04_Process&Network

Moatasem Elsayed

Linux commands can be categorized into various groups based on their functionality

1-File and Directory Management
-Help and Documentation

2-File Viewing and Editing

3-Controls: Redirection,Piping,Wildcard

4-File Permissions:

5-File Compression and Archiving:

6-Text Processing

7-System Information:

8-System Monitoring and Logging

9-search

10-Process Management

11-Networking:

12-User Management:

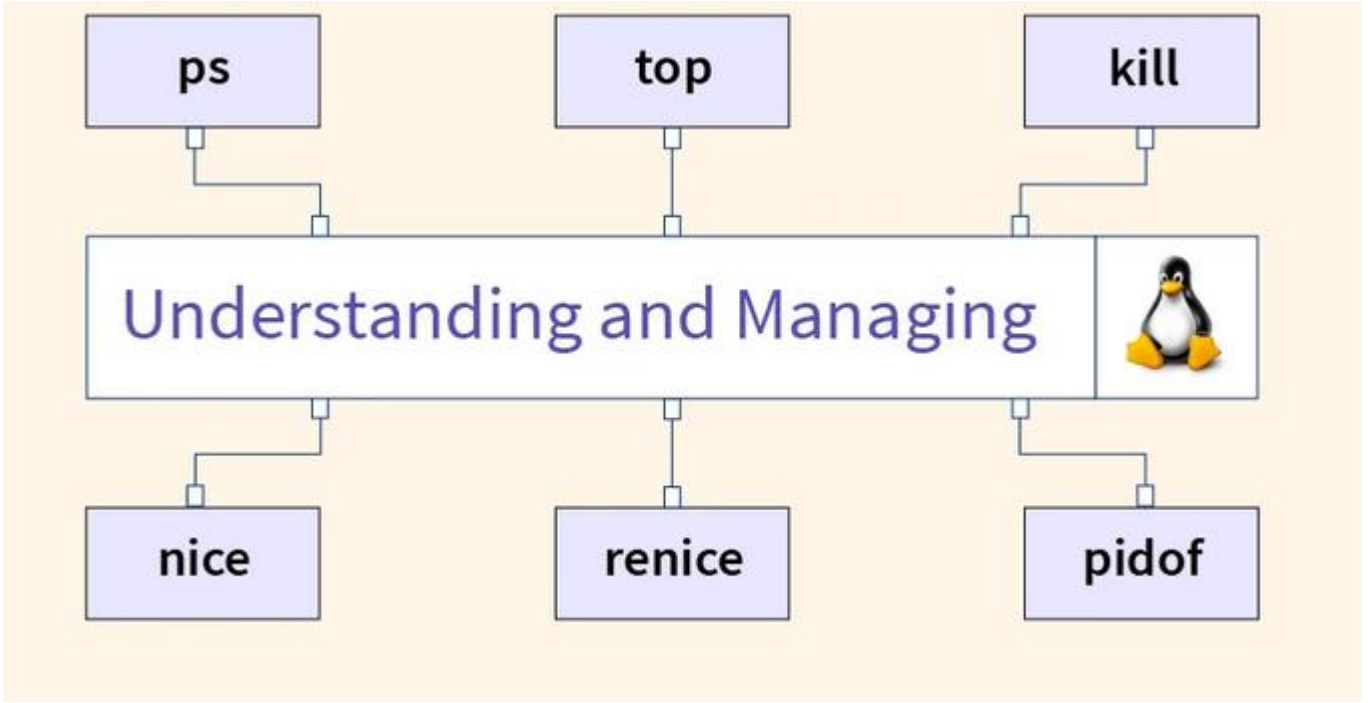
13-Package Management:

14-Tracking

15-Links

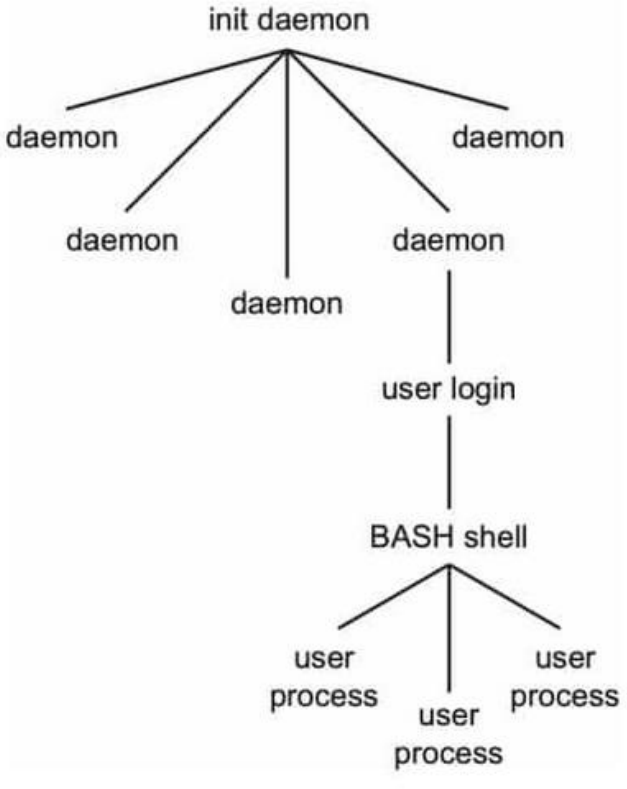
16- FS

10-Process Management



10-Process Management

1- Monitoring and info
2- Control back, fg,jobs
3- Signals
4- procfs
5- Change nice
6- Scheduler task



Ps –help all

- Ps aux
- Ps -elf
- ps

```
For more details see ps(1).
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps --help all

Usage:
 ps [options]

Basic options:
 -A, -e               all processes
 -a                   all with tty, except session leaders
  a                   all with tty, including other users
 -d                   all except session leaders
 -N, --deselect       negate selection
  r                   only running processes
  T                   all processes on this terminal
  x                   processes without controlling ttys

Selection by list:
 -C <command>         command name
 -G, --Group <GID>    real group id or name
 -g, --group <group>  session or effective group name
 -p, p, --pid <PID>   process id
        --ppid <PID>  parent process id
 -q, q, --quick-pid <PID>
                      process id (quick mode)
 -s, --sid <session>  session id
 -t, t, --tty <tty>   terminal
 -u, U, --user <UID>  effective user id or name
 -U, --User <UID>     real user id or name

  The selection options take as their argument either:
    a comma-separated list e.g. '-u root,nobody' or
    a blank-separated list e.g. '-p 123 4567'

Output formats:
 -F                   extra full
 -f                   full-format, including command lines
  f, --forest         ascii art process tree
 -H                   show process hierarchy
 -j                   jobs format
  j                   BSD job control format
 -l                   long format
```

PS

- **a**: option outputs all running processes of all users in the system.
- **u** option provides additional information like memory and CPU usage percentage, the process state code, and the owner of the processes.
- **x** option lists all processes not executed from the terminal. A perfect example of this are **daemons**, which are system-related processes that run in the background when the system is booted up.
- -f full-format, including command lines
- -A, -e all processes
- -l long format
- T terminal

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -lt /dev/pts/2
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000   29824   19603  0  80   0 -  5973 do_wai pts/2    00:00:00 bash
4 R  1000   30074   29824  0  80   0 -  5354 -      pts/2    00:00:00 ps
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$
```

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 167108 11708 ?        Ss   Sep13   0:02 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    Sep13   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   Sep13   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   Sep13   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   Sep13   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   Sep13   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   Sep13   0:00 [kworker/0:0H-events_highpri
root          10  0.0  0.0      0     0 ?        I<   Sep13   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        I    Sep13   0:00 [rcu_tasks_kthread]
root          12  0.0  0.0      0     0 ?        I    Sep13   0:00 [rcu_tasks_rude_kthread]
root          13  0.0  0.0      0     0 ?        I    Sep13   0:00 [rcu_tasks_trace_kthread]
root          14  0.0  0.0      0     0 ?        S    Sep13   0:00 [ksoftirqd/0]
root          15  0.1  0.0      0     0 ?        I    Sep13   0:28 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    Sep13   0:00 [migration/0]
root          17  0.0  0.0      0     0 ?        S    Sep13   0:00 [idle_inject/0]
root          19  0.0  0.0      0     0 ?        S    Sep13   0:00 [cpuhp/0]
```

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -elf
F S UID          PID    PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
4 S root           1       0  0  80   0 - 41777 -      Sep13 ?        00:00:02 /sbin/init splash
1 S root           2       0  0  80   0 -     0 -      Sep13 ?        00:00:00 [kthreadd]
1 I root           3       2  0  60 -20 -     0 -      Sep13 ?        00:00:00 [rcu_gp]
1 I root           4       2  0  60 -20 -     0 -      Sep13 ?        00:00:00 [rcu_par_gp]
1 I root           5       2  0  60 -20 -     0 -      Sep13 ?        00:00:00 [slub_flushwq]
1 I root           6       2  0  60 -20 -     0 -      Sep13 ?        00:00:00 [netns]
```

Examples

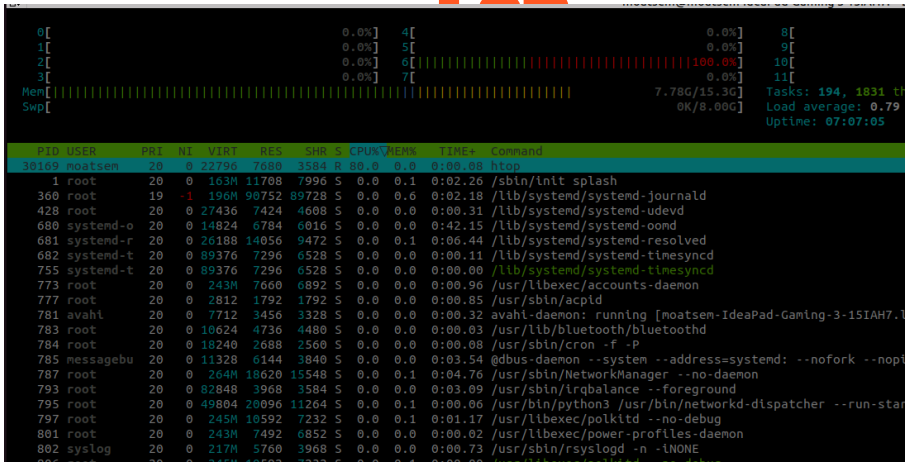
```
EXAMPLES
       To see every process on the system using standard syntax:
          ps -e
          ps -ef
          ps -eF
          ps -ely

       To see every process on the system using BSD syntax:
          ps ax
          ps axu

       To print a process tree:
          ps -ejH
          ps axjf

       To get info about threads:
          ps -eLf
          ps axms

       To get security info:
          ps -eo euser,ruser,suser,fuser,f,comm,label
          ps axZ
          ps -eM

       To see every process running as root (real & effective ID) in user format:
          ps -U root -u root u

       To see every process with a user-defined format:
          ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm
          ps axo stat,euid,ruid,tty,tpgid,sess,pgrp,ppid,pid,pcpu,comm
          ps -Ao pid,tt,user,fname,tmout,f,wchan

       Print only the process IDs of syslogd:
          ps -C syslogd -o pid=

       Print only the name of PID 42:
          ps -q 42 -o comm=
```

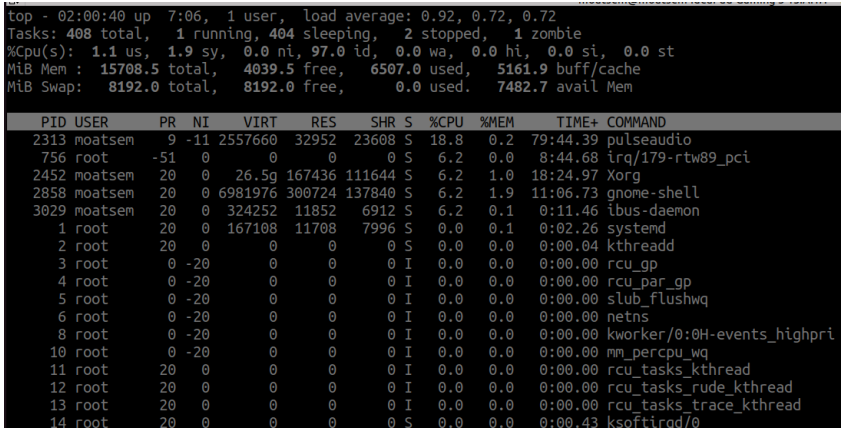
```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -e | head -n 3
    PID TTY          TIME CMD
      1 ?        00:00:02 systemd
      2 ?        00:00:00 kthreadd
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -ef | head -n 3
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 Sep13 ?        00:00:02 /sbin/init splash
root           2       0  0 Sep13 ?        00:00:00 [kthreadd]
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -eF | head -n 3
UID          PID    PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
root           1       0  0 41777 11708   8 Sep13 ?        00:00:02 /sbin/init splash
root           2       0  0     0     0   2 Sep13 ?        00:00:00 [kthreadd]
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -ely | head -n 3
S   UID     PID    PPID  C PRI  NI    RSS    SZ WCHAN  TTY          TIME CMD
S     0       1       0  0  80   0 11708 41777 -      ?        00:00:02 systemd
S     0       2       0  0  80   0     0     0 -      ?        00:00:00 kthreadd
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps aux  | head -n 3
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0 167108 11708 ?        Ss   Sep13   0:02 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    Sep13   0:00 [kthreadd]
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -ejH  | head -n 5
    PID    PGID     SID TTY          TIME CMD
      2       0       0 ?        00:00:00 kthreadd
      3       0       0 ?        00:00:00   rcu_gp
      4       0       0 ?        00:00:00   rcu_par_gp
      5       0       0 ?        00:00:00   slub_flushwq
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$
```

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -C firefox -o pid,cmd,ppid,tty,user
    PID CMD                         PPID TT       USER
   3555 /snap/firefox/3068/usr/lib/    2858 ?        moatsem
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$
```

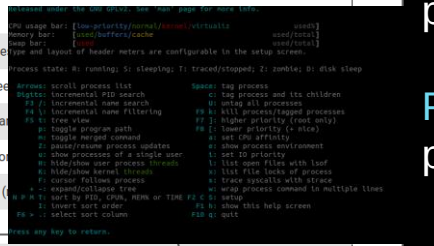
Htop

Top



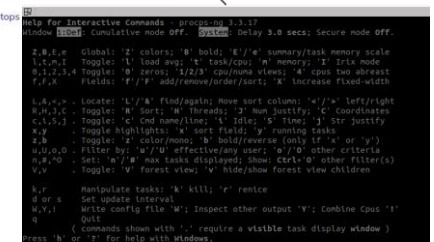


Function / Shortcut	Description
u	Sort processes by username
p	Toggle with the program path
F2 or S	Enter Setup
F3 or /	Search proce
F5 or t	Sorted or tre
F6 +/-	Select the pa
F7 or [	Increase prio
F8 or]	Low priority (
F9 or k	Kill process
H	Toggles with user process threads
K	Toggles with kernel process threads



pidof prog –name: find the process ID of a running program

Pstree :Display a tree of processes > pstree -p



Common needs

How to get process id ?

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ ps -aux | grep bash
moatsem    390431  0.0  0.0  15236  9076 pts/0    Ss    23:08   0:00 /bin/bash
moatsem    390899  0.0  0.0   9168  2432 pts/0    S+    23:09   0:00 grep --color=auto
moatsem   1244442  0.0  0.0  17084  2808 pts/5    Ss+   Oct18   0:00 /usr/bin/bash --i
```

Top => L + search

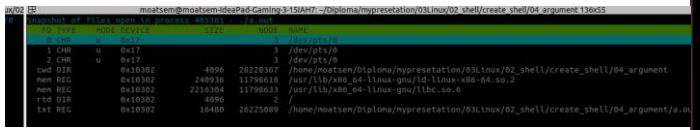
```
top - 23:13:40 up 9 days, 10:37,  1 user,  load average: 1.53, 1.48, 1.59
Tasks: 410 total,   1 running, 408 sleeping,   0 stopped,   1 zombie
%Cpu(s):  1.8 us,  1.6 sy,  0.0 ni, 96.4 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15708.5 total,    840.4 free,   6383.1 used,   8485.0 buff/cache
MiB Swap:   8192.0 total,   6099.2 free,   2092.8 used.   7328.2 avail Mem
Locate string bash
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 387000 root      20   0       0      0      0 I   0.0   0.0   0:00.17 kworker/8:2-
 387001 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/2:1-
 387725 root      20   0       0      0      0 I   0.0   0.0   0:00.10 kworker/6:2-
 388408 root      20   0       0      0      0 I   0.0   0.0   0:00.18 kworker/4:1-
 388409 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/13:2
 390431 moatsem   20   0   15236   9204   3840 S   0.0   0.1   0:00.03 bash
```

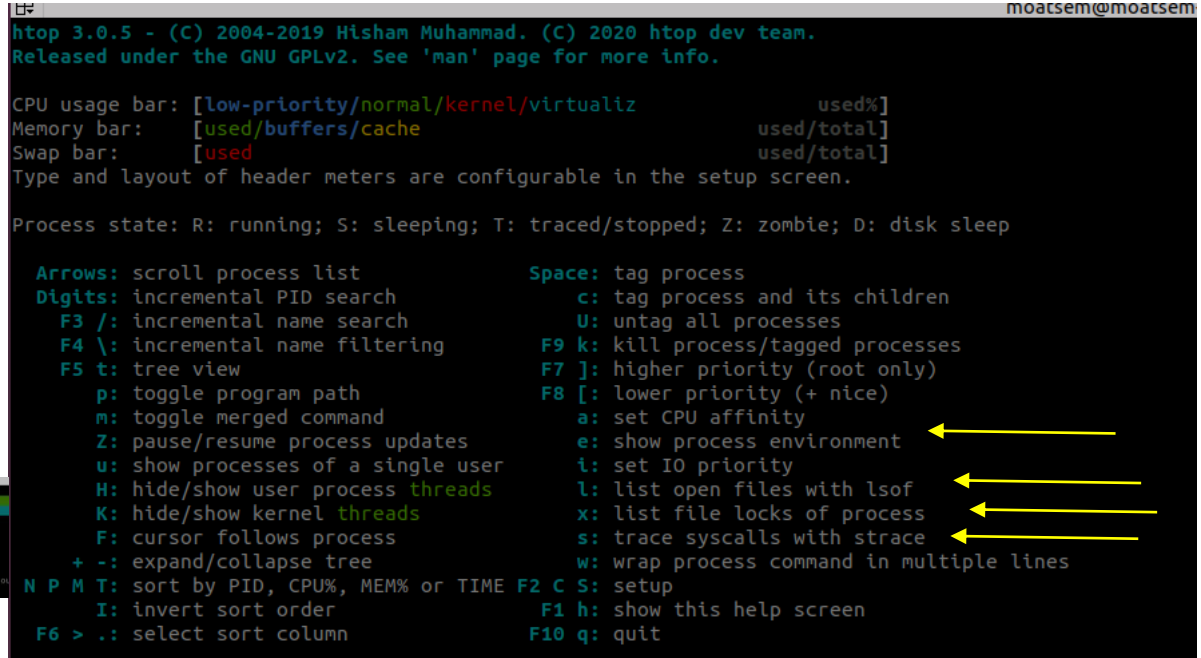
HTop => \ + search

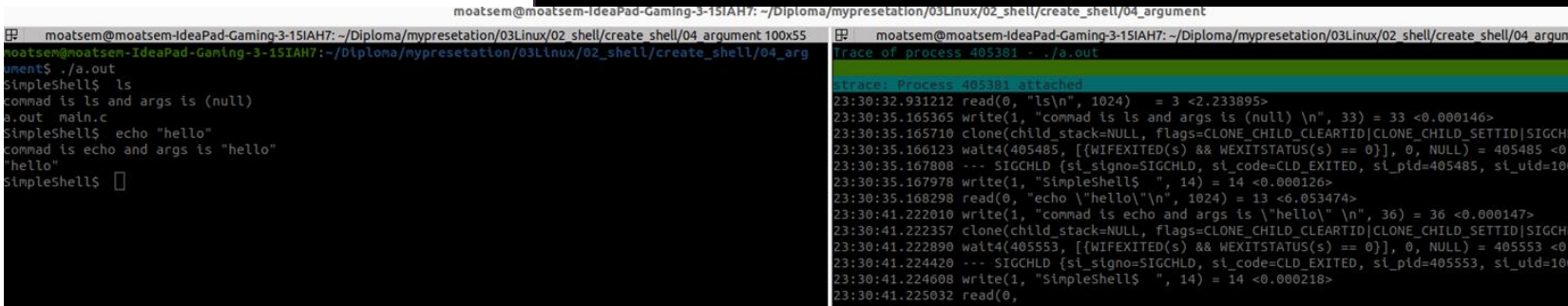
/ + search

```
 44417 moatsem    20   0  699M  9344  5800 S  0.0  0.1  0:00.31 /usr/libexec/gsd-power
   708 root       20   0  535M  9312  6752 S  0.0  0.1  1:52.39 /usr/sbin/NetworkManager --no-daemon
   837 root       20   0  535M  9312  6752 S  0.0  0.1  0:12.69 /usr/sbin/NetworkManager --no-daemon
   842 root       20   0  535M  9312  6752 S  0.0  0.1  0:20.40 /usr/sbin/NetworkManager --no-daemon
390431 moatsem    20   0 15236  9204  3840 S  0.0  0.1  0:00.03 /bin/bash
F3 Next  S-F3 Prev   Esc Cancel   Search: bash
```

HTOP


moatsem@moatsem-IdeaPad-Gaming-3-15IAH7: ~/Diploma/mypresetation/03Linux/02_shell/create_shell/04_argument 136x55


moatsem@moatsem
htop 3.0.5 - (C) 2004-2019 Hisham Muhammad. (C) 2020 htop dev team.
Released under the GNU GPLv2. See 'man' page for more info.
CPU usage bar: [low-priority/normal/kernel/virtualiz used%]
Memory bar: [used/buffers/cache used/total]
Swap bar: [used used/total]
Type and layout of header meters are configurable in the setup screen.
Process state: R: running; S: sleeping; T: traced/stopped; Z: zombie; D: disk sleep
Arrows: scroll process list
Digits: incremental PID search
F3 /: incremental name search
F4 \: incremental name filtering
F5 t: tree view
p: toggle program path
m: toggle merged command
Z: pause/resume process updates
u: show processes of a single user
H: hide/show user process threads
K: hide/show kernel threads
F: cursor follows process
+ -: expand/collapse tree
N P M T: sort by PID, CPU%, MEM% or TIME
I: invert sort order
F6 > .: select sort column
Space: tag process
c: tag process and its children
U: untag all processes
F9 k: kill process/tagged processes
F7]: higher priority (root only)
F8 [: lower priority (+ nice)
a: set CPU affinity
e: show process environment
i: set IO priority
l: list open files with lsof
x: list file locks of process
s: trace syscalls with strace
w: wrap process command in multiple lines
F2 C S: setup
F1 h: show this help screen
F10 q: quit


moatsem@moatsem-IdeaPad-Gaming-3-15IAH7: ~/Diploma/mypresetation/03Linux/02_shell/create_shell/04_argument
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7: ~/Diploma/mypresetation/03Linux/02_shell/create_shell/04_argument 100x55
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/Diploma/mypresetation/03Linux/02_shell/create_shell/04_arg
ument$ ./a.out
SimpleShell$ ls
commad is ls and args is (null)
a.out main.c
SimpleShell$ echo "hello"
commad is echo and args is "hello"
"hello"
SimpleShell$
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7: ~/Diploma/mypresetation/03Linux/02_shell/create_shell/04_argument 101x55
Trace of process 405381 - ./a.out
strace: Process 405381 attached
23:30:32.931212 read(0, "ls\n", 1024) = 3 <2.233895>
23:30:35.165365 write(1, "commad is ls and args is (null) \n", 33) = 33 <0.000146>
23:30:35.165710 clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_
23:30:35.166123 wait4(405485, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 405485 <0.001599>
23:30:35.167808 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=405485, si_uid=1000, si_sta
23:30:35.167978 write(1, "SimpleShell$ ", 14) = 14 <0.000126>
23:30:35.168298 read(0, "echo \"hello\"\n", 1024) = 13 <6.053474>
23:30:41.222010 write(1, "commad is echo and args is \"hello\" \n", 36) = 36 <0.000147>
23:30:41.222357 clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_
23:30:41.222890 wait4(405553, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 405553 <0.001442>
23:30:41.224420 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=405553, si_uid=1000, si_sta
23:30:41.224608 write(1, "SimpleShell$ ", 14) = 14 <0.000218>
23:30:41.225032 read(0,

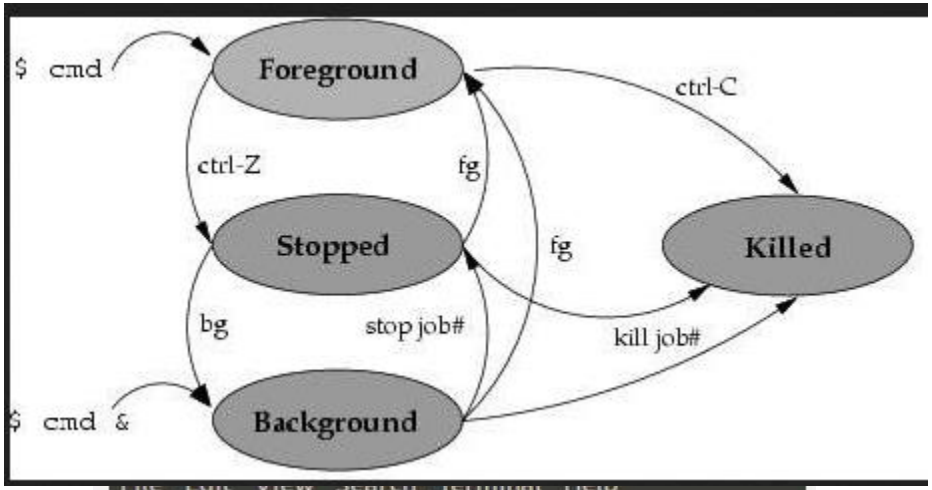
control back, fg,jobs

fg %1

Bg %1

jobs

Ctrl+z or &




```
naman@root:~$ jobs
naman@root:~$ sleep 500
^Z
[1]+  Stopped                 sleep 500
naman@root:~$ jobs
[1]+  Stopped                 sleep 500
naman@root:~$ bg %1
[1]+ sleep 500 &
naman@root:~$ jobs
[1]+  Running                 sleep 500 &
naman@root:~$
```

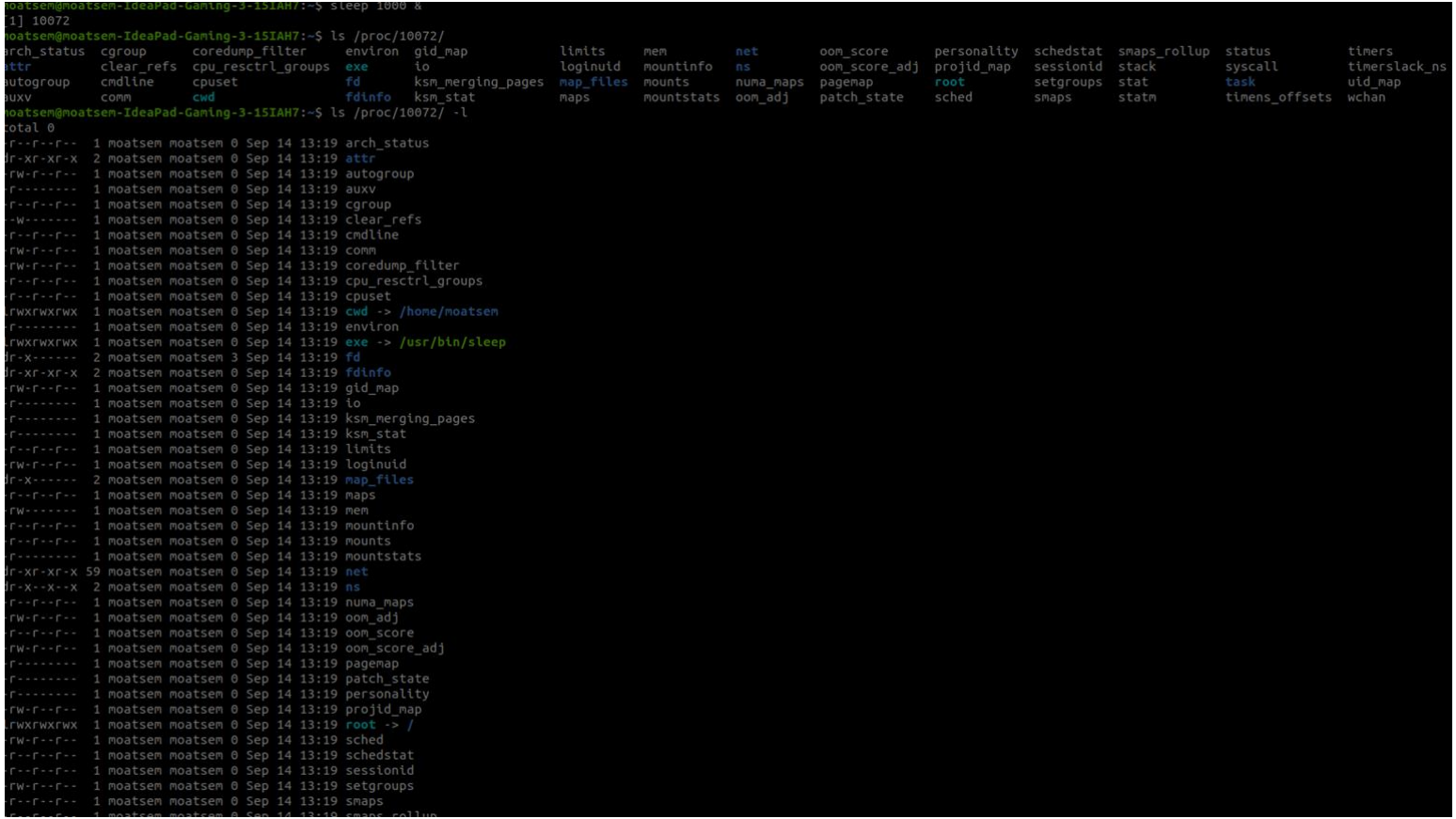
signals

Kill
Killall
pkill

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$
```

Procfs

1-fd
2-exe
3-cmd
4-environ



renice

`renice` is a Linux command that allows you to adjust the priority (niceness) of running processes. It is used to change the scheduling priority of a process, making it run with higher or lower priority in the CPU scheduler. Lower niceness values mean higher priority, while higher values mean lower priority. Here are some examples of how to use `renice`

```
# Increase the priority of a process with PID 123 to a higher value (e.g., -10)

sudo renice -n -10 -p 123

# Decrease the priority of a process with PID 456 to a higher value (e.g., 10)

sudo renice -n 10 -p 456

# Display the current niceness value of a process with PID 123

renice -p 123 -n
```

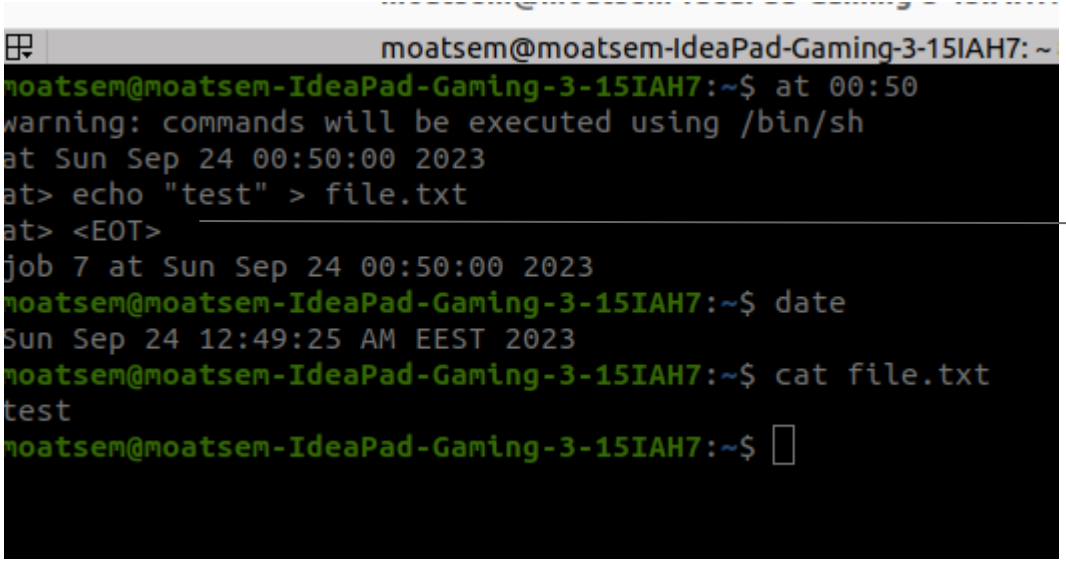
Scheduler Task (at,crontab)

```
- Edit the crontab file for the current user:
  crontab -e

- Edit the crontab file for a specific user:
  sudo crontab -e -u user

- Replace the current crontab with the contents of the given file:
  crontab path/to/file

- View a list of existing cron jobs for current user:
  crontab -l

- Remove all cron jobs for the current user:
  crontab -r

- Sample job which runs at 10:00 every day (* means any value):
  0 10 * * * command_to_execute

- Sample crontab entry, which runs a command every 10 minutes:
  */10 * * * * command_to_execute

- Sample crontab entry, which runs a certain script at 02:30 ev
  30 2 * * Fri /absolute/path/to/script.sh
```

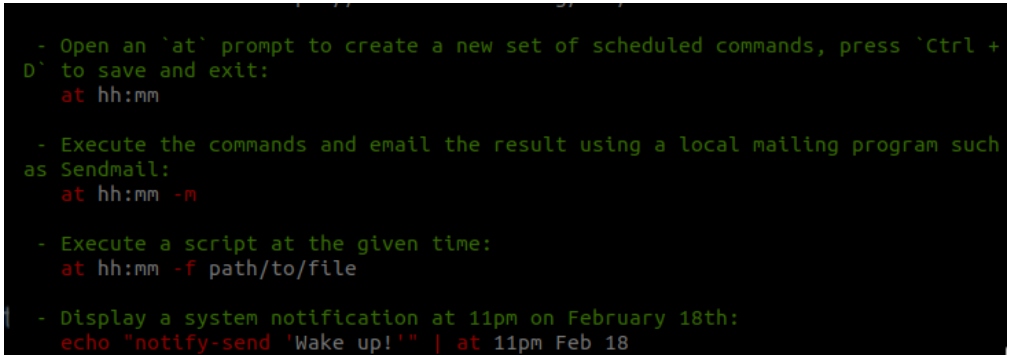
```
FILES
       /etc/cron.allow
       /etc/cron.deny
       /var/spool/cron/crontabs

       The files /etc/cron.allow and /etc/cron.deny if, they exist, must be either world-readabl
       the permissions are fixed.

       There  is  one  file for each user's crontab under the /var/spool/cron/crontabs directory
       lowed by the system to run periodic tasks can add them, and only syntactically correct cr
       group and configuring crontab command with the setgid bid set for that specific group.

STANDARDS
```

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ sudo cat /var/spool/cron/crontabs/moatsem
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.3UUmhm/crontab installed on Wed Sep 20 16:14:20 2023)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

  #* *  *    *   *   /home/moatsem/scripts/test.sh
  #* *  *    *   *    date > ~/.file2.txt
  */30 * * * *        XDG_RUNTIME_DIR=/run/user/$(id -u) notify-send Hey "please Moatasem you should Leave office and do some
```

at


moatsem@moatsem-IdeaPad-Gaming-3-15IAH7: ~
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ at 00:50
warning: commands will be executed using /bin/sh
at Sun Sep 24 00:50:00 2023
at> echo "test" > file.txt
at> <EOT>
job 7 at Sun Sep 24 00:50:00 2023
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ date
Sun Sep 24 12:49:25 AM EEST 2023
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$ cat file.txt
test
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~$

Ctrl +D


- Open an `at` prompt to create a new set of scheduled commands, press `Ctrl + D` to save and exit:
at hh:mm
- Execute the commands and email the result using a local mailing program such as Sendmail:
at hh:mm -m
- Execute a script at the given time:
at hh:mm -f path/to/file
- Display a system notification at 11pm on February 18th:
echo "notify-send 'Wake up!'" | at 11pm Feb 18

Task

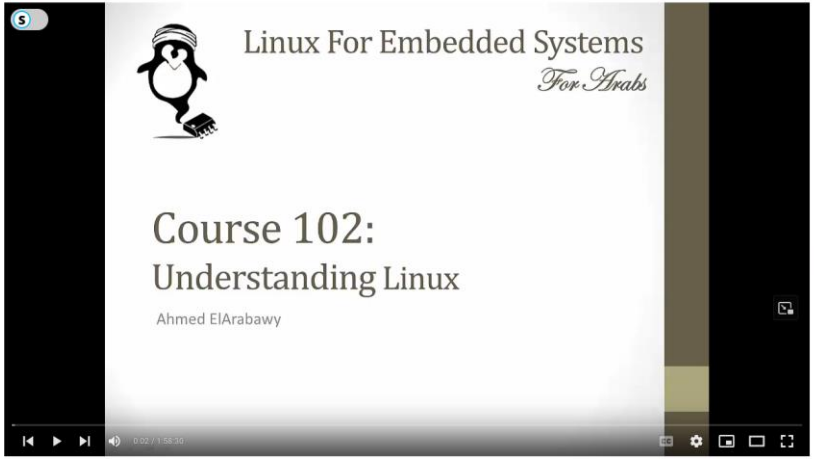
1- Create task to log the uptime every monday on 1:00 pm
2- Create task to run python script which sending an email to friend with report of uptime every sunday at 9:00 am

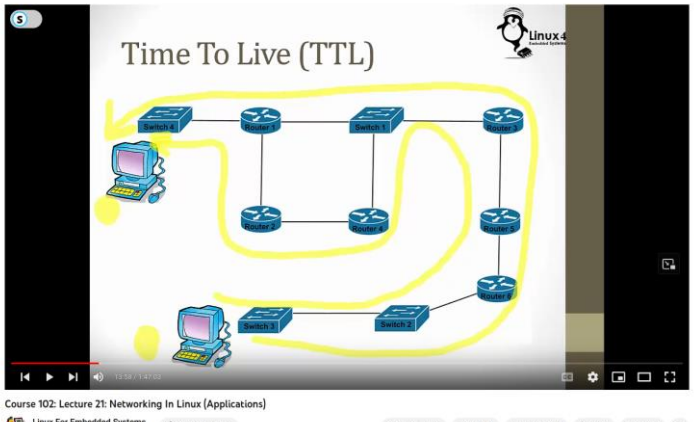
3- create task run every half hour to check battery value and report if it full or not

4- create task to run on 1/1 every year at 12:00 sending message to whatsapp with happy anniversary

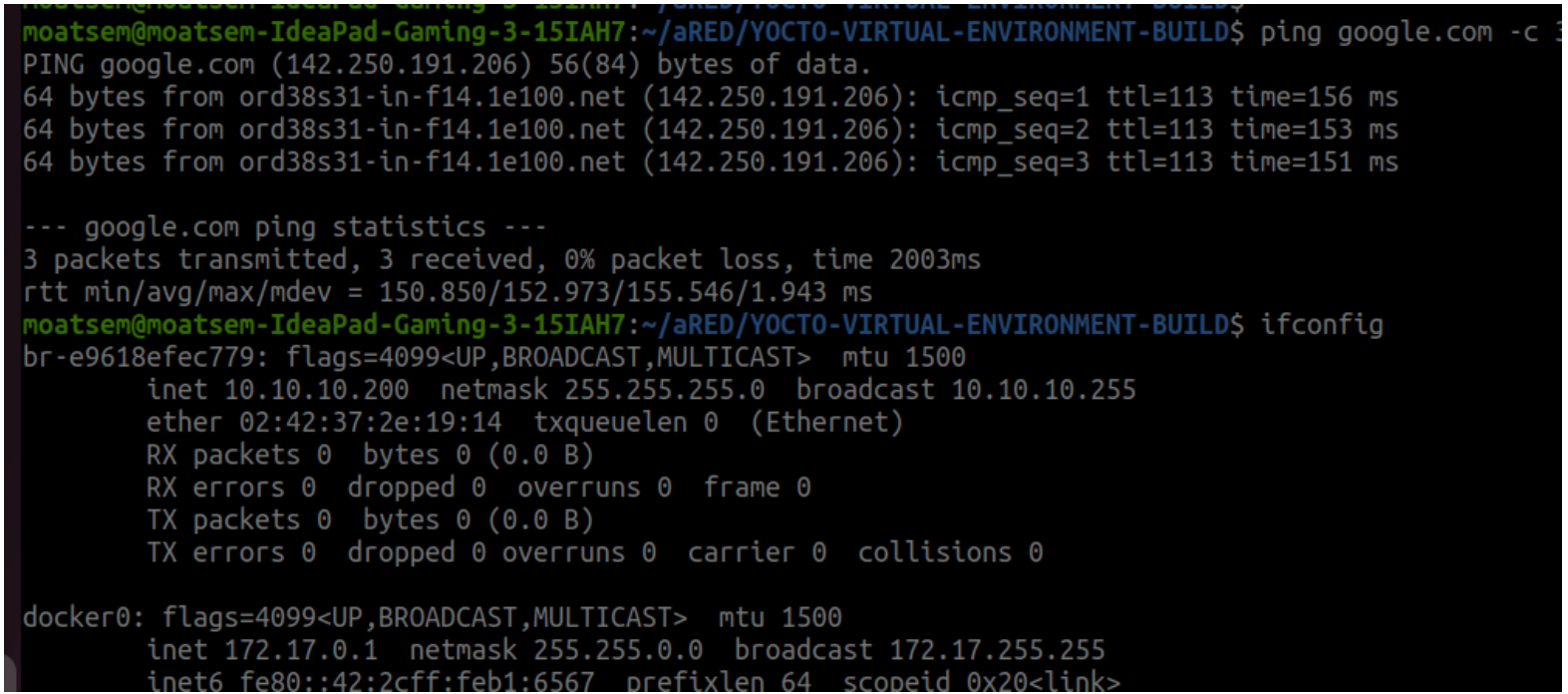
11-Networking

1- ping, ifconfig
2- netstat
3- ssh
4- scp
5-route , arp,tracepath





ifconfig,ping



ssh,scp,ssh-copy-id,netstat

```
alias connectOldLab='ssh moatasem@192.168.100.2'
alias connectOldLab_network2='ssh moatasem@192.168.1.12'
alias connectrpi='ssh moatasem@raspberrypi.local'
alias connecttablet='ssh moatasem@192.168.100.221'
alias connecttablet_network2='ssh moatasem@192.168.1.13'
```

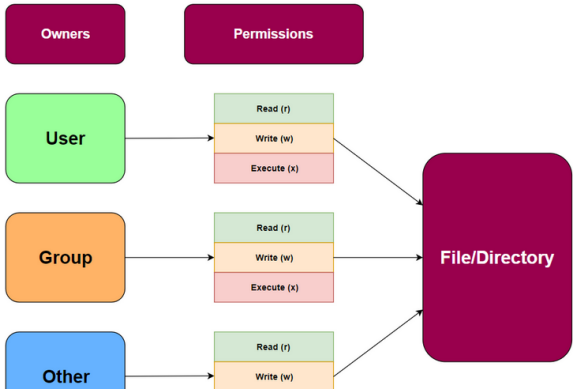
- Copy a local file to a remote host:
  scp path/to/local_file remote_host:path/to/remote_file

- Recursively copy the contents of a directory from a remote host to a local directory:
  scp -r remote_host:path/to/remote_directory path/to/local_directory

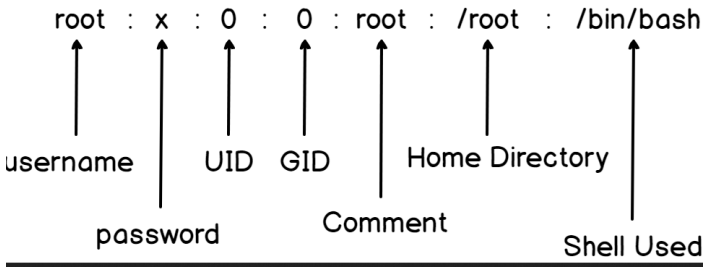
```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/aRED/YOCTO-VIRTUAL-ENVIRONMENT-BUILD$ netstat -tlup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:8888            0.0.0.0:*               LISTEN      880/hello_world
tcp        0      0 127.0.0.53:domain       0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:netbios-ssn     0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:microsoft-ds    0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:29754         0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN      -
```

12-User Management:

Useradd
Userdel
Usermod
Su
Passwd
Groupmod
Groupdel
groupadd

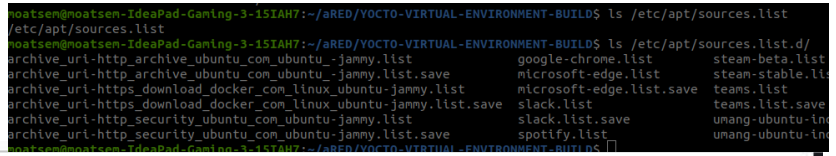


/etc/passwd columns




```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/aRED/YOCTO-VIRTUAL-ENVIRONMENT-BUILD$ tldr useradd

  useradd

  Create a new user.
  See also: `users`, `userdel`, `usermod`.
  More information: https://manned.org/useradd.

  - Create a new user:
    sudo useradd username

  - Create a new user with the specified user id:
    sudo useradd --uid id username

  - Create a new user with the specified shell:
    sudo useradd --shell path/to/shell username

  - Create a new user belonging to additional groups (mind the lack of whitespace):
    sudo useradd --groups group1,group2,... username

  - Create a new user with the default home directory:
    sudo useradd --create-home username

  - Create a new user with the home directory filled by template directory files:
    sudo useradd --skel path/to/template_directory --create-home username

  - Create a new system user without the home directory:
    sudo useradd --system username

moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/aRED/YOCTO-VIRTUAL-ENVIRONMENT-BUILD$ tldr usermod

  usermod

  Modifies a user account.
  See also: `users`, `useradd`, `userdel`.
  More information: https://manned.org/usermod.

  - Change a username:
    sudo usermod --login new_username username

  - Change a user id:
    sudo usermod --uid id username

  - Change a user shell:
    sudo usermod --shell path/to/shell username

  - Add a user to supplementary groups (mind the lack of whitespace):
    sudo usermod --append --groups group1,group2,... username

  - Change a user home directory:
    sudo usermod --move-home --home path/to/new_home username
```

13-Package Management:



-source code **(Youtube Channel)**

-dpkg

-apt-get

```
- Install a package:
  dpkg -i path/to/file.deb

- Remove a package:
  dpkg -r package

- List installed packages:
  dpkg -l pattern

- List a package's contents:
  dpkg -L package

- List contents of a local package file:
  dpkg -c path/to/file.deb

- Find out which package owns a file:
  dpkg -S path/to/file
```

apt-get	function of the command
apt-get install	Installs a package
apt-get remove	Removes a package
apt-get update	Refreshes repository index
apt-get upgrade	Upgrades all upgradable packages
apt-get dist-upgrade	Upgrades packages with auto-handling of dependencies
apt-cache search	Searches for the program
apt-cache show	Shows package details

14-Tracking

-strace

-readelf

-ldd

-gdb

-objdump

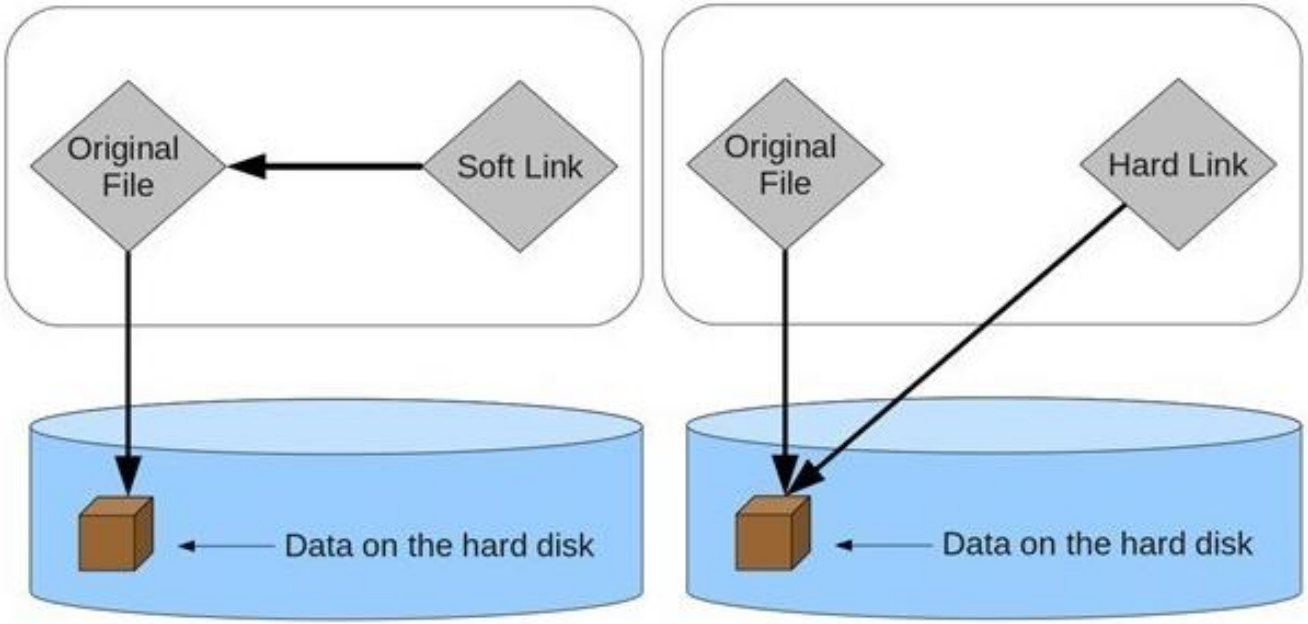
-

```
moatsem@moatsem-IdeaPad-Gaming-3-15IAH7:~/aRED/YOCTO-VIRTUAL-ENVIRONMENT-BUILD$ strace ls
execve("/usr/bin/ls", ["ls"], 0x7ffe307cbd90 /* 68 vars */) = 0
brk(NULL)                               = 0x5639fb146000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffe94f5ef40) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f6d24efa000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=130795, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 130795, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f6d24eda000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=166280, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 177672, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6d24eae000
mprotect(0x7f6d24eb4000, 139264, PROT_NONE) = 0
mmap(0x7f6d24eb4000, 106496, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7f6d24eb400
mmap(0x7f6d24ece000, 28672, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x20000) = 0x7f6d24ece000
mmap(0x7f6d24ed6000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x27000) = 0x7f6d24ed600
mmap(0x7f6d24ed8000, 5640, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6d24ed8000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340."..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2216304, ...}, AT_EMPTY_PATH) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6d24c00000
mmap(0x7f6d24c28000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f6d24c28
mmap(0x7f6d24dbd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f6d24dbd000
mmap(0x7f6d24e15000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f6d24e15
mmap(0x7f6d24e1b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f6d24e1b000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcre2-8.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=613064, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 615184, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f6d24b69000
mmap(0x7f6d24b6b000, 438272, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f6d24b6b00
mmap(0x7f6d24bd6000, 163840, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6d000) = 0x7f6d24bd6000
mmap(0x7f6d24bfe000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x94000) = 0x7f6d24bfe00
```

15-Links

-Hard Link

-Symbolic link



A link in UNIX is a pointer to a file. Like pointers in any programming languages, links in UNIX are pointers pointing to a file or a directory. Creating links is a kind of shortcuts to access a file. Links allow more than one file name to refer to the same file, elsewhere.

There are two types of links :

1-Soft Link or Symbolic links

2-Hard Links

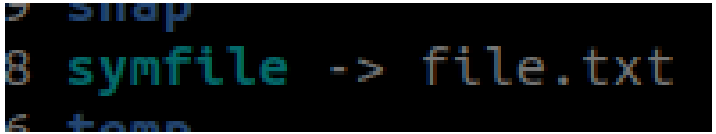
links

```
file.txt': File exists
H7:~$ ln -s file.txt symfile
```

```
ln -s /data/directory1 ~/directory1
```

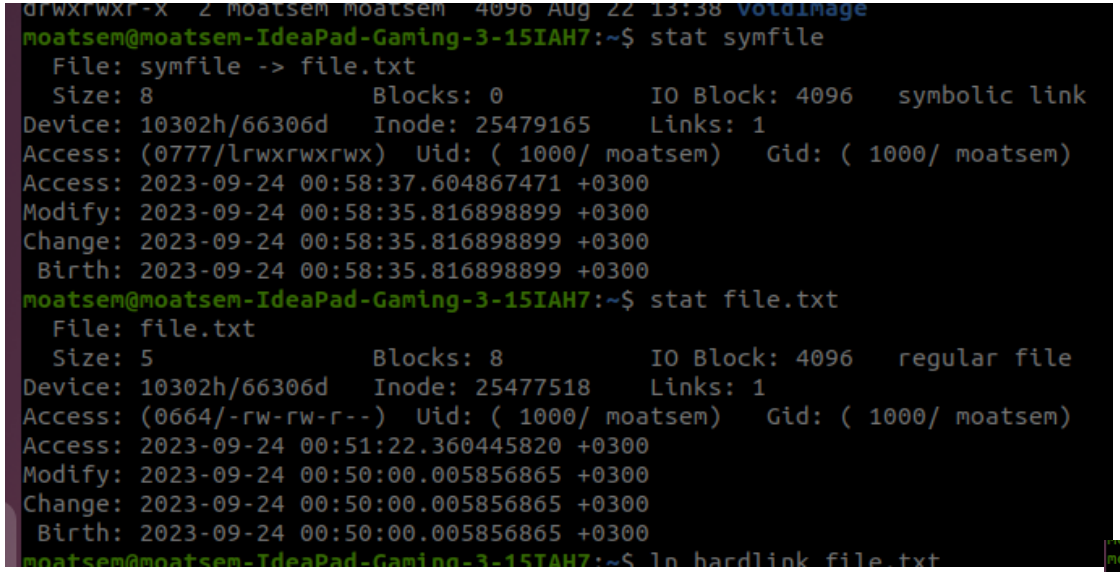
```
ln -s /data/file2 ~/file2
```

```
ln  file.txt hardlink
```

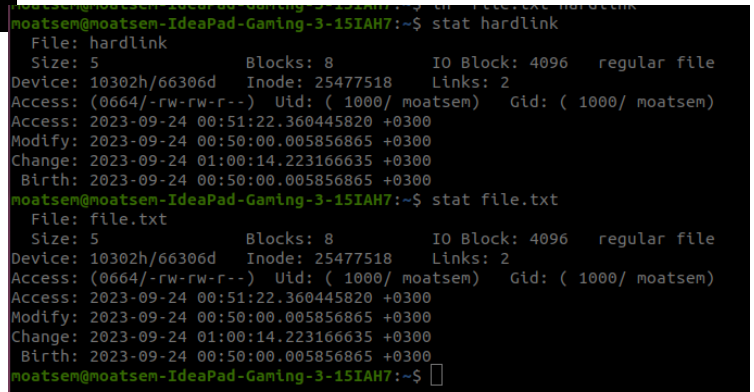


stat

Soft Link



Hard Link



16-FS

-mount

-dd

mount

- Displays information about file systems mounted:

```
vivek@vivek-X556UQK:~$ sudo mount -l -t ext4
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
/dev/sda4 on /media/vivek type ext4 (rw,relatime,data=ordered)
/dev/sda5 on /media/vivek type ext4 (rw,relatime,data=ordered)
vivek@vivek-X556UQK:~$ sudo mount -l -t fuseblk
/dev/sda6 on /media/vivek type fuseblk (rw,relatime,user_id=0,group_id=0,allow_o
ther,blksize=4096)
vivek@vivek-X556UQK:~$
```

- Mounts file systems:

```
vivek@vivek-X556UQK:~$ sudo mount /dev/sda4 /media/vivek
vivek@vivek-X556UQK:~$ sudo mount /dev/sda5 /media/vivek
vivek@vivek-X556UQK:~$ sudo mount /dev/sda6 /media/vivek
vivek@vivek-X556UQK:~$ sudo mount -l -t ext4
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
/dev/sda4 on /media/vivek type ext4 (rw,relatime,data=ordered)
/dev/sda5 on /media/vivek type ext4 (rw,relatime,data=ordered)
vivek@vivek-X556UQK:~$ sudo mount -l -t fuseblk
/dev/sda6 on /media/vivek type fuseblk (rw,relatime,user_id=0,group_id=0,allow_o
ther,blksize=4096)
vivek@vivek-X556UQK:~$
```

- Displays version information:

```
vivek@vivek-X556UQK:~$ sudo mount -V
mount from util-linux 2.27.1 (libmount 2.27.0: selinux, assert, debug)
vivek@vivek-X556UQK:~$
```

- Unmounts file systems:

```
vivek@vivek-X556UQK:~$ sudo umount /dev/sda6
vivek@vivek-X556UQK:~$ sudo umount /dev/sda5
vivek@vivek-X556UQK:~$ sudo umount /dev/sda4
vivek@vivek-X556UQK:~$ sudo mount -l -t fuseblk
vivek@vivek-X556UQK:~$ sudo mount -l -t ext4
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
vivek@vivek-X556UQK:~$
```

dd

Clear the flash card with zeros:

To clear the flash card with zeros, you can use the following `dd` command:

bash ✓ Copied!

```
sudo dd if=/dev/zero of=/dev/sdX bs=1M status=progress
```

```
- Make a bootable USB drive from an isohybrid file (such like `archlinux-xxx.iso`) and show the progress:
  dd if=path/to/file.iso of=/dev/usb_drive status=progress
```

```
Generate a file of 100 random bytes by using kernel random driver:
dd if=/dev/urandom of=path/to/random_file bs=100 count=1
```

1. **To backup the entire harddisk :** To backup an entire copy of a hard disk to another hard disk connected to the same system, execute the dd command as shown. In this dd command example, the UNIX device name of the source hard disk is /dev/hda, and device name of the target hard disk is /dev/hdb.

   ```
   # dd if=/dev/sda of=/dev/sdb
   ```

   - *"if"* represents inputfile, and *"of"* represents output file. So the exact copy of */dev/sda* will be available in */dev/sdb*.
   - If there are any errors, the above command will fail. If you give the parameter *"conv=noerror"* then it will continue to copy if there are read errors.
   - Input file and output file should be mentioned very carefully. Just in case, you mention source device in the target and vice versa, you might loss all your data.
   - To copy, hard drive to hard drive using dd command given below, sync option allows you to copy everything using synchronized I/O.

     ```
     # dd if=/dev/sda of=/dev/sdb conv=noerror, sync
     ```

2. **To backup a Partition :** You can use the device name of a partition in the input file, and in the output either you can specify your target path or image file as shown in the dd command.

   ```
   # dd if=/dev/hda1 of=~/partition.img
   ```

Tasks

1- Write code to catch interrupt signal and print exit before termination
2- write code to whenever it runs it kills all firefox processes
3- using ssh , have access to your pc with ssh
4-on VM create users with their bash and home dir then add them to sudeors group then delete them
5-mount flash and unmount the flash
6- clear sdcard and write random values on the sdcard

Linux

Session1 (3hr)	Introduction To Linux World · Intro History Why Linux and Embedded Linux booting sequence system calls · Ubuntu Installation · Linux File System . Folder Structure
Session2 (3hr)	-shell -memory allocation -commands for navigation
Session3 (3hr)	-commands for creation - commands for editing - piping - logic -tracing
Session4 (3hr)	- commands for searching - commands for networking - hardware information - systemd vs systemv -kernel info

our journey is just beginning

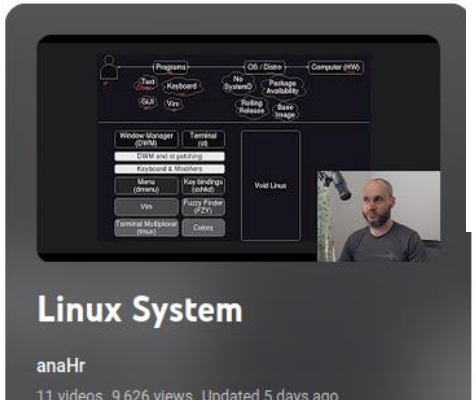
Linux Certified System Administrator 8 (Arabic)

Shako Mako Tech شكو ماكو

Learn Linux for beginners (in Arabic) Episode 1- عربي (الحلقة ١) تعلم نظام لينكس للمبتدئين • 14:15

Learn Virtualization concept Arabic- عربي(الحلقة٢) تعلم مبادئ القيرچولايزيشن (الافتراضية) في ٧ دقائق • 7:24

VIEW FULL PLAYLIST



Linux System

anaHr

11 videos 9,626 views Updated 5 days ago

The Linux Cast

@TheLinuxCast 35.6K subscribers 884 videos

Daily FOSS related content, weekly Linux podcasts, and n

patreon.com/thelinuxcast and 4 more links

The Linux Foundation

@LinuxfoundationOrg 170K subscribers 5.8K vide

The Linux Foundation is a nonprofit consortium de

The Linux Experiment

@TheLinuxEXP 295K subscribers 628 v

Making Linux accessible: no techno babb

patreon.com/thelinuxexperiment and 1 n

tutoriaLinux

@tutoriaLinux 202K subscribers 232 videos

Linux, DevOps, Cloud, and Programming tutorial vic

tutorialinux.com and 3 more links