

LEC 4

Created by : Israa Abdelghany

LinkedIn : www.linkedin.com/in/israa-abdelghany-4872b0222

GitHub : <https://github.com/IsraaAbdelghany9>

Revision

* PS1 → environment variable & PS2

* execute folder → enter the folder

- to print all shell environment variables
↳ env or printenv

echo \$HOME

pwd

echo \$PWD

echo \$SHELL

hostname or echo \$HOSTNAME

* if we want file to be visible for all users we can put the file or folder in the `/usr/bin` folder not even `sbin`

\$ → tells terminal that the next is variable using echo will print its value

* to add new path to be included each time we can use in the `.bashrc` ⇒ `PATH = $PATH:/home/file`

Lecture :-

* processes run in back ground is daemons

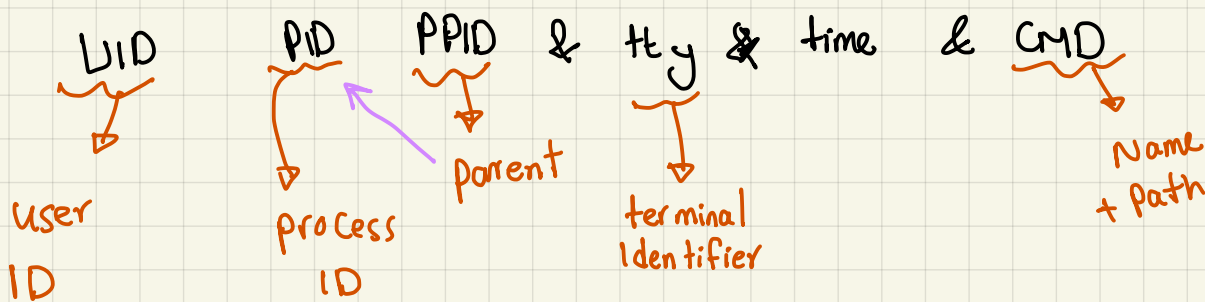
* each process has a unique ID

* when process creates another, the first is the parent process and the second is the child process → the new process

Types of processes

- Shell jobs
- Daemons
- Kernel threads

ps \Rightarrow display processes in the current shell



ps -e \Rightarrow all system processes

ps -f

ps -u username \Rightarrow all processes run by this user

ps -f -u noha \Rightarrow full information about processes run by noha

ps -xf \Rightarrow shows a نظام هرو showing parents full info + all other processes

top \Rightarrow not snapshot \Rightarrow it's live like task manager in windows

ps -a \Rightarrow show all processes in all windows (tabs)

Notes:-

* parent ID = 0 \Rightarrow No parent (kthread & systemd)

pts1 \rightarrow second tab in the terminal

pts0 \rightarrow first tab in the terminal

tty \Rightarrow for terminal

* to search in the manual type /what_you_want_to_search_about then enter

sleep 2000 \$

sleep 5000 \$

ps \Rightarrow

jobs \Rightarrow the running

ctrl + C \Rightarrow kill signal for process

ctrl + Z \Rightarrow stop signal for process

bg %jobid

kill -SIGSTOP %2 \Rightarrow stop it

kill -SIGCONT %2 \Rightarrow continue job that has been started in the back ground

\rightarrow bg %jobid \rightarrow if you start the job in the foreground and want it to continue in background

fg process_name \rightarrow if there is something background and run in foreground

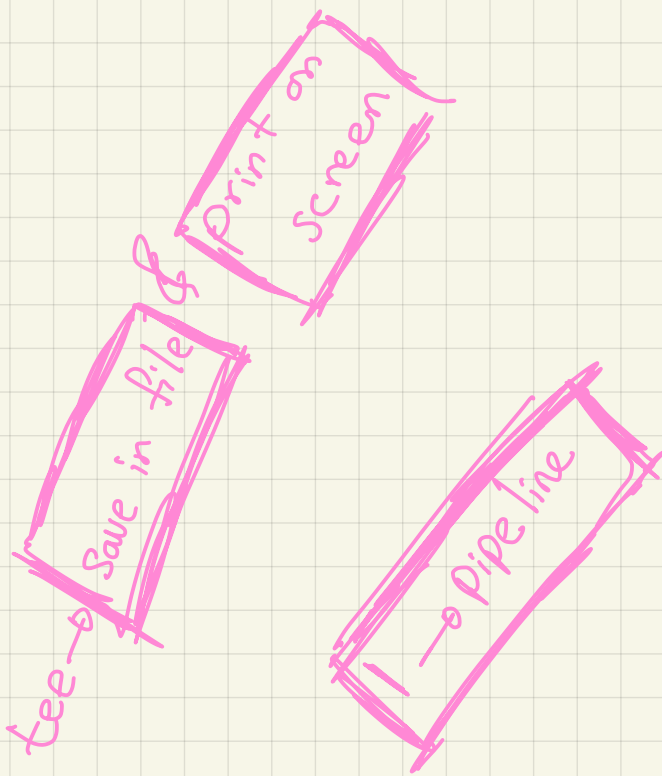
SIGTERM \Rightarrow when used it terminate signal and not able to continue later it will be appeared just first time as terminated then delete it

* pkill process name

we have standard input / output and error
we can re direction them to where we want

to re direct output \Rightarrow `ls -l >> result.txt`

if `>` it will over write but \gg add new data
to old data



Search what difference
between `more` & `less`

wc \Rightarrow word Count (Line, word, char)

wc -l line

wc -c character

wc -w word

diff F1 F2
out put

1c2
└─┘

line1 in file1

Compared by line2 file2