**Practical : 11**

**Practical : 11(a)**

* **Aim :** catch the terminal interrupts by using appropriate signals.
* **Code :**

#include <stdio.h>

#include <signal.h>

static void myHandler(int iSig){

printf("Caught signal is: %d\n", iSig);

}

void sig\_handler(int signum)

{

printf("Alarm single is caught\n");

}

int main(void){

signal(SIGINT, myHandler);

signal(SIGTSTP, myHandler);

signal(SIGALRM,sig\_handler);

alarm(2);

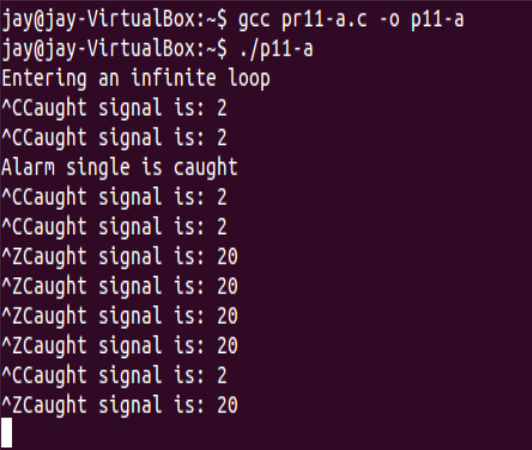
printf("Entering an infinite loop\n");

for (;;);

return 0;

}

* **Output :**



**Practical : 11(b)**

* **Aim :** Schedule a signal for a process.
* **Code :**

#include <stdio.h>

#include <signal.h>

void sig\_handler(int signum)

{

printf("Alarm single is caught\n");

}

int main(void){

signal(SIGALRM,sig\_handler);

alarm(3);

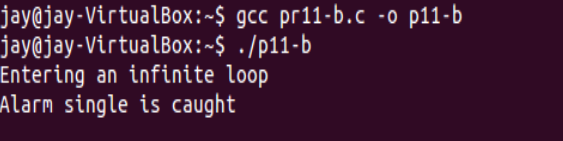
printf("Entering an infinite loop\n");

for (;;);

return 0;

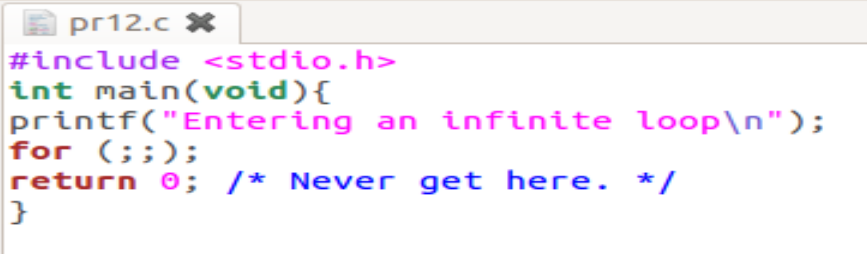
}

* **Output :**

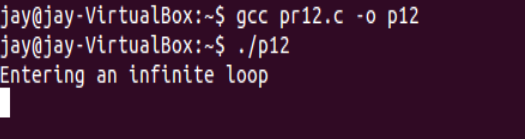


**Practical : 11(b)**

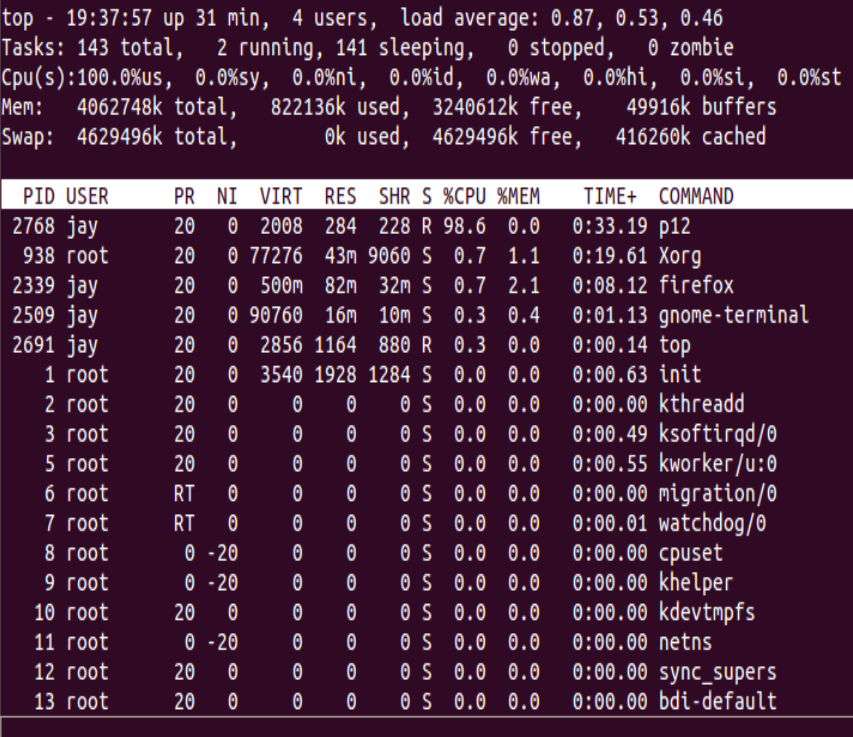
* **Aim :** Kill the process using signals.
* **Code with output :**



* Run the program

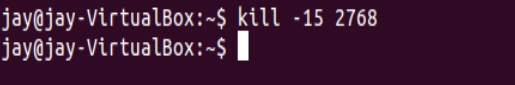


* Here we can see that there is a process name p12 with process id “2768”

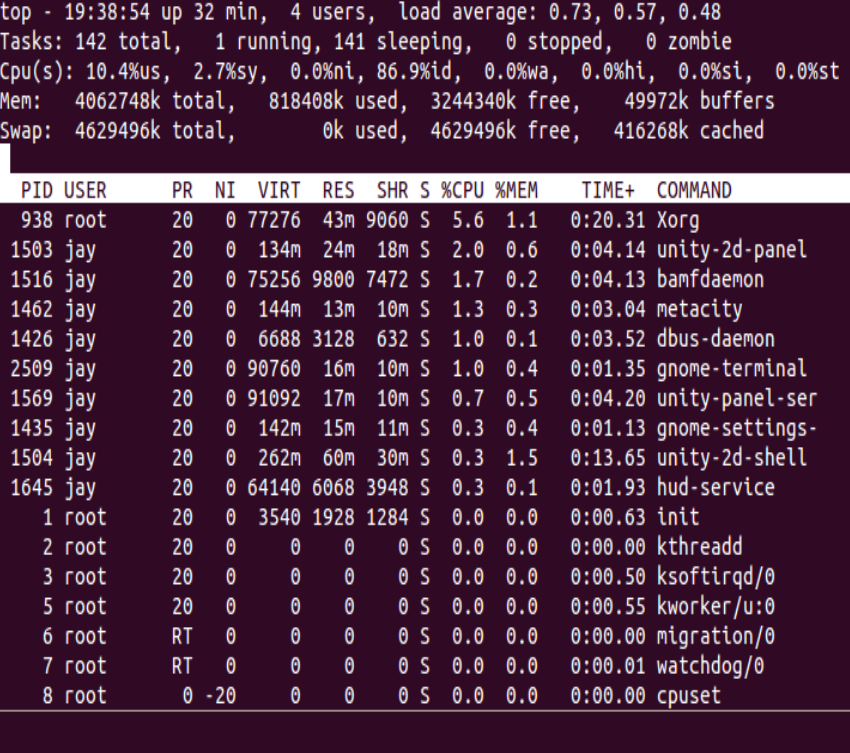


* Use Kill command to kill the process

Kill -15 [process\_id]



* We can see that there is no any process with name p12



**Assignment : 9**

* **Aim :**

This program should count the number of times the user has sent the SIGINT signal to the running process. Pressing Ctl-C from the keyboard send this signal. When the process receives the SIGTSTP signal (Ctl-Z), it should print to the terminal the number of SIGINT signals it has received. After it receives 5 SIGINT signals, the program should prompt the user to exit. If the user does not respond within 10 seconds, an SIGALRM signal should force the program to exit.

* **Code :**

#include <stdio.h>

#include<stdlib.h>

#include <signal.h>

#include<stdbool.h>

int sigc=0;

bool flag=false;

void static sigAlrm(int signum);

static void myHandlerC(int iSig){

printf("Caught signal is: %d\n", iSig);

sigc++;

if(sigc==5){

signal(SIGALRM,sigAlrm);

printf("\nplease give response in 10s\n");

alarm(6);

wait(5000);

}

if(sigc==6)

flag=true;

if(sigc==10)

exit (0);

}

static void myHandlerP(int iSig){

printf("Received SIGINT number : %d\n", sigc);

}

static void start(){

signal(SIGINT, myHandlerC);

signal(SIGTSTP, myHandlerP);

}

void static sigAlrm(int signum){

if(flag){

printf("\nReceived Response..\n");

alarm(0);

flag=false;

start();}

else

exit (0);

}

int main(void){

start();

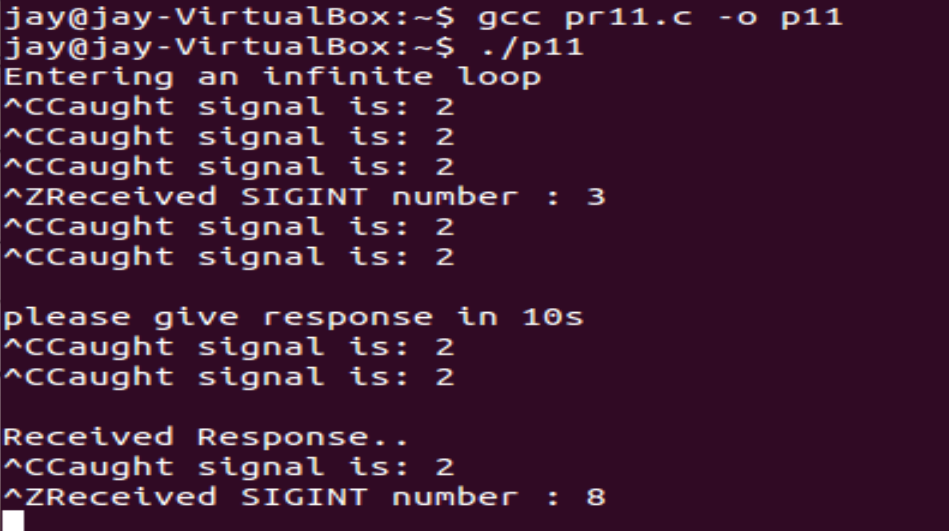
printf("Entering an infinite loop\n");

for (;;);

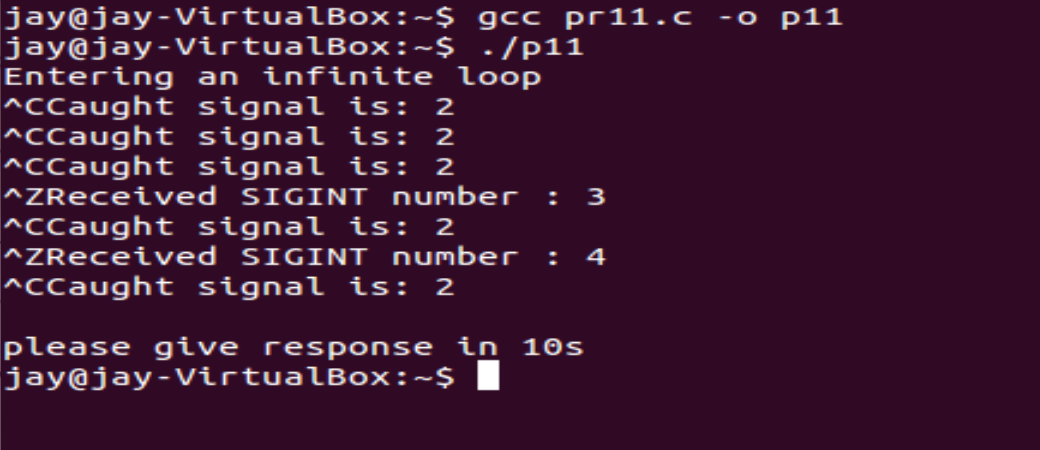
return 0;

}

* **Output :**
* After 5 SIGINT single If user response within 10 sec then user can continue.



* After 5 SIGINT single If user not response within 10 sec then user will exit.

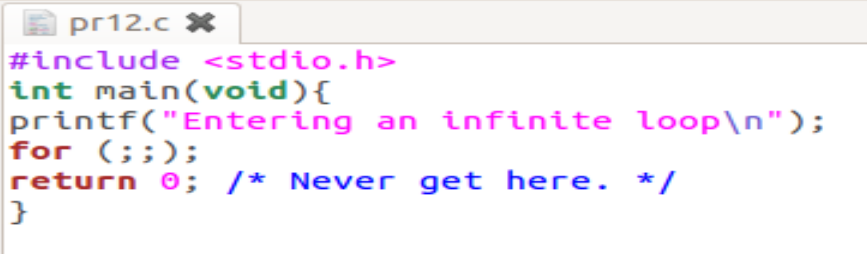


**Practical : 12 with Assignment : 10**

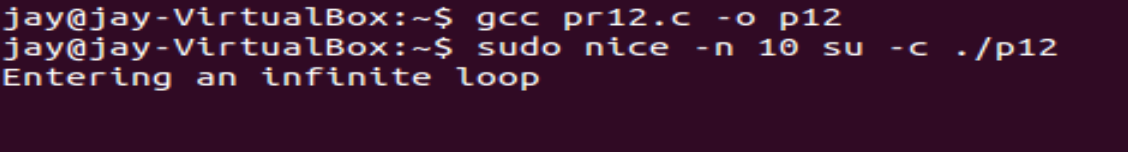
* **Aim :**

Create an infinite running process. Initialize the process with 10 as nice value and after the process starts change it by 15.

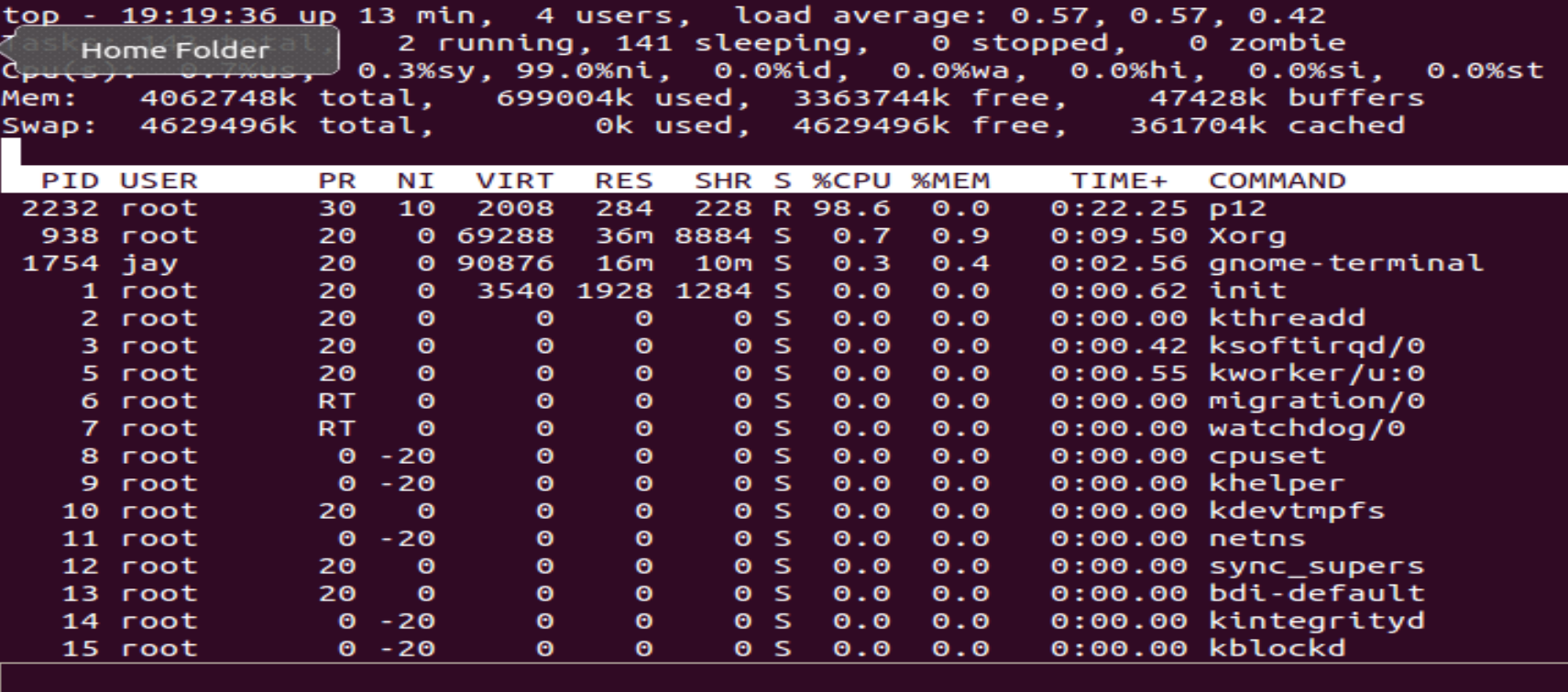
* **Code :**



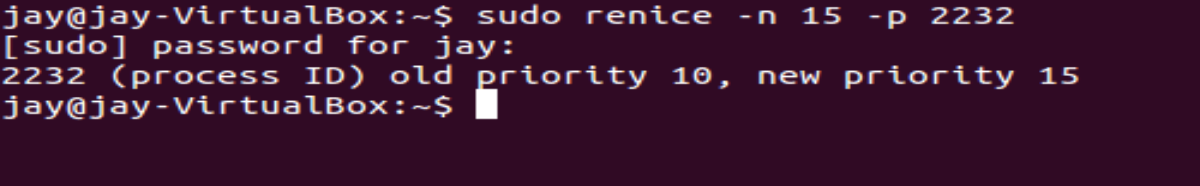
* **Output :**
* Run the process with nice value 10



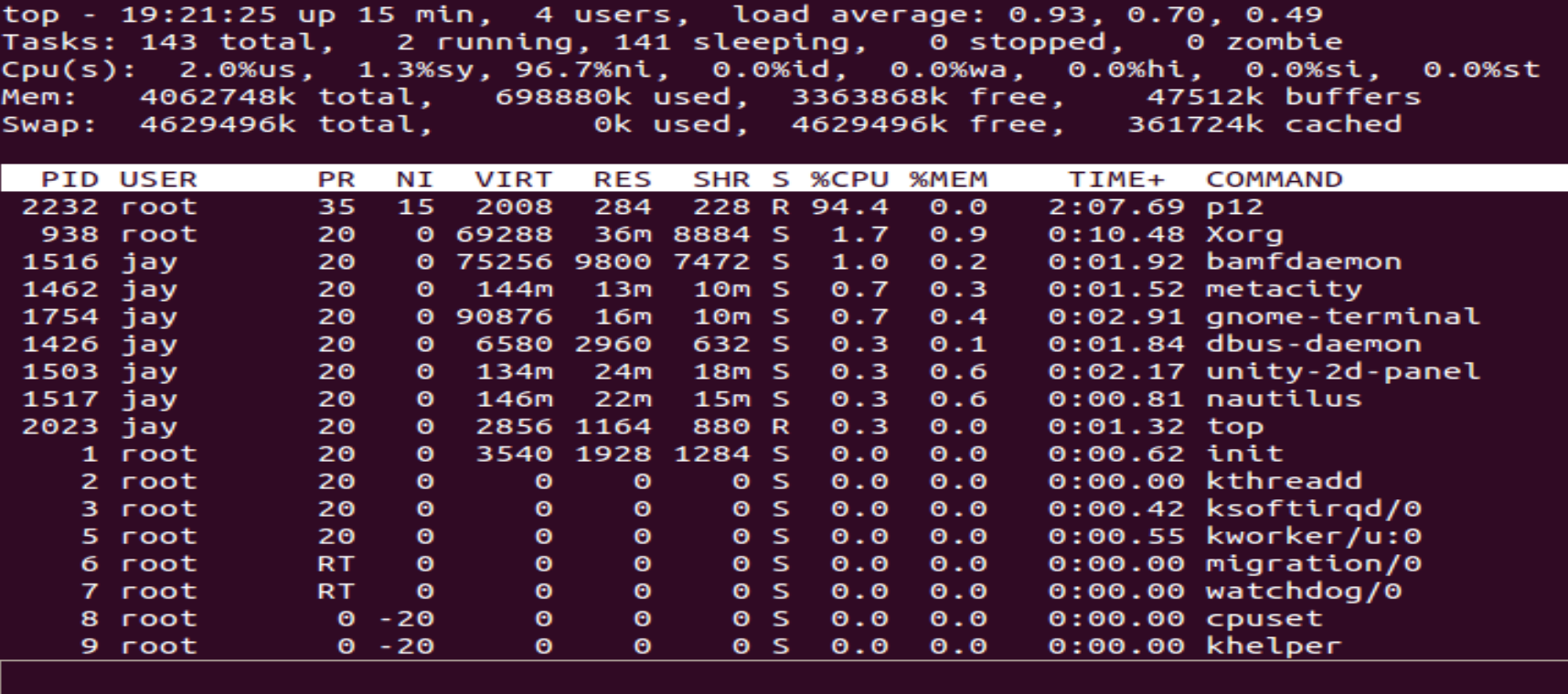
* We can see that process “p12” with process id “2232“ and nice value “10“



* Change the Renice “15” value with below command



* We can see that nice value of process “p12” has changed into 15



**Conclusion :**

In this practical we have learned about various like SIGINT , SIGTSTP , SIGALRM and learned about nice value of process and kill process using single and also implemented it.