molemahang@molemahang-Latitude-E5550:~$ sleep 300 & sleep 600 & sleep 900 &

[1] 17039

[2] 17040

[3] 17041

//here we had to create some of the background processes using the & sign so that while retrieving the jobs command we can see their display

molemahang@molemahang-Latitude-E5550:~$ jobs

[1] Running sleep 300 &

[2]- Running sleep 600 &

[3]+ Running sleep 900 &

//here is the list of the processes running on the background with some different signs to demostrate their priorities the one with [+] is the one of the higher priority, [-] will be the next process to be brought to the foreground and the one with no sign will be last.

molemahang@molemahang-Latitude-E5550:~$ fg

sleep 900

^Z[1] Done sleep 300

[3]+ Stopped sleep 900

//here process [1] is done with its operations meaning is no longer available on the background and when running the fg the process with [+] is brought on the foreground.

molemahang@molemahang-Latitude-E5550:~$ jobs

[2]- Running sleep 600 &

[3]+ Stopped sleep 900

molemahang@molemahang-Latitude-E5550:~$ fg %2

sleep 600

//here we were specifically bringing process [2] to foreground regardless of the priorities

molemahang@molemahang-Latitude-E5550:~$ bg

[3]+ sleep 900 &

//the process is taken back to the background

molemahang@molemahang-Latitude-E5550:~$ bg

bash: bg: job 3 already in background

molemahang@molemahang-Latitude-E5550:~$ jobs

[3]+ Running sleep 900 &

// And now it is the only one job in the background

molemahang@molemahang-Latitude-E5550:~$ jobs -l

[3]+ 17041 Running sleep 900 &

// this is done to show the <PID>

molemahang@molemahang-Latitude-E5550:~$ kill 17041

// here we kill the process using its ID

molemahang@molemahang-Latitude-E5550:~$ jobs

[3]+ Terminated sleep 900

// the process is be terminated

molemahang@molemahang-Latitude-E5550:~$ while true; do echo "Running..."; sleep 1; done &

[1] 17432

Running...

Running...

Running...

while true; do echo "Running..."; sleep 1; done > output.txt &

Running... while true; do echo "Running..."; sleep 1; done > output.txt &

[2] 17790

// the process is actually running on the background but printing the output on the foreground of the bash and making terminal interactive

molemahang@molemahang-Latitude-E5550:~$ Running...

Running...

Running...

molemahang@molemahang-Latitude-E5550:~$ kill -9 17432

//since the terminal was interactive we had to create new terminal to kill the process

molemahang@molemahang-Latitude-E5550:~$ while true; do echo "Running..."; sleep 1; done > output.txt &

[1] 18267

// we created the process and now directly gave it the order on where to print the output( output.txt)

molemahang@molemahang-Latitude-E5550:~$ kill -9 18267

//now we can interact with the terminal while the infinite loop is still running in the background because the output is delivered to the output.txt no on the console

molemahang@molemahang-Latitude-E5550:~$ jobs

[1]+ Killed while true; do

echo "Running..."; sleep 1;

done > output.txt

molemahang@molemahang-Latitude-E5550:~$

**nohup command**

molemahang@molemahang-Latitude-E5550:~$ nohup ./hello.sh &

[1] 22764

molemahang@molemahang-Latitude-E5550:~$ nohup: ignoring input and appending output to 'nohup.out'

//on the new terminal checking whether the processe still exist

molemahang@molemahang-Latitude-E5550:~$ ps aux | grep hello.sh

molemah+ 22609 0.0 0.0 9972 3456 ? S 22:55 0:00 /bin/bash ./hello.sh

molemah+ 22764 0.0 0.0 9972 3328 ? S 22:56 0:00 /bin/bash ./hello.sh

molemah+ 23121 0.0 0.0 9212 2560 pts/0 S+ 22:59 0:00 grep hello.sh

//displaying what have been printed as output

molemahang@molemahang-Latitude-E5550:~$ cat nohup.out

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

Hello, World!

molemahang@molemahang-Latitude-E5550:~$ kill 22764

molemahang@molemahang-Latitude-E5550:~$ ps aux | grep hello.sh

molemah+ 22609 0.0 0.0 9972 3456 ? S 22:55 0:00 /bin/bash ./hello.sh

molemah+ 23646 0.0 0.0 9212 2560 pts/0 S+ 23:04 0:00 grep hello.sh

molemahang@molemahang-Latitude-E5550:~$ kill 22609

molemahang@molemahang-Latitude-E5550:~$ ps aux | grep hello.sh

molemah+ 23698 0.0 0.0 9212 2560 pts/0 S+ 23:04 0:00 grep hello.sh

molemahang@molemahang-Latitude-E5550:~$ kill -9 22609

bash: kill: (22609) - No such process

**exit**

//while entering this command and hitting enter the terminal is immediately terminated

**tail command**

**//**creating the file

molemahang@molemahang-Latitude-E5550:~$ echo "This is line 1" > example.txt

echo "This is line 2" >> example.txt

echo "This is line 3" >> example.txt

echo "This is line 4" >> example.txt

echo "This is line 5" >> example.txt

echo "This is line 6" >> example.txt

echo "This is line 7" >> example.txt

echo "This is line 8" >> example.txt

echo "This is line 9" >> example.txt

echo "This is line 10" >> example.txt

echo "This is line 11" >> example.txt

echo "This is line 12" >> example.txt

//printing the last 10 lines

molemahang@molemahang-Latitude-E5550:~$ tail example.txt

This is line 3

This is line 4

This is line 5

This is line 6

This is line 7

This is line 8

This is line 9

This is line 10

This is line 11

This is line 12

//stipulating number of line to print

molemahang@molemahang-Latitude-E5550:~$ tail -n 5 example.txt

This is line 8

This is line 9

This is line 10

This is line 11

This is line 12

molemahang@molemahang-Latitude-E5550:~$ tail -f example.txt

This is line 3

This is line 4

This is line 5

This is line 6

This is line 7

This is line 8

This is line 9

This is line 10

This is line 11

This is line 12

echo "this is the line 13">>example.txt

echo "this is the line 14">>example.txt

^Z

[1]+ Stopped tail -f example.txt

molemahang@molemahang-Latitude-E5550:~$ tail example.txt

This is line 3

This is line 4

This is line 5

This is line 6

This is line 7

This is line 8

This is line 9

This is line 10

This is line 11

This is line 12

**watch command**

molemahang@molemahang-Latitude-E5550:~$ watch date

[2]+ Stopped watch date

//here we stipulated our interval rather than using default 2 seconds

molemahang@molemahang-Latitude-E5550:~$ watch -n 5 date

Every 5.0s: date molemahang-Latitude-E5550: Sun Oct 27 23:28:47 2024

Sun 27 Oct 2024 23:28:47 SAST

//here it displays the output differences that are occurring by displaying animation

molemahang@molemahang-Latitude-E5550:~$ watch -d date

[4]+ Stopped watch -d date

Every 2.0s: date molemahang-Latitude-E5550: Sun Oct 27 23:32:51 2024

Sun 27 Oct 2024 23:32:51 SAST

//here the time,number of user and the load from a minute, 5 and 15 are also shown and can see that the computer is not overloaded as all the value still reflect below 1

molemahang@molemahang-Latitude-E5550:~$ uptime

23:38:10 up 1 day, 10:47, 1 user, load average: 0.70, 0.52, 0.43

molemahang@molemahang-Latitude-E5550:~$