Linux 101 Workshop

What’s In It For Me

(WIIFM)

- Linux is a basic requirement
- Our application is running on Linux

Linux is a family of free and open-source operating systems build around the Linux kernel

Distributions

(in short - distro)

- Debian
- Ubuntu
- Fedora
- Red Hat
- CentOS
- SuSE
- Android (?)

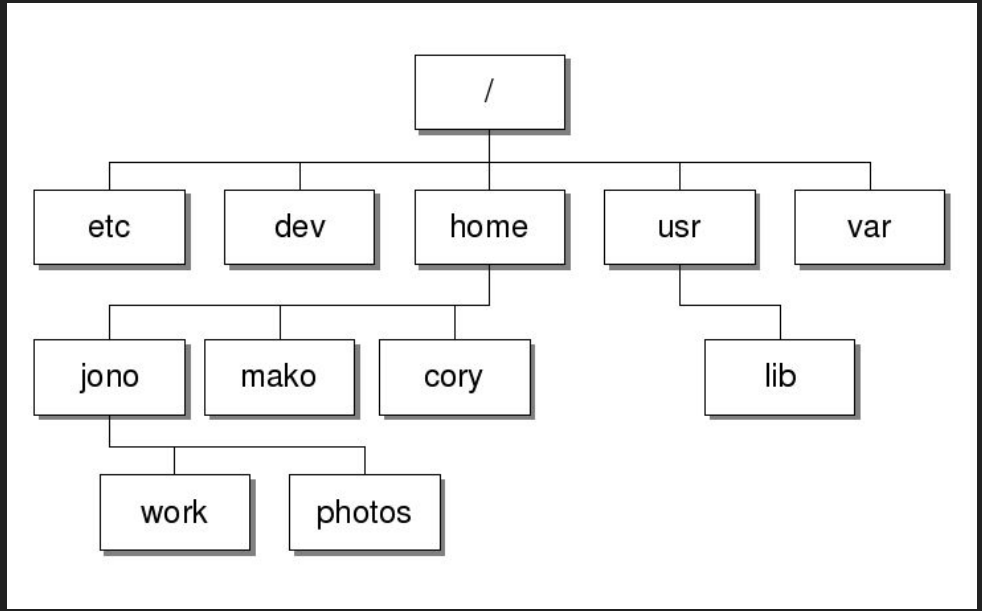
Agenda

- Filesystem
- Path
- High Level Structure
- Commands
- I/O Redirections

Filesystem

Represents data on storage devices as files


10101010101010010100101001100110101
/
bin
etc
home
user
notes
src


/
etc
dev
home
usr
var
jono
mako
cory
lib
work
photos

Path

/

Slash, the root directory

Absolute Path

Points to the same location regardless of the current directory

```
/var/ftp/pub
/etc/samba.smb.conf
/boot/grub/grub.conf
```

Relative Path

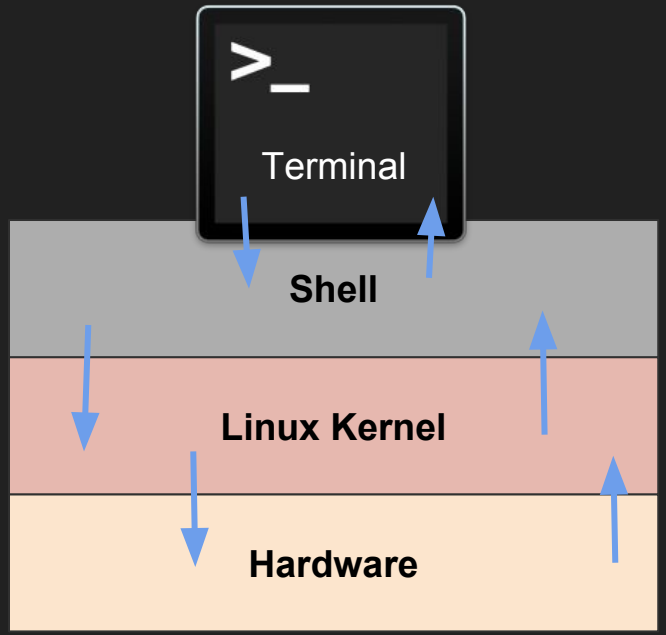
Points to the same location relatively to the current working directory

```
../../tmp
./pub
documents/private/../public
```

Type of files

- “Everything is a file” in Linux
  - Regular
  - Directory
  - Block
  - Link
  - Socket
- In every directory you’ll find two hidden files:
  - ..
  - .

High Level Structure



Commands

First command: pwd

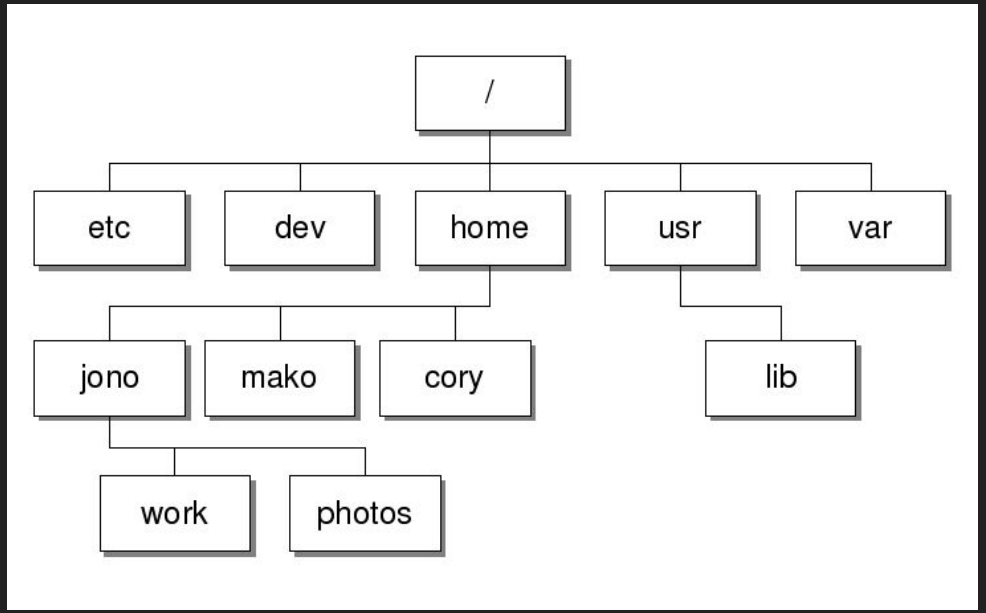
- Print working/current directory
- Answers the question “Where Am I?”

```
[arie@localhost ~]$ pwd

/home/arie
```

3 Directories:

- /
- home
- arie



ls

- List directory contents and information

- Useful options
  - -a for hidden files
  - -l for long listing format
  - -F -> can you guess what it does?

```
[arie@localhost ~]$ ls

pictures docs file
```

```
[arie@localhost ~]$ ls /tmp

a b c

# How do I know if they are directories, regular files or perhaps some are directories and some are regular?
```

Command <Options> <Arguments>

cd

- Change the current directory to another dir

```
[arie@localhost ~]$ cd /tmp
```

```
[arie@localhost ~]$ cd $HOME
[arie@localhost ~]$ cd ~
```

- Try changing directory to /tmp and type ‘pwd’. What do you see?

```
[arie@localhost ~]$ cd ..
```

You can see all shell variables by running ‘set’

```
[arie@localhost ~]$ cd -
```

mkdir

- Create a directory
- Useful option: -p for creating multiple directories at once

```
[arie@localhost ~]$ mkdir new
```

```
[arie@localhost ~]$ mkdir /tmp/new
```

- After creating a new directory, run ‘ls -lF’
- Try also ‘ls -ld new’

```
[arie@localhost ~]$ mkdir /tmp/a/b

mkdir: cannot create directory ‘/tmp/a/b’: Not a directory
```

```
[arie@localhost ~]$ mkdir -p /tmp/a/b
```

touch

- Create files and modify their timestamp

```
[arie@localhost ~]$ touch new_file
```

```
[arie@localhost ~]$ touch a b
```

- After creating a new file, try to ‘touch’ it again. Can you see its timestamp changed?

Opening Closes

- Questions so far?
  - How many commands left to learn?
    - 2000 installed, more than 10,000,000 uninstalled
  - I don’t like typing. Is there a way I can use only my mouse?
    - Get out of the room
  - How am I supposed to remember all the commands?
    - MORE practice

Closing Closes

Exercise

- Go to /tmp directory and create there a directory called pizza
- In your new directory, create 3 files named after toppings you like on your pizza
  - If you don’t like pizza, get out of the room
- Verify they have been created (in pizza directory)
- Go back to your home directory

Did you know? Every time you’ll reboot your operating system, the files and directories in /tmp will be removed!

Commands we learned so far:

pwd - current location
cd - change directory
ls - list content
mkdir - create directory
touch - create empty file

Solution

```
[arie@localhost ~]$ cd /tmp
[arie@localhost ~]$ mkdir pizza
[arie@localhost ~]$ cd pizza
[arie@localhost ~]$ touch mushrooms mozzarella tomatoes
[arie@localhost ~]$ ls
mushrooms mozzarella tomatoes
[arie@localhost ~]$ cd ~
```

echo

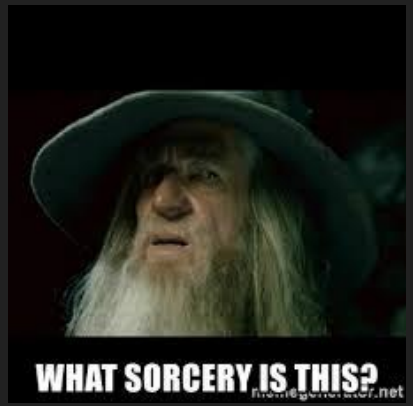
- Display a line of text

```
[arie@localhost ~]$ echo “Look mom, I know Linux!”

Look mom, I know Linux
```

- You can create a new file using echo

```
[arie@localhost ~]$ echo “Amazing” > new_file
```



I/O Redirection

- Manipulation of I/O
- Three streams
  - stdin (Input) -> 0
  - stdout (Output) -> 1
  - stderr (Error) -> 2

```
[arie@localhost ~]$ blabla
blabla: command not found...
[arie@localhost ~]$ blabla 2> error_info
```

```
[arie@localhost ~]$ echo 1 > my_echos
[arie@localhost ~]$ echo 2 > my_echos
[arie@localhost ~]$ cat my_echos
2
```

- > for output redirection
- < for input redirection
- 2> for error
- Use ‘>>’ for appending
- 2>&1 redirect error to output target

```
[arie@localhost ~]$ echo 1 > my_echos
[arie@localhost ~]$ echo 2 >> my_echos
[arie@localhost ~]$ cat my_echos
1
2
```

cat

- Concatenate files.
- Probably most common use: read files :)
- You can also create files with content

```
[arie@localhost ~]$ cat /etc/passwd
```

```
[arie@localhost ~]$ echo “a” > file1
[arie@localhost ~]$ echo “b” > file2
[arie@localhost ~]$ cat file1 file2
a
b
```

Did you know? /etc/passwd used for storing data on all users in the system and their information (e.g. home directory, shell, …)

```
[arie@localhost ~]$ echo > file3
Hello
there
(Press ctrl+d)
[arie@localhost ~]$ cat file3
Hello
there
```

mv

- Move files
- Also rename files

```
[arie@localhost ~]$ touch new
[arie@localhost ~]$ mv new /tmp/
[arie@localhost ~]$ ls /tmp
new
```

```
[arie@localhost ~]$ touch x
[arie@localhost ~]$ ls
x
[arie@localhost ~]$ mv x y
[arie@localhost ~]$ ls
y
```

Exercise 2

- Create a file called kit
- Move the file to /tmp and rename its name to kat
- Add the line “I was kit, now I’m kat” to the file
- Read the file to make sure the line is there
- Now add another line to the file “I’m complete”

New commands we learned:

cat - read file
mv - move and rename
echo - print a line

Solution

```
[arie@localhost ~]$ touch kit
[arie@localhost ~]$ mv kit /tmp/kat
[arie@localhost ~]$ echo “I was kit, now I’m kat” > /tmp/kat
[arie@localhost ~]$ cat /tmp/kat
I was kit, now I’m kat
[arie@localhost ~]$ echo “I’m complete” >> /tmp/kat
```

rm

- Remove files or directories

```
[arie@localhost ~]$ touch x
[arie@localhost ~]$ ls
x
[arie@localhost ~]$ rm x
[arie@localhost ~]$ ls x
ls: cannot access 'x': No such file or directory
```

- Useful options
  - -r for removing directory its content
  - -i -> prompt before every removal

```
[arie@localhost ~]$ mkdir directory
[arie@localhost ~]$ rm directory
rm: cannot remove 'directory': Is a directory
[arie@localhost ~]$ rm -r directory
[arie@localhost ~]$ ls
```

cp

- copy files and directories

```
[arie@localhost ~]$ touch x
[arie@localhost ~]$ cp x y
[arie@localhost ~]$ ls
x y
```

- Useful options
  - -r -> like with rm, for copying an entire directory and its content

```
[arie@localhost ~]$ mkdir directory
[arie@localhost ~]$ touch directory/y
[arie@localhost ~]$ cp -r directory /tmp/
```

man

- Manual references
- Try it for yourself:
  - man pwd
  - man ls
  - man rm
  - ...

You can even run ‘man man’