## linux basic

home

## commands:

pwd

ls

-l

mkdir

rmdir

cd advC

clear

vi

mv

gcc

-c

-w

-o

who

whoami

ps

==lots of options

-ax

pstree

kill

pkill

ipcs

-q -m -s -l

ipcrm

-q -m -s

## vi

insert

command

:w

:wq

:w!

:q!

1010 0011 163 0 to 255

0101 1100

0101 1101 -93 -128 to 127

-93 0101 1101

1010 0010

1010 0011 A3

163 1010 0011 A3

OS scheduling:

Round Robin

Round Robin with interrupts

…

...

Function Queue Scheduling

…

…

## paths:

relative

absolute

.. parent

. current folder

~ home

(mount point)

- previous folder

### shell :

sh (bourne)

zsh

csh

ksh (korn)

tcsh

bash (bourne again shell)

born again

users & groups

files & folders

processes

inode

pid

ppid

init

systemd

file descriptor

chmod

chown

chgrp

stdin 0

stdout 1

stderr 2

quotes.txt 3

### Signals:

kill send a signal

SIGTERM 15

default

terminates

can be handled

SIGINT 2

interrupt from the keyboard

terminates

can be handled

can be sent from keyboard

SIGQUIT

terminates

can be handled

can be sent from keyboard

core dump

SIGKILL 9

terminates

can not be handled

ctrl+C SIGINT

ctrl+Z SIGSTOP

ctrl+\ SIGQUIT

can not be handled, blocked, ignored…

SIGKILL

SIGSTOP

### man:

1 commands

2 system calls

3 functions, library calls

4 special files

5 configuration files

6 games, entertainment etc

7 miscellaneous ,

protocols, signals, filesystem layout...

### states:

R run

S wait, sleep

T STOP

I kernel state

---------------------------

Z zombie state

D uninterruptible state

X dead state

subscripts:

+ foreground (needs to display)

l multi-thread

s session leader

ps

pid

ppid

stat

pri

ni

etime

stime (system when it was launched)

time (time spent in the privileged mode)

cmd (command used to launch)

lwp (threads)

nlwp (num of threads)

daemon

background of system

not in the terminal

no parent (it will have a parent only for the records)

no child

vfs

virtual file system

procfs

### fork:

i=fork()

parent:

i = child’s pid

child:

i =0

## POSIX:

1a OS

1b RTOS

1c threads (pthreads)

### system calls:

getpid()

getppid()

open()

close()

read()

write()

create()

lseek()

signal()

kill()

fork()

wait()

msgget()

msgsnd()

msgrcv()

msgctl()

### special functions:

sleep()

system()

ftok()

## IPC

### primitive :

pipes

fifos (named pipes)

### system V

msg queue

shared memory

semaphores

### POSIX:

msg queue

shared memory

semaphores

mutex

### pipes:

data once read is deleted

read has to wait for write process

unidirectional

related process

### named pipes

data once read is deleted

read has to wait for write process

unidirectional

unrelated process

both ends should be open for fifo to be created in RAM

file tokens

ftok()

### sys V message queue:

ftok()

msgget()

// semget, shmget

msgsnd()

msgrcv()

msgctl()

IPC\_NOWAIT

1. files

read-write

lseek

create

1. fork

wait

1. fork

read-write

1. fork

signal

1. fork

1 parent , 3 children

1. fork

1 parent, 3 children

wait

1. chatting
2. fifo
3. msg queues