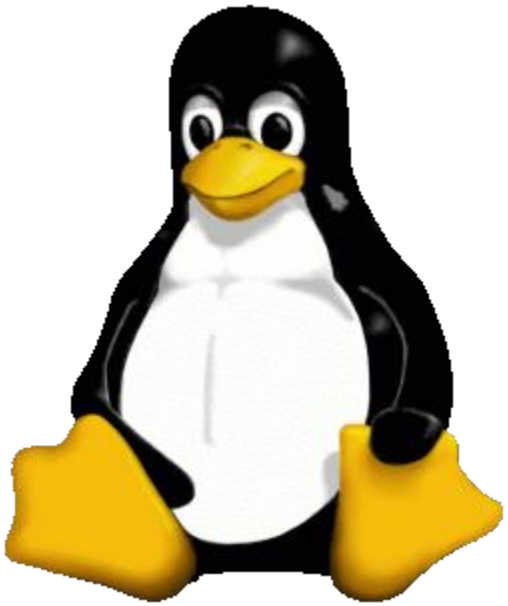
Lesson 5

By Dr. Amir

GNU/Linux

User and Groups



Users

A *user* is anyone who uses a computer.

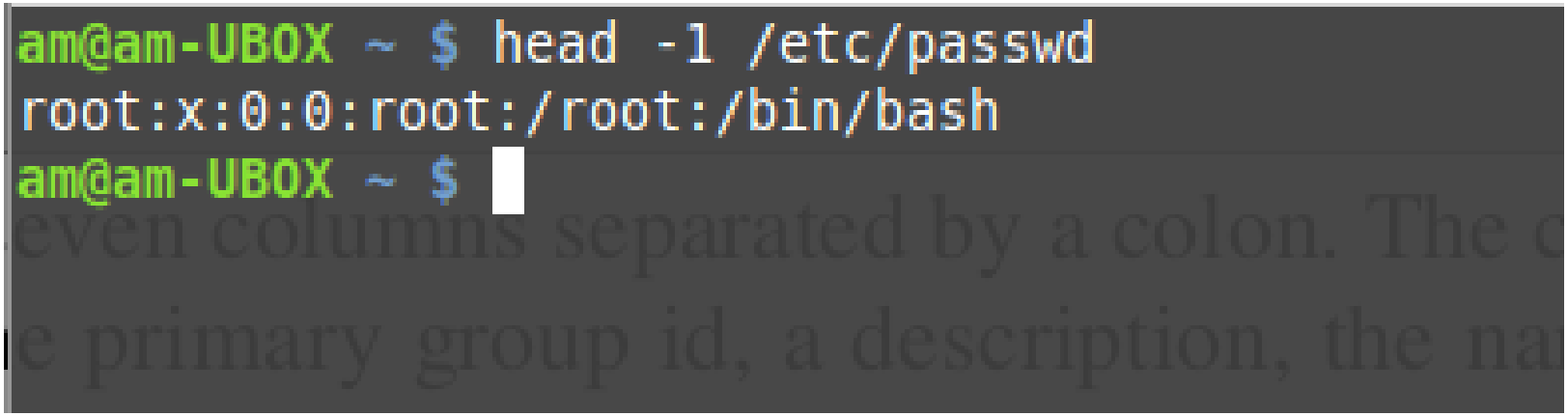
Managing users is done for the purpose of security by limiting access in certain specific ways

The superuser (root) has complete access to the operating system and its configuration; it is intended for administrative use only.

root

The root user also called the superuser is the most powerful account on your Linux system.

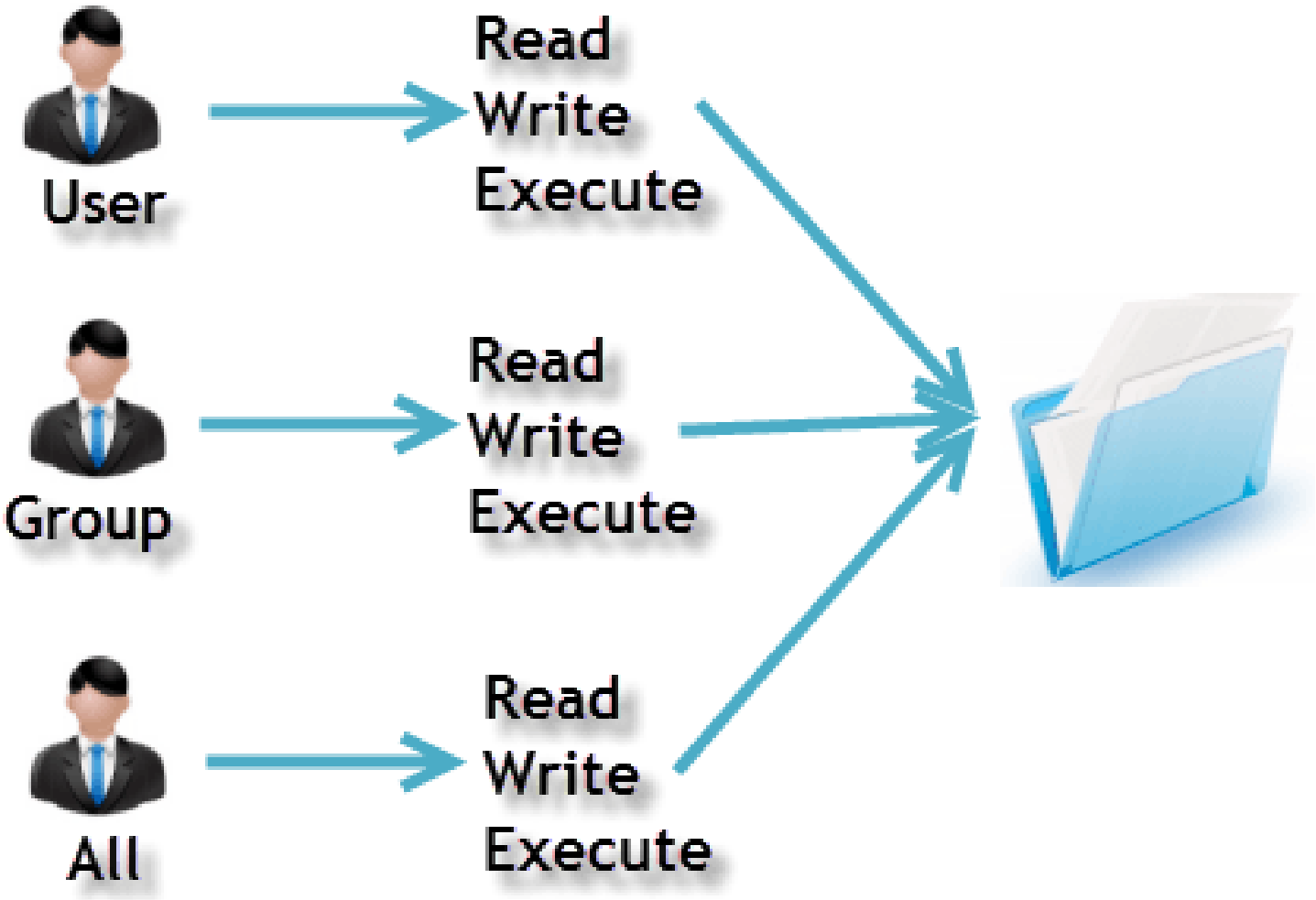
This user can do almost anything, including the creation of other users. The root user always has userid 0



In Linux everything is a file

Different users has different permissions and access to a wide range of input/output resources: documents, directories, hard-drives, CD-ROMs, modems, keyboards, printers, monitors, terminals and even some inter-process and network communications.

Owners assigned Permission On Every File and Directory



Command: useradd

To add a user:

m: creating home directory

d: setting the name of home directory

c: setting description

```
am@am-UBOX ~ $ useradd -m -d /home/john -c "John Nash" john
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
am@am-UBOX ~ $ sudo useradd -m -d /home/john -c "John Nash" john
[sudo] password for am:
am@am-UBOX ~ $ tail -1 /etc/passwd
john:x:1001:1001:John Nash:/home/john:
am@am-UBOX ~ $
```

The user named john received userid 1001 and primary group id 1001.

Command: userdel

To delete a user:

r: To delete home directory

```
am@am-UBOX ~ $ userdel -r john
userdel: Permission denied.
userdel: cannot lock /etc/passwd; try again later.
am@am-UBOX ~ $ sudo userdel -r john
userdel: john mail spool (/var/mail/john) not found
am@am-UBOX ~ $ tail -1 /etc/passwd
mysql:x:115:126:MySQL Server,,,:/nonexistent:/bin/false
am@am-UBOX ~ $
```

Command: usermod

To change the properties of a user

```
am@am-UBOX ~ $ sudo useradd -m -d /home/harry -c "Harry Potter" harry
am@am-UBOX ~ $
am@am-UBOX ~ $ tail -1 /etc/passwd
harry:x:1001:1001:Harry Potter:/home/harry:
```

```
am@am-UBOX ~ $ sudo usermod -c 'wizard' harry
am@am-UBOX ~ $
am@am-UBOX ~ $ tail -1 /etc/passwd
harry:x:1001:1001:wizard:/home/harry:
```

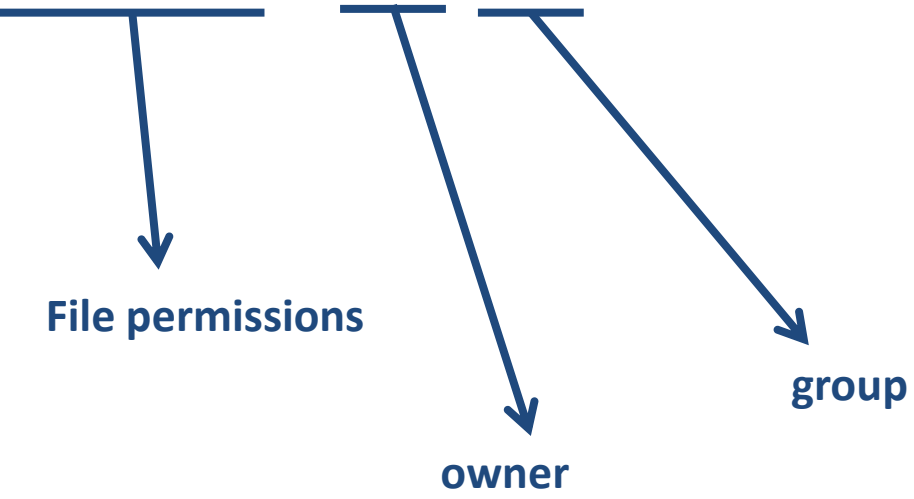
Access and permissions

Every file on a GNU/Linux system is owned by a user and a group. In addition, there are three types of access permissions: read, write, and execute. (rwx)

```
am@am-UBOX ~ $ ls -l /boot/
total 41286
-rw-r--r-- 1 root root  1207096 May  8  2015 abi-3.16.0-38-generic
-rw-r--r-- 1 root root   171817 May  8  2015 config-3.16.0-38-generic
drwxr-xr-x 3 root root     4096 Jan  1  1970 efi
drwxr-xr-x 5 root root     1024 Aug 21  2015 grub
-rw-r--r-- 1 root root 30306971 Mar 10 07:48 initrd.img-3.16.0-38-generic
drwx------ 2 root root    12288 Aug 21  2015 lost+found
-rw-r--r-- 1 root root   176500 Mar 12  2014 memtest86+.bin
-rw-r--r-- 1 root root   178176 Mar 12  2014 memtest86+.elf
-rw-r--r-- 1 root root   178680 Mar 12  2014 memtest86+_multiboot.bin
-rw------- 1 root root  3513313 May  8  2015 System.map-3.16.0-38-generic
-rw-r--r-- 1 root root  6351952 Jun 27  2015 vmlinuz-3.16.0-38-generic
am@am-UBOX ~ $
```

A file permission and ownership

```
am@am-UBOX ~ $ ls -l /boot/
total 41286
-rw-r--r-- 1 root root  1207096 May  8  2015 abi-3.16.0-38-generic
```



Command: stat

To find out the user ownership of a file

```
am@am-UBOX ~ $ stat -c %U /boot/abi-3.16.0-38-generic
root
am@am-UBOX ~ $
```

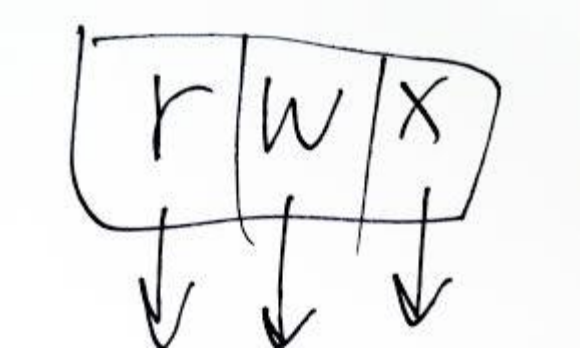
To find out the group ownership of a file

```
am@am-UBOX ~ $ stat -c %G /boot/abi-3.16.0-38-generic
root
am@am-UBOX ~ $
```

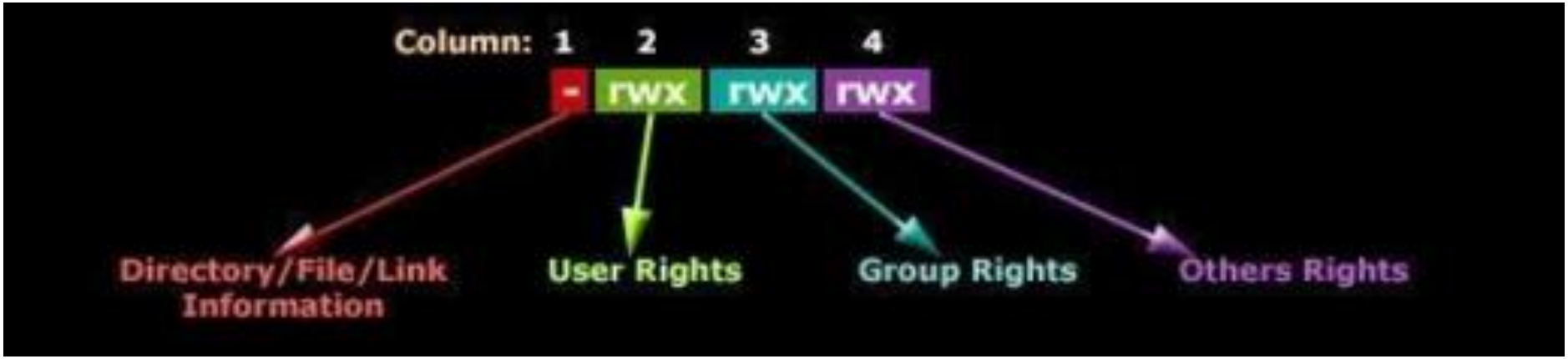
To find out the access permissions of a file

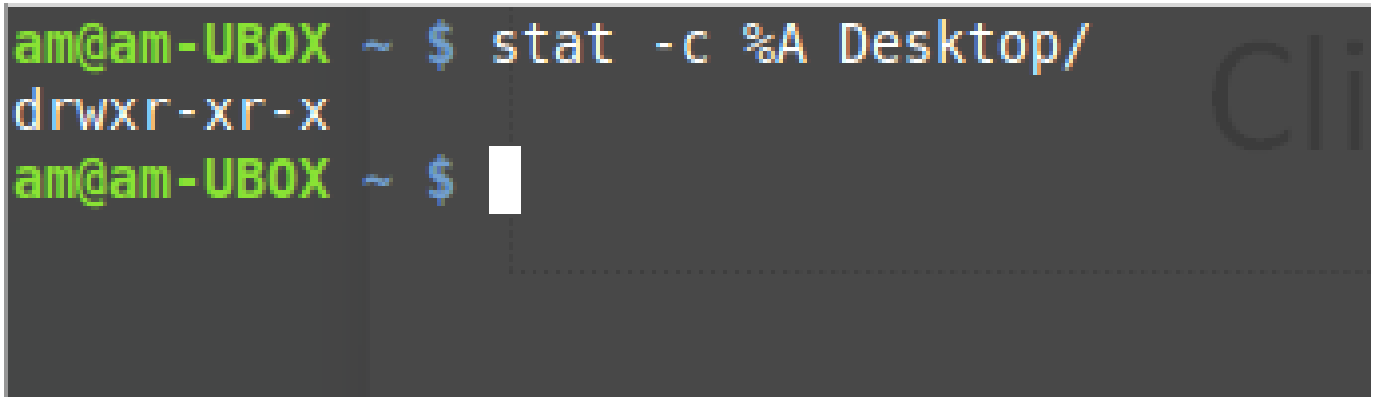
```
am@am-UBOX ~ $ stat -c %A /boot/abi-3.16.0-38-generic
-rw-r--r--
am@am-UBOX ~ $
```

Access Permissions



Access permissions are displayed in three groups of characters, representing the permissions of the owning user, owning group, and others, respectively.





Owner	Group	Other
r w x	r - x	r - x
4+2+1	4+0+1	4+0+1
7	5	5

Creating a user home directory

```
am@am-UBOX ~ $ sudo useradd john
```

```
am@am-UBOX ~ $ ls /home
am  harry
```

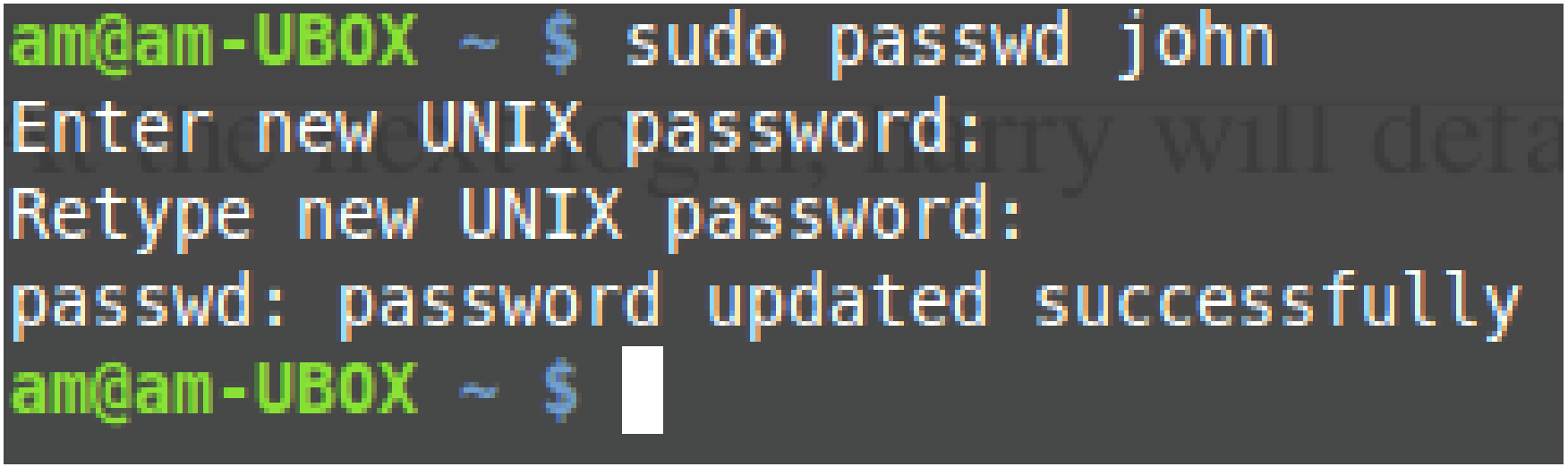
```
am@am-UBOX ~ $ sudo mkdir /home/john
am@am-UBOX ~ $ sudo chown john:john /home/john
```

```
am@am-UBOX ~ $ ls -lh /home
total 28K
drwx------ 71 am    am     20K Mar 27 15:54 am
drwxr-xr-x  4 harry harry 4.0K Mar 27 23:09 harry
drwxr-xr-x  2 john  john  4.0K Mar 27 23:16 john
```

```
am@am-UBOX ~ $ sudo chmod 700 /home/john
am@am-UBOX ~ $ ls -ld /home/john/
drwx------ 2 john john 4096 Mar 27 23:16 /home/john/
```

Command: passwd

To set a password for a user.



Command: find

To find files owning by a group or a user:

```
am@am-UBOX ~ $ find / -group root
```

```
am@am-UBOX ~ $ find / -group amir
```

Command: chown (change owner)

***chown* Change owner.**

Ex: chown <owner1> <filename> : Change ownership of a file to owner1.

***chgrp* Change group.**

Ex: chgrp <group1> <filename> : Change group of a file to group1.

Change owner of a file

```
am@am-UBOX ~ $ ls -lh b321
-rw-r--r-- 1 am am 0 Sep 21  2015 b321
am@am-UBOX ~ $ chown root b321
chown: changing ownership of ‘b321’: Operation not permitted
am@am-UBOX ~ $ sudo chown root b321
[sudo] password for am:
am@am-UBOX ~ $ ls -lh b321
-rw-r--r-- 1 root am 0 Sep 21  2015 b321
am@am-UBOX ~ $ sudo chown am b321
am@am-UBOX ~ $ ls -lh b321
-rw-r--r-- 1 am am 0 Sep 21  2015 b321
am@am-UBOX ~ $
```

Command: groups

To list groups membership for a user:

```
am@am-UBOX ~ $ sudo groups am
[sudo] password for am:
am : am adm dialout cdrom sudo dip plugdev lpadmin sambashare vboxusers
am@am-UBOX ~ $ sudo groups harry
harry : harry
am@am-UBOX ~ $ sudo groups john
john : john
am@am-UBOX ~ $
```

Command: id user

To list user id (UID) and group id (GID) for a user

```
am@am-UBOX ~ $ id am
uid=1000(am) gid=1000(am) groups=1000(am),4(adm),20(dialout),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),110(sambashare),124(vboxusers)
am@am-UBOX ~ $ id harry
uid=1001(harry) gid=1001(harry) groups=1001(harry)
am@am-UBOX ~ $ id john
uid=1002(john) gid=1002(john) groups=1002(john)
am@am-UBOX ~ $
```

To list all groups in the system:

```
am@am-UBOX ~ $ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,am
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:am
fax:x:21:
voice:x:22:
cdrom:x:24:am
floppy:x:25:
tape:x:26:
sudo:x:27:am
```

Command: groupadd

To create a new group:

```
am@am-UBOX ~ $ sudo groupadd hyit
am@am-UBOX ~ $
```

To add a user to a group:

```
am@am-UBOX ~ $ sudo gpasswd -a john hyit
Adding user john to group hyit
```

```
am@am-UBOX ~ $ groups john
john : john hyit
am@am-UBOX ~ $
```

Command: groupdel

To remove existing groups:

```
am@am-UBOX ~ $ sudo groupdel hyit
am@am-UBOX ~ $
am@am-UBOX ~ $ groups john
john : john
am@am-UBOX ~ $
```

Removing a user from a group

```
am@am-UBOX ~ $ sudo groupadd 1141
am@am-UBOX ~ $
am@am-UBOX ~ $ sudo gpasswd -a john 1141
Adding user john to group 1141
am@am-UBOX ~ $
am@am-UBOX ~ $ sudo groups john
john : john 1141
```

```
am@am-UBOX ~ $ sudo gpasswd -d john 1141
Removing user john from group 1141
am@am-UBOX ~ $ groups john
john : john
```

Next lesson:

Understanding ‘shell’

Control Operators

Exercises

Exercises will be uploaded to the course website.