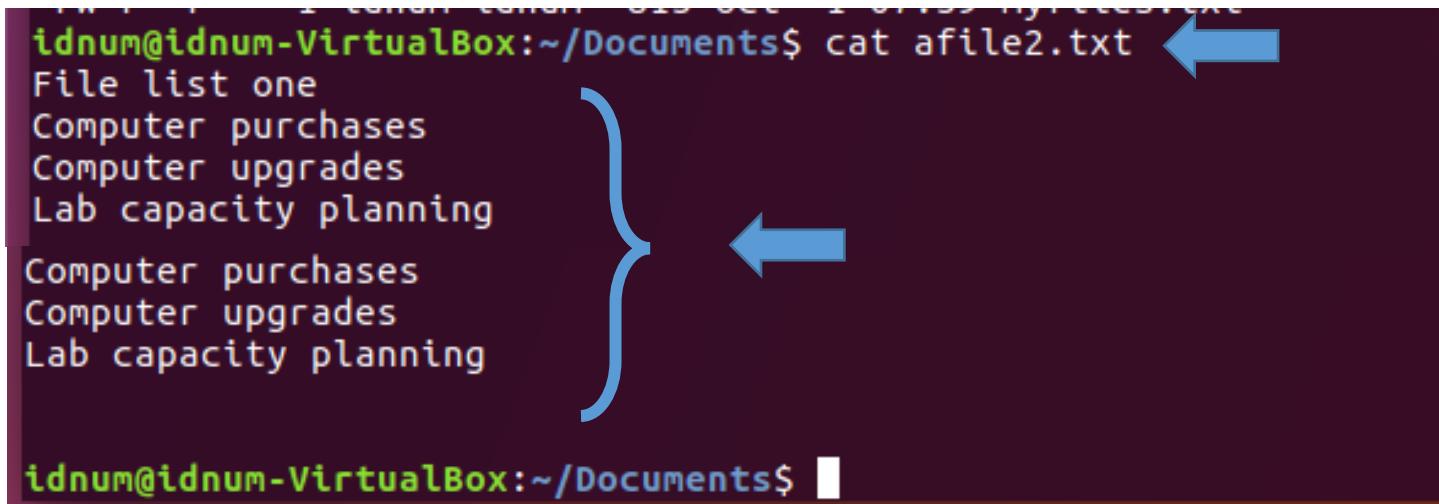
- Execute the command “sleep --help”.
- From the help command for sleep it has much more capabilities than the windows timeout command.
- You can specify a SUFFIX
 - e.g. s - seconds,
 - m - minutes,
 - h - hours,
 - d - days
- You can also specify a floating point number like 5.6 seconds or even 2.4 days.

Linux Terminal Output Redirection - >

```
idnum@idnum-VirtualBox:~/Documents$ cat myfiles.txt > afile2.txt ←
idnum@idnum-VirtualBox:~/Documents$ ls -alh ←
total 32K
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:15 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum  815 Oct  1 10:16 afile2.txt ←
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum     0 Aug  2 2016 myfile2
-rw-r--r--  1 idnum idnum  815 Oct  1 07:59 myfiles.txt
idnum@idnum-VirtualBox:~/Documents$
```

- Execute the command “**cat myfiles.txt > afile2.txt**”.
- What happened?
- Execute the command “**ls -alh**”.
- What do you see?
- A file called “**afile2.txt**” was created. It has the same size as “**myfiles.txt**”.
- Just like the windows cli, the output of the type command was redirected to a file using the “**>**” redirection arrow.

Linux Terminal Output Redirection - >



```
idnum@idnum-VirtualBox:~/Documents$ cat myfile2.txt
File list one
Computer purchases
Computer upgrades
Lab capacity planning
Computer purchases
Computer upgrades
Lab capacity planning

idnum@idnum-VirtualBox:~/Documents$
```

- View the contents of the **myfile2.txt** file.
- Execute the command “**cat myfile2.txt**”.
- It should have the same content as **myfile.txt**.

Linux Terminal Output Redirection - >

```
idnum@idnum-VirtualBox:~/Documents$ ls -alh > afile2.txt
idnum@idnum-VirtualBox:~/Documents$ cat afile2.txt
total 28K
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:15 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum    0 Oct  1 10:24 afile2.txt
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum    0 Aug  2 2016 myfile2
-rw-r--r--  1 idnum idnum  815 Oct  1 07:59 myfiles.txt
idnum@idnum-VirtualBox:~/Documents$
```

- Just like the windows cli, the “>” redirect creates a new file if it doesn’t exist but it always overwrites the contents.
- This has multiple uses especially in batch files, for instance you want to clear the contents of a file first before writing to it.

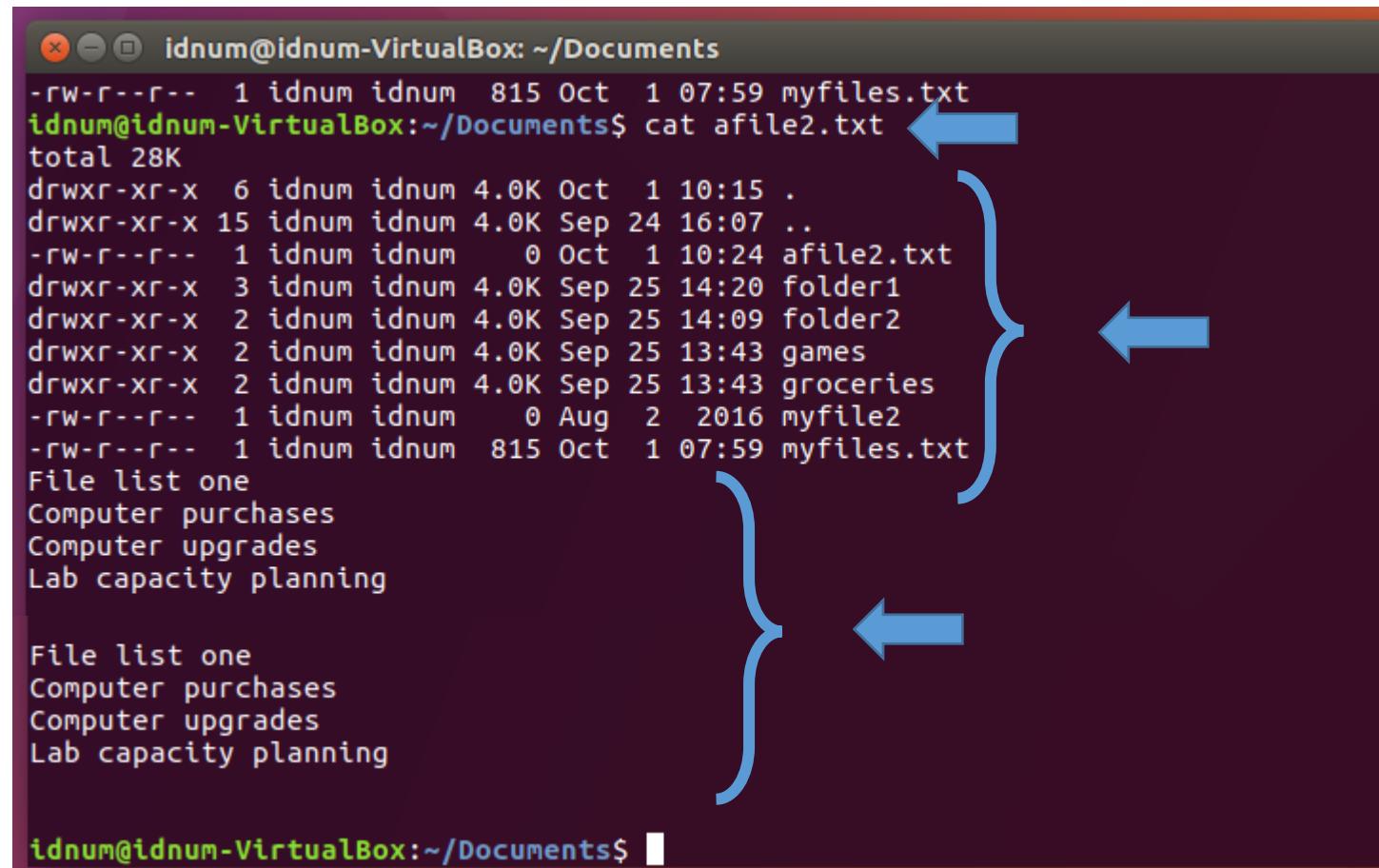
- Execute the command “ls -alh > afile2.txt”.
- What is the possible outcome here?
- Execute the command “cat afile2.txt”.
- What can be observed here?
- The output of the ls command was sent to the afile2.txt but it was overwritten.

Linux Terminal Output Redirection - >>

```
idnum@idnum-VirtualBox:~/Documents$ cat myfiles.txt >> afile2.txt ←
idnum@idnum-VirtualBox:~/Documents$ ls -alh ←
total 32K ←
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:15 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum 1.3K Oct  1 10:29 afile2.txt ←
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1 ←
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2 ←
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games ←
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries ←
-rw-r--r--  1 idnum idnum    0 Aug  2 2016 myfile2 ←
-rw-r--r--  1 idnum idnum 815 Oct  1 07:59 myfiles.txt ←
idnum@idnum-VirtualBox:~/Documents$
```

- Execute the command “**cat myfiles.txt >> afile2.txt**”.
- Execute the command “**ls -alh**”.
- What can be observed with the file sizes of **myfiles.txt** and **afile2.txt**.
- The file sizes are different.

Linux Terminal Output Redirection - >>



A screenshot of a Linux terminal window titled "idnum@idnum-VirtualBox: ~/Documents". The terminal displays the output of the command "cat afile2.txt". The output shows a file listing and the contents of "myfile2.txt". Blue arrows and curly braces highlight specific parts of the output: one arrow points to the command "cat afile2.txt", another points to the file listing, and two curly braces group the file listing and the contents of "myfile2.txt".

```
idnum@idnum-VirtualBox:~/Documents$ cat afile2.txt
total 28K
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:15 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum    0 Oct  1 10:24 afile2.txt
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum    0 Aug  2 2016 myfile2
-rw-r--r--  1 idnum idnum 815 Oct  1 07:59 myfile.txt
File list one
Computer purchases
Computer upgrades
Lab capacity planning

File list one
Computer purchases
Computer upgrades
Lab capacity planning

idnum@idnum-VirtualBox:~/Documents$
```

- Execute the command “**cat afile2.txt**”.
- What can be observed?
- You can see the output of the **ls** command and the contents of **myfile.txt**.
- The “**>>**” redirection appends data to the file.
- Also used in batch files but very good for keeping records or logs.

Linux Terminal Output Redirection - <

```
idnum@idnum-VirtualBox: ~/Documents
idnum@idnum-VirtualBox:~/Documents$ more < afile2.txt
total 28K
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:15 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum    0 Oct  1 10:24 afile2.txt
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
Computer upgrades
Lab capacity planning

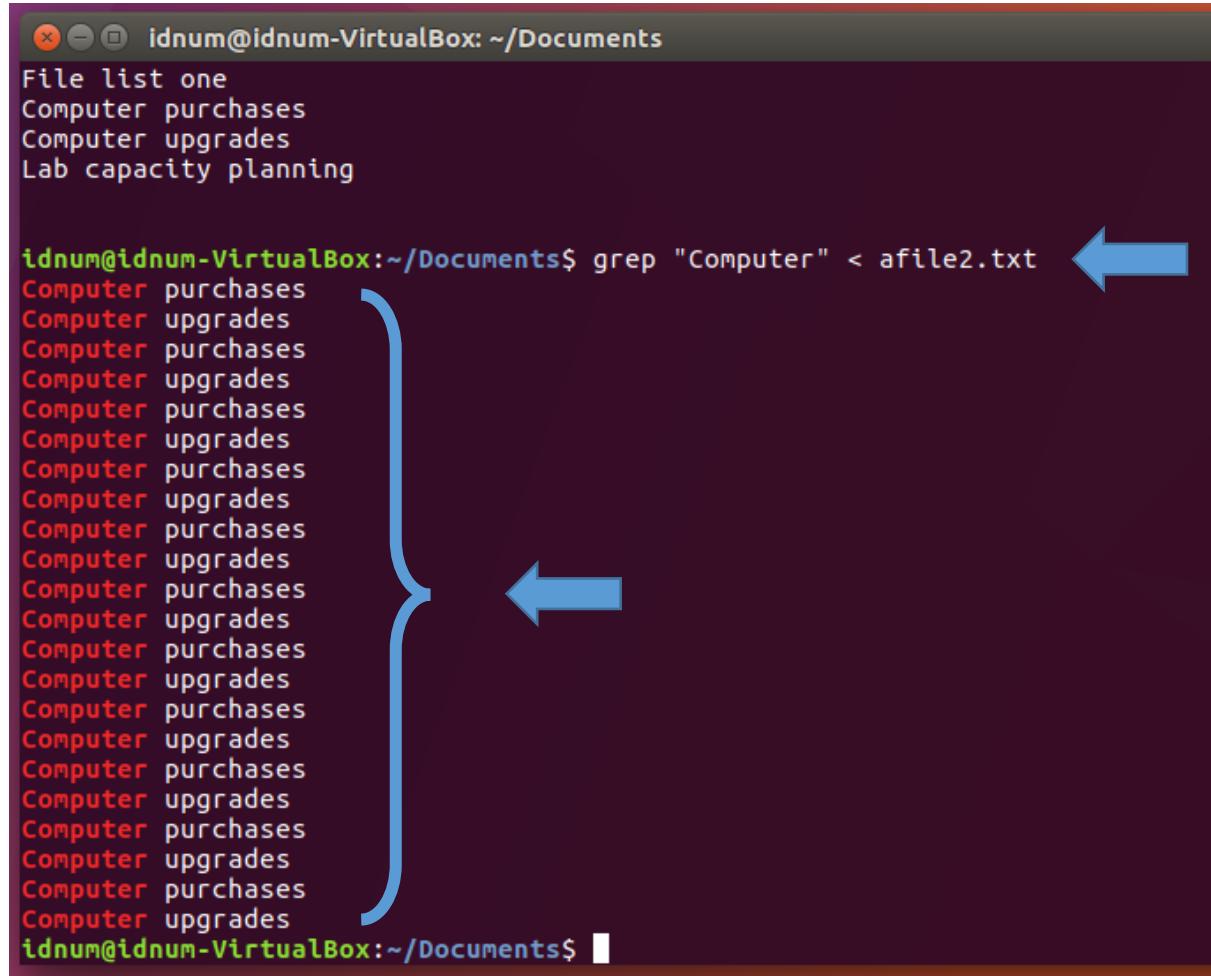
File list one
Computer purchases
Computer upgrades
Lab capacity planning
--More--
```

```
File list one
Computer purchases
Computer upgrades
Lab capacity planning

idnum@idnum-VirtualBox:~/Documents$
```

- Execute the command “**more < afile2.txt**”.
- What happens here?
- We are displaying the contents of the **afile2.txt** file one screen at a time using the **more** and **<** command combination.
- Press the spacebar a few times to display all the contents of the file.

Linux Terminal Output Redirection - <



```
idnum@idnum-VirtualBox: ~/Documents
File list one
Computer purchases
Computer upgrades
Lab capacity planning

idnum@idnum-VirtualBox:~/Documents$ grep "Computer" < afile2.txt
```

The terminal window shows the user's session. The user has run a command to list files in the current directory, which includes 'Computer purchases' and 'Computer upgrades'. Then, the user runs a command to search for the string 'Computer' in a file named 'afile2.txt'. The output of the file listing command is being redirected as input to the grep command. This is indicated by a brace on the left side of the terminal window grouping the first 15 lines of output, and two blue arrows pointing from this brace to the '<' symbol in the command line.

- Execute the command ‘grep “Computer” < afile2.txt’.
- What happens here?
 - NB: grep is case sensitive.
- The contents of **afile2.txt** was sent to the **grep** command and it filtered all lines with the string computer.
- The < can be used to take content from a file and send it as input to a program.

Linux Terminal Output Redirection - 2<&1

```
Computer upgrades
idnum@idnum-VirtualBox:~/Documents$ invalidcommand > afile3.txt ←
invalidcommand: command not found
idnum@idnum-VirtualBox:~/Documents$ ls -alh ←
total 32K ←
drwxr-xr-x 6 idnum idnum 4.0K Oct  1 10:47 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum 1.3K Oct  1 10:29 afile2.txt ←
-rw-r--r--  1 idnum idnum    0 Oct  1 10:47 afile3.txt ←
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum    0 Aug  2 2016 myfile2
-rw-r--r--  1 idnum idnum 815 Oct  1 07:59 myfiles.txt
idnum@idnum-VirtualBox:~/Documents$ █
```

- Execute the command “**invalidcommand > afile3.txt**”.
- What happens here?
- Execute the command “**ls -alh**”.
- The file **afile3.txt** was created but zero bytes in size.

Linux Terminal Output Redirection - 2<&1

```
idnum@idnum-VirtualBox:~/Documents$ invalidcommand > afile3.txt 2>&1
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 36K
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:47 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum 1.3K Oct  1 10:29 afile2.txt
-rw-r--r--  1 idnum idnum   34 Oct  1 10:50 afile3.txt
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum     0 Aug  2 2016 myfile2
-rw-r--r--  1 idnum idnum  815 Oct  1 07:59 myfiles.txt
idnum@idnum-VirtualBox:~/Documents$
```

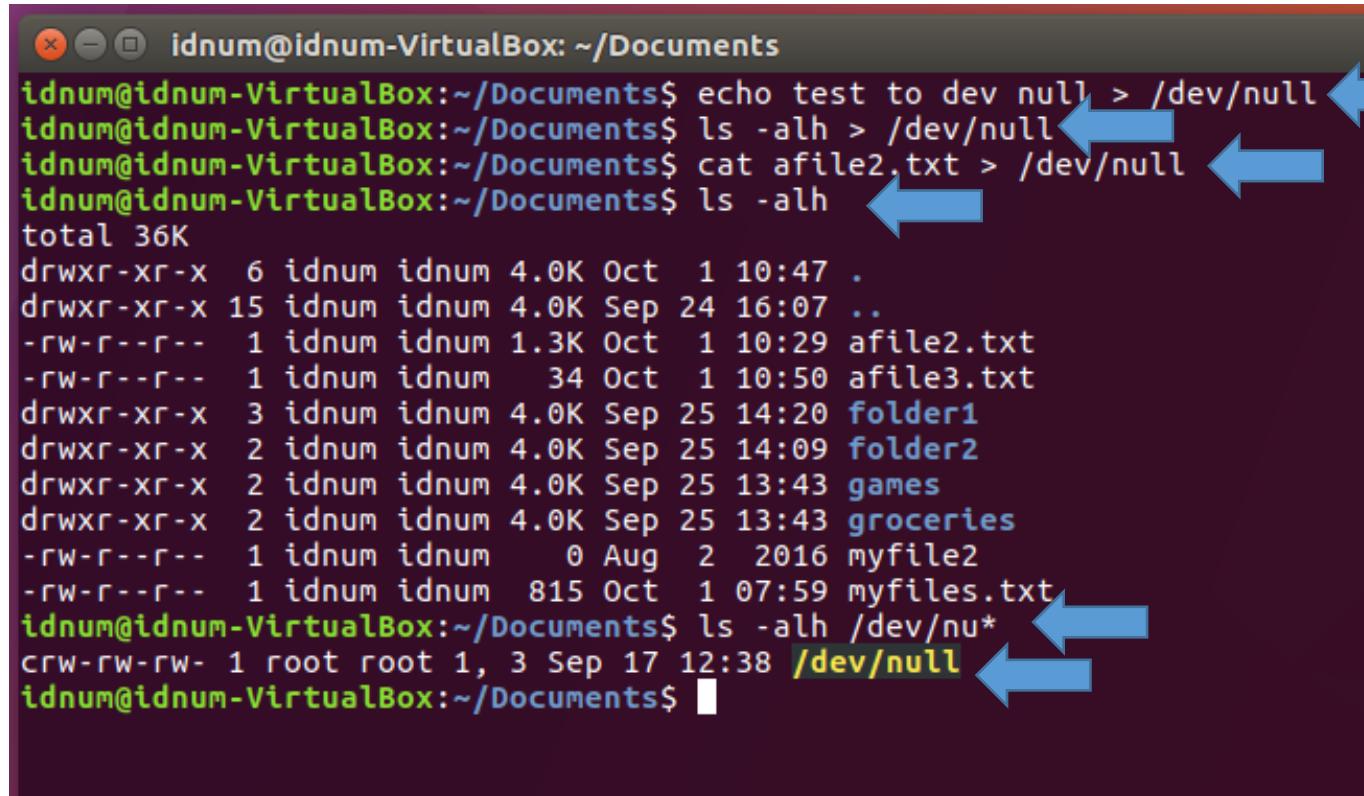
- Execute the command “**invalidcommand > afile3.txt 2>&1**”.
- What happens here?
- The error wasn’t displayed on the screen.
- Execute the command “**ls -alh**”.
- The file **afile3.txt** was created and has some content in it.

Linux Terminal Output Redirection - 2<&1

```
idnum@idnum-VirtualBox:~/Documents$ cat afile3.txt ←  
invalid command: command not found ←  
idnum@idnum-VirtualBox:~/Documents$ █
```

- Execute the command “**cat > afile3.txt**”.
- What happens here?
- The error was sent to the file **afile3.txt**.
- The **2>&1** redirect, redirects errors to a file.
- Where can this be used?

Linux Terminal Output Redirection - /dev/null



The screenshot shows a terminal window with the following session:

```
idnum@idnum-VirtualBox: ~/Documents
idnum@idnum-VirtualBox:~/Documents$ echo test to dev null > /dev/null
idnum@idnum-VirtualBox:~/Documents$ ls -alh > /dev/null
idnum@idnum-VirtualBox:~/Documents$ cat afile2.txt > /dev/null
idnum@idnum-VirtualBox:~/Documents$ ls -alh
total 36K
drwxr-xr-x  6 idnum idnum 4.0K Oct  1 10:47 .
drwxr-xr-x 15 idnum idnum 4.0K Sep 24 16:07 ..
-rw-r--r--  1 idnum idnum 1.3K Oct  1 10:29 afile2.txt
-rw-r--r--  1 idnum idnum   34 Oct  1 10:50 afile3.txt
drwxr-xr-x  3 idnum idnum 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 games
drwxr-xr-x  2 idnum idnum 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 idnum idnum    0 Aug  2 2016 myfile2
-rw-r--r--  1 idnum idnum  815 Oct  1 07:59 myfiles.txt
idnum@idnum-VirtualBox:~/Documents$ ls -alh /dev/nu*
crw-rw-rw-  1 root root 1, 3 Sep 17 12:38 /dev/null
idnum@idnum-VirtualBox:~/Documents$
```

Blue arrows point from the right margin to specific command lines in the terminal session, highlighting them for analysis.

- Execute the command “echo test to dev null > /dev/null”.
- What happened here?
- Execute the command “ls -alh > /dev/null”.
- Execute the command “cat afile2.txt > /dev/null”.
- Execute the command “ls -alh”.
- Is the /dev/null “file” present?
- Execute the command “ls -alh /dev/nu*”.
- Can you see the /dev/null file?
- Similar to the NUL device in the windows cli, the /dev/null is a device that discards all output that is sent to it. Very good if you do not want to display an error or output but don't want to record/log it either.

Linux Terminal Output Redirection - /dev/zero

```
lina@lina-VirtualBox:~/Documents$ echo test to dev zero > /dev/zero
lina@lina-VirtualBox:~/Documents$ ls -alh > /dev/zero
lina@lina-VirtualBox:~/Documents$ cat afile2.txt > /dev/zero
lina@lina-VirtualBox:~/Documents$ ls -alh
total 36K
drwxr-xr-x  6 lina lina 4.0K Oct  1 10:47 .
drwxr-xr-x 15 lina lina 4.0K Sep 24 16:07 ..
-rw-r--r--  1 lina lina 1.3K Oct  1 10:29 afile2.txt
-rw-r--r--  1 lina lina   34 Oct  1 10:50 afile3.txt
drwxr-xr-x  3 lina lina 4.0K Sep 25 14:20 folder1
drwxr-xr-x  2 lina lina 4.0K Sep 25 14:09 folder2
drwxr-xr-x  2 lina lina 4.0K Sep 25 13:43 games
drwxr-xr-x  2 lina lina 4.0K Sep 25 13:43 groceries
-rw-r--r--  1 lina lina     0 Aug  2 2016 myfile2
-rw-r--r--  1 lina lina  815 Oct  1 07:59 myfiles.txt
lina@lina-VirtualBox:~/Documents$ ls -alh /dev/ze*
crw-rw-rw-  1 root root 1, 5 Sep 17 12:38 /dev/zero
lina@lina-VirtualBox:~/Documents$
```

- The **/dev/zero** device is NOT the same as the **/dev/null** device.
- The **/dev/zero** outputs a stream of 0's while the **/dev/null** doesn't. We will look at this later on.

- Execute the command “echo test to dev zero > /dev/zero”.
- What happened here?
- Execute the command “ls -alh > /dev/zero”.
- Execute the command “cat afile2.txt > /dev/zero”.
- Execute the command “ls -alh”.
- Is the /dev/zero “file” present?
- Execute the command “ls -alh /dev/ze*”.
- Can you see the /dev/zero file?

Further Reading

- <http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html>
- <https://www.tecmint.com/linux-directory-structure-and-important-files-paths-explained/>
- <http://www.linuxnix.com/absolute-path-vs-relative-path-in-linuxunix/>
- <https://www.lifewire.com/uses-of-linux-command-find-2201100?i10c.encReferrer=aHR0cHM6Ly93d3cuZ29vZ2xlLnR0Lw%3D%3D>
- <http://www.binarytides.com/linux-find-command-examples/>

Further Reading

- http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_04_02.html
- <https://ss64.com/bash/grep.html>
- <http://www.linfo.org/pipes.html>
- <https://ss64.com/bash/touch.html>
- <http://www.linfo.org/touch.html>
- <https://stackoverflow.com/questions/876239/how-can-i-redirect-and-append-both-stdout-and-stderr-to-a-file-with-bash>
- <https://unix.stackexchange.com/questions/254384/difference-between-dev-null-and-dev-zero>