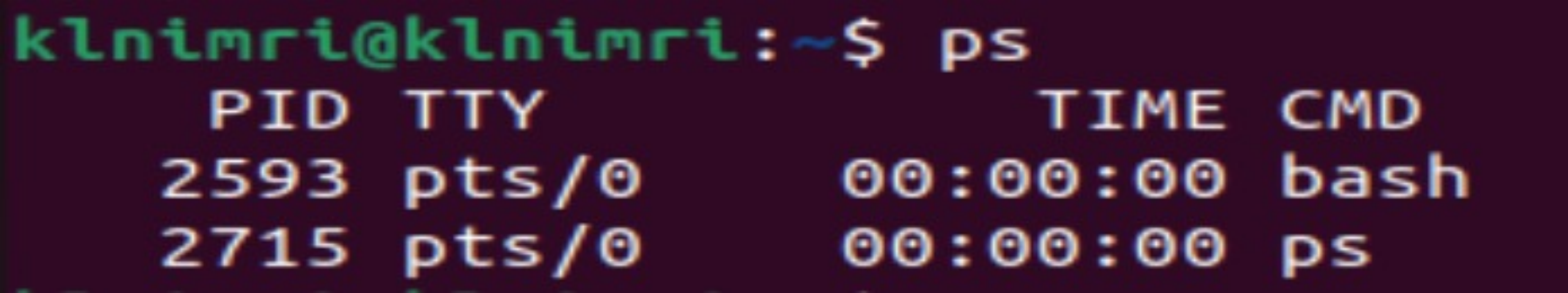
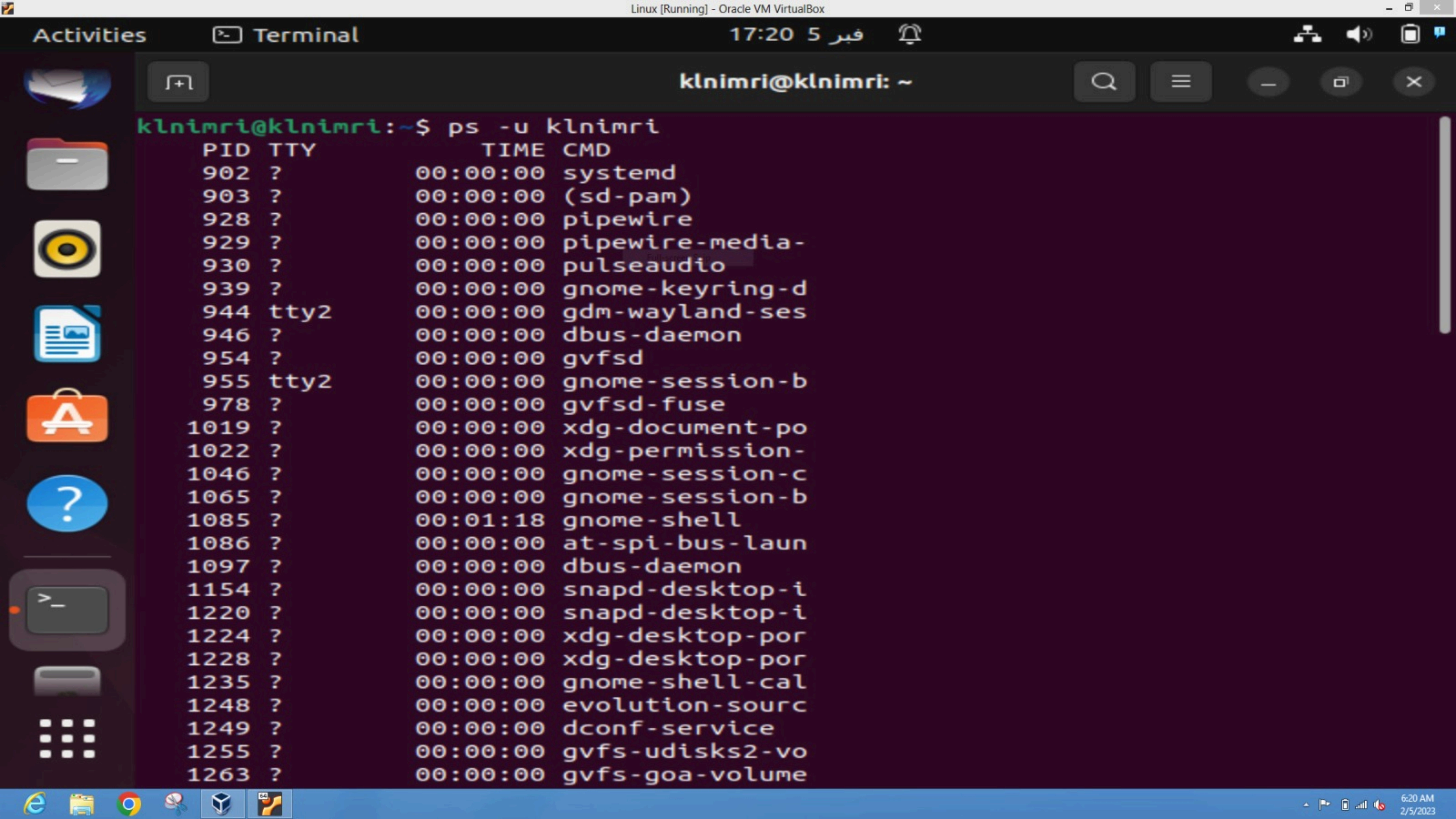
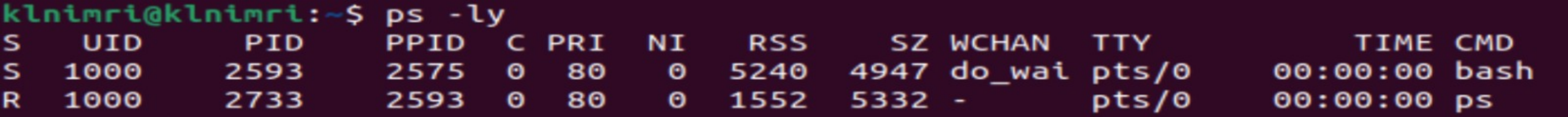


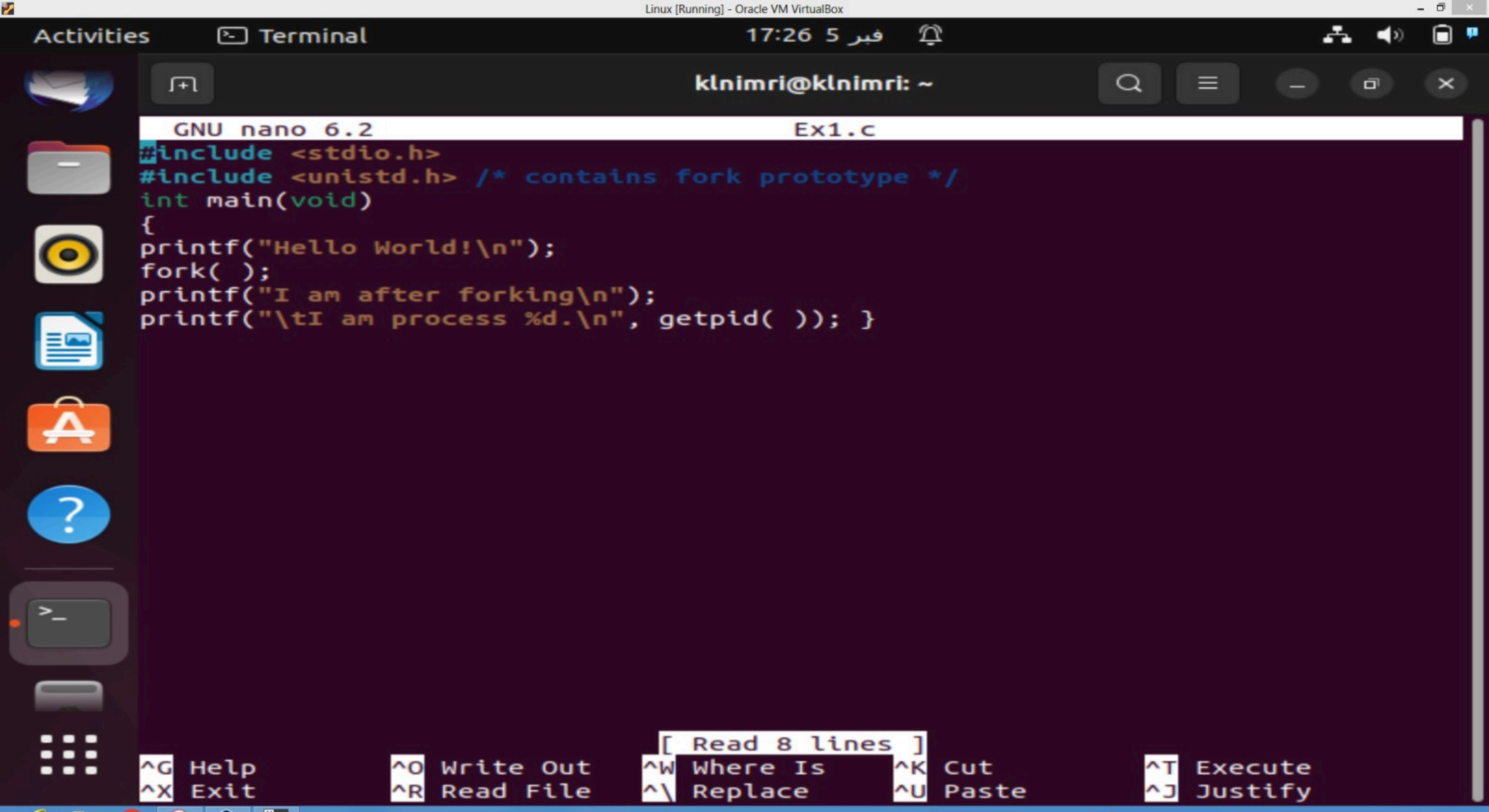
```
klnimri@klnimri:~$ ps
    PID TTY          TIME CMD
   2593 pts/0    00:00:00 bash
   2715 pts/0    00:00:00 ps
```



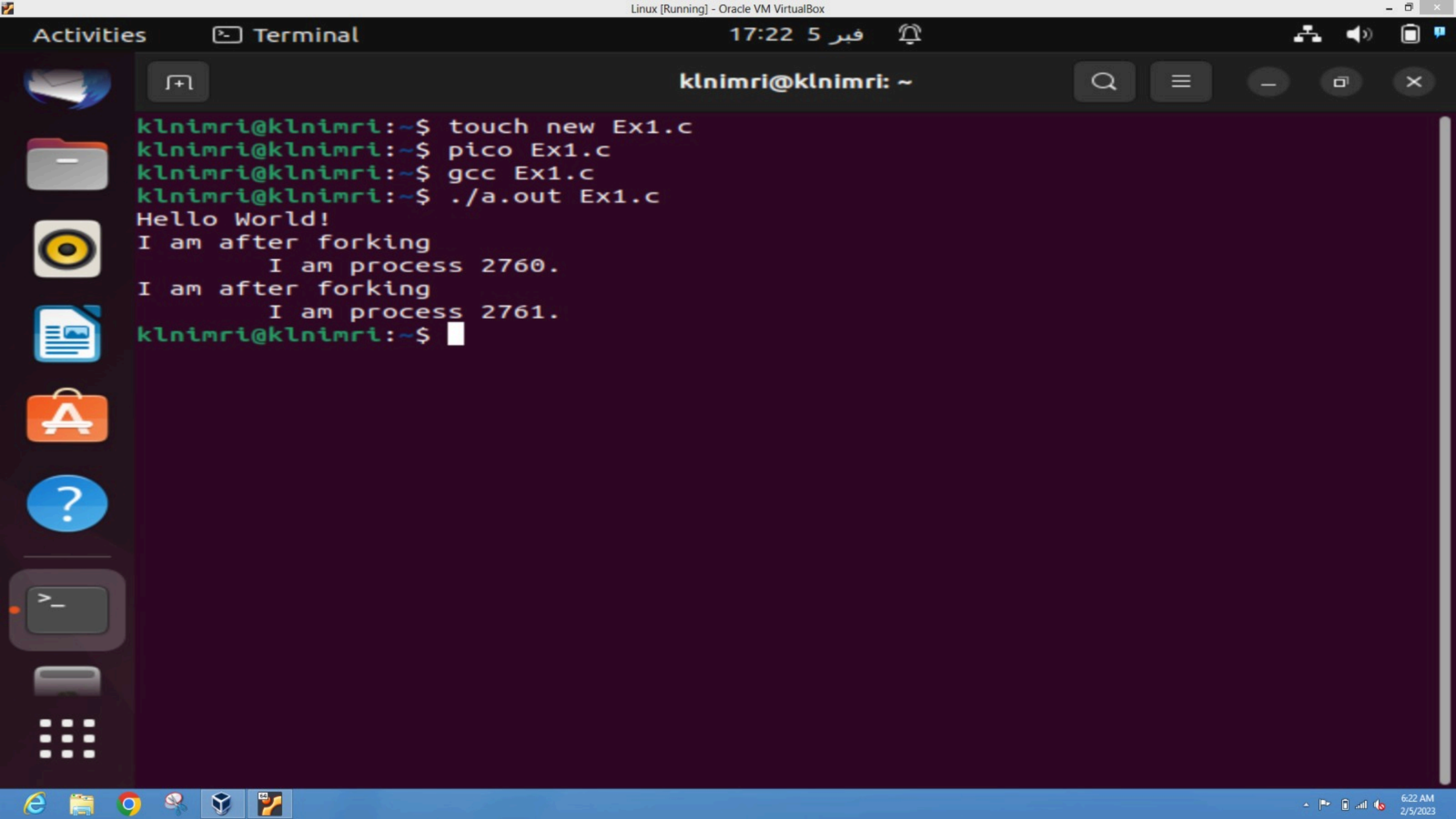
```
klnimri@klnimri:~$ ps -u klnimri
    PID TTY          TIME CMD
    902 ?        00:00:00 systemd
    903 ?        00:00:00 (sd-pam)
    928 ?        00:00:00 pipewire
    929 ?        00:00:00 pipewire-media-
    930 ?        00:00:00 pulseaudio
    939 ?        00:00:00 gnome-keyring-d
    944 tty2     00:00:00 gdm-wayland-ses
    946 ?        00:00:00 dbus-daemon
    954 ?        00:00:00 gvfsd
    955 tty2     00:00:00 gnome-session-b
    978 ?        00:00:00 gvfsd-fuse
   1019 ?        00:00:00 xdg-document-po
   1022 ?        00:00:00 xdg-permission-
   1046 ?        00:00:00 gnome-session-c
   1065 ?        00:00:00 gnome-session-b
   1085 ?        00:01:18 gnome-shell
   1086 ?        00:00:00 at-spi-bus-laun
   1097 ?        00:00:00 dbus-daemon
   1154 ?        00:00:00 snapd-desktop-i
   1220 ?        00:00:00 snapd-desktop-i
   1224 ?        00:00:00 xdg-desktop-por
   1228 ?        00:00:00 xdg-desktop-por
   1235 ?        00:00:00 gnome-shell-cal
   1248 ?        00:00:00 evolution-sourc
   1249 ?        00:00:00 dconf-service
   1255 ?        00:00:00 gvfs-udisks2-vo
   1263 ?        00:00:00 gvfs-goa-volume
```



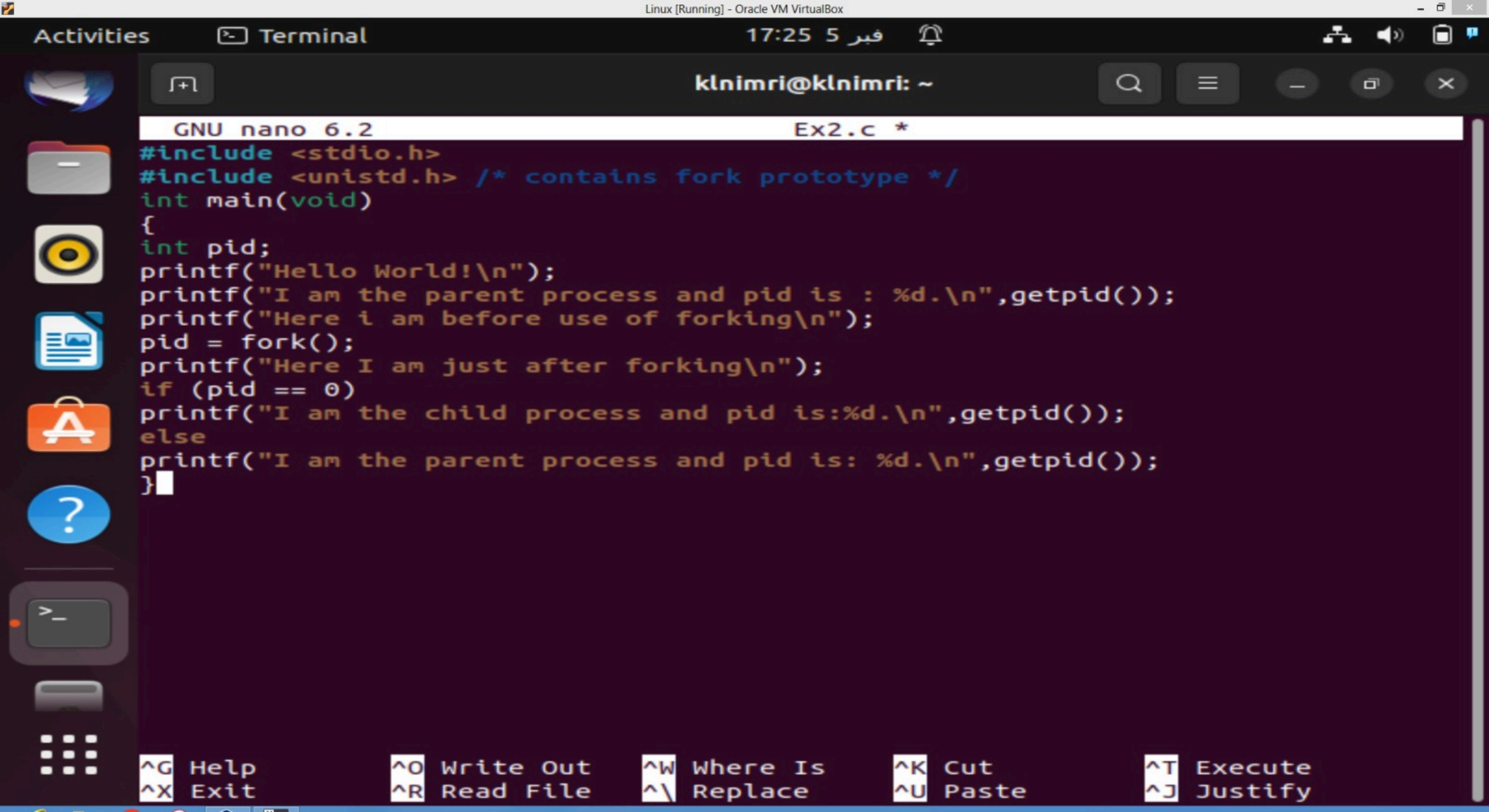
```
klnimri@klnimri:~$ ps -ly
S   UID     PID    PPID  C PRI  NI   RSS    SZ WCHAN  TTY          TIME CMD
S  1000    2593    2575  0  80   0  5240  4947 do_wai pts/0    00:00:00 bash
R  1000    2733    2593  0  80   0  1552  5332 -      pts/0    00:00:00 ps
```



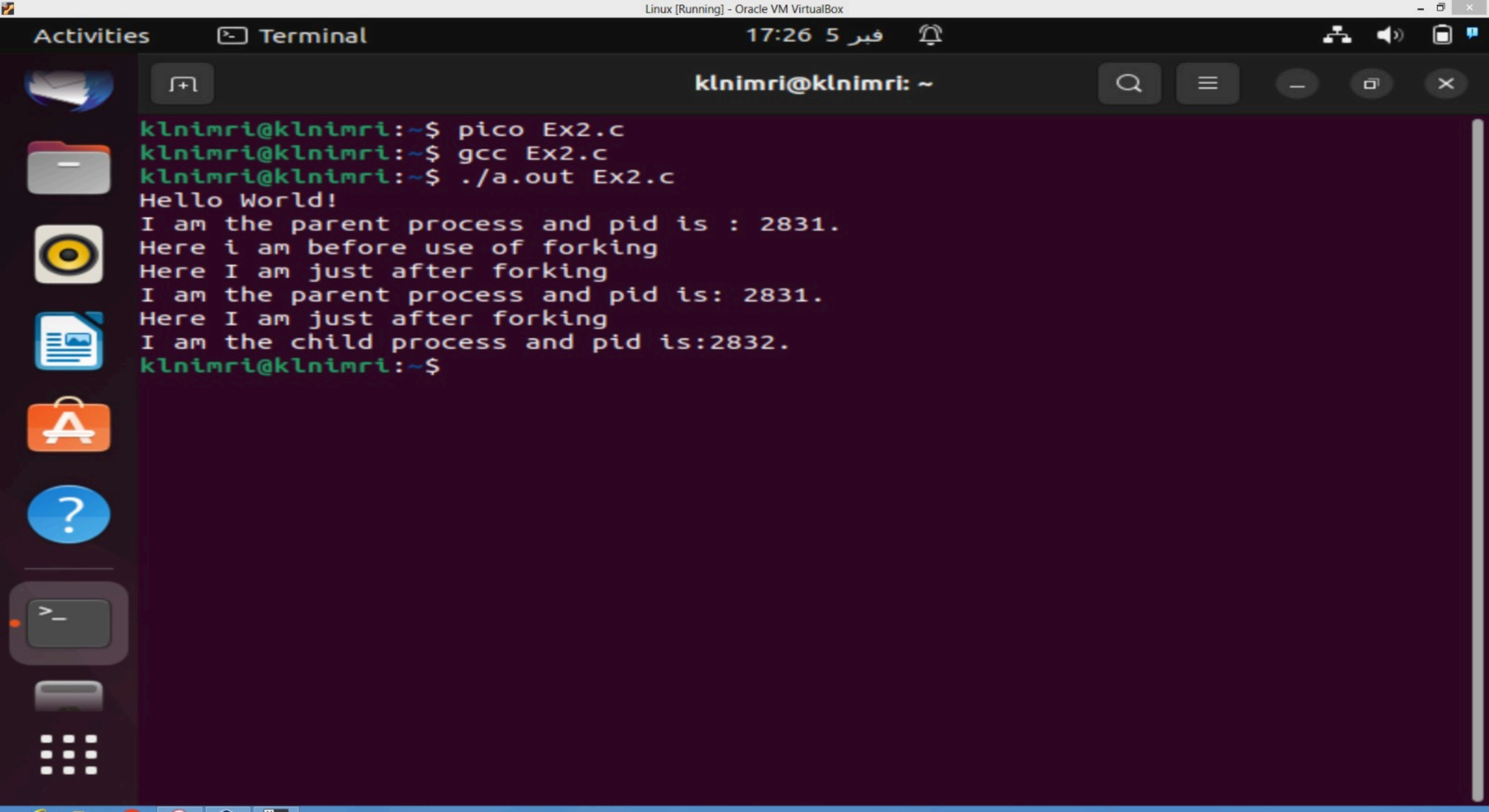
```
  GNU nano 6.2                                    Ex1.c
#include <stdio.h>
#include <unistd.h> /* contains fork prototype */
int main(void)
{
printf("Hello World!\n");
fork( );
printf("I am after forking\n");
printf("\tI am process %d.\n", getpid( )); }

                                   [ Read 8 lines ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```



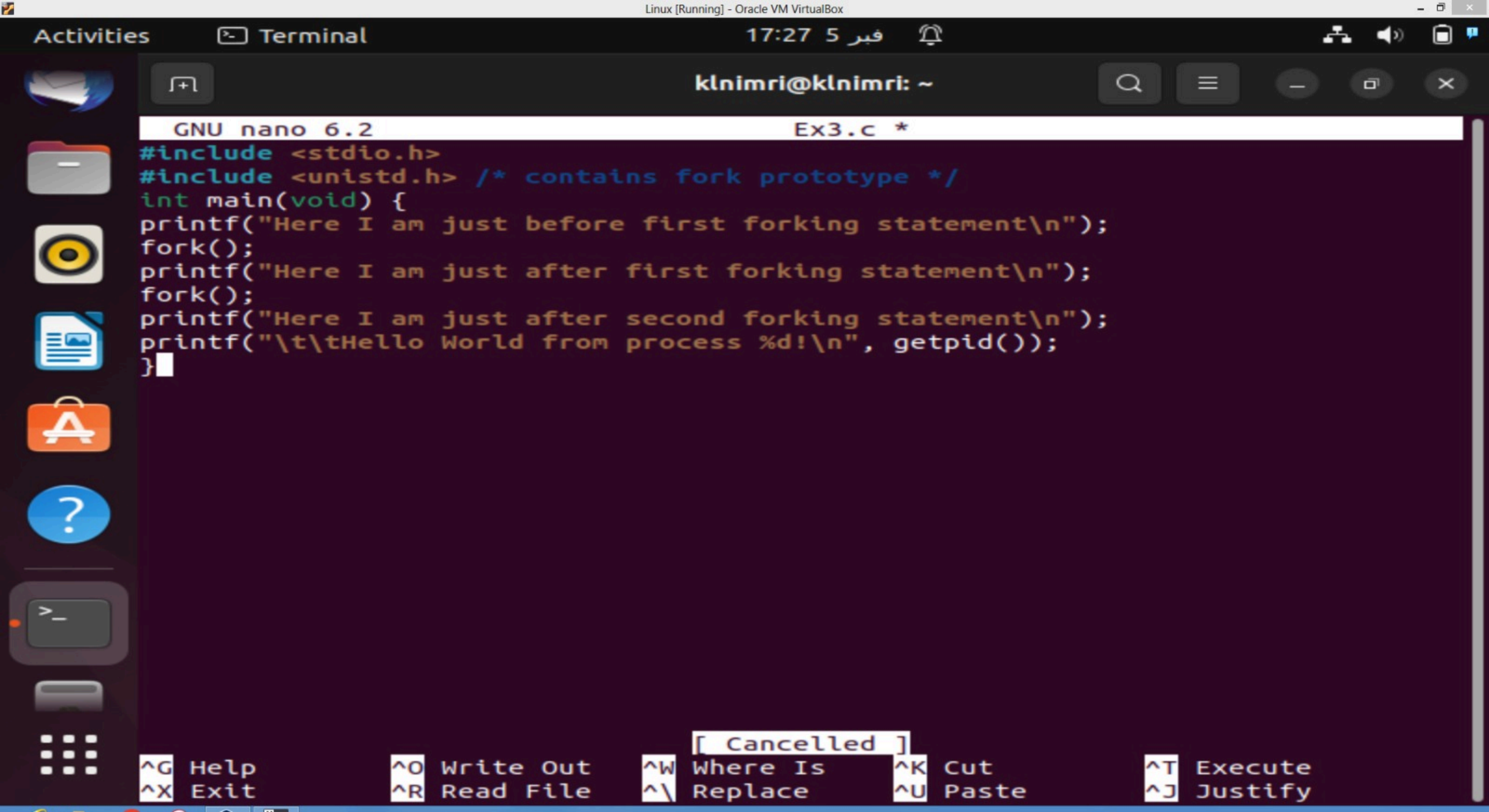
```
klnimri@klnimri:~$ touch new Ex1.c
klnimri@klnimri:~$ pico Ex1.c
klnimri@klnimri:~$ gcc Ex1.c
klnimri@klnimri:~$ ./a.out Ex1.c
Hello World!
I am after forking
        I am process 2760.
I am after forking
        I am process 2761.
klnimri@klnimri:~$ 
```



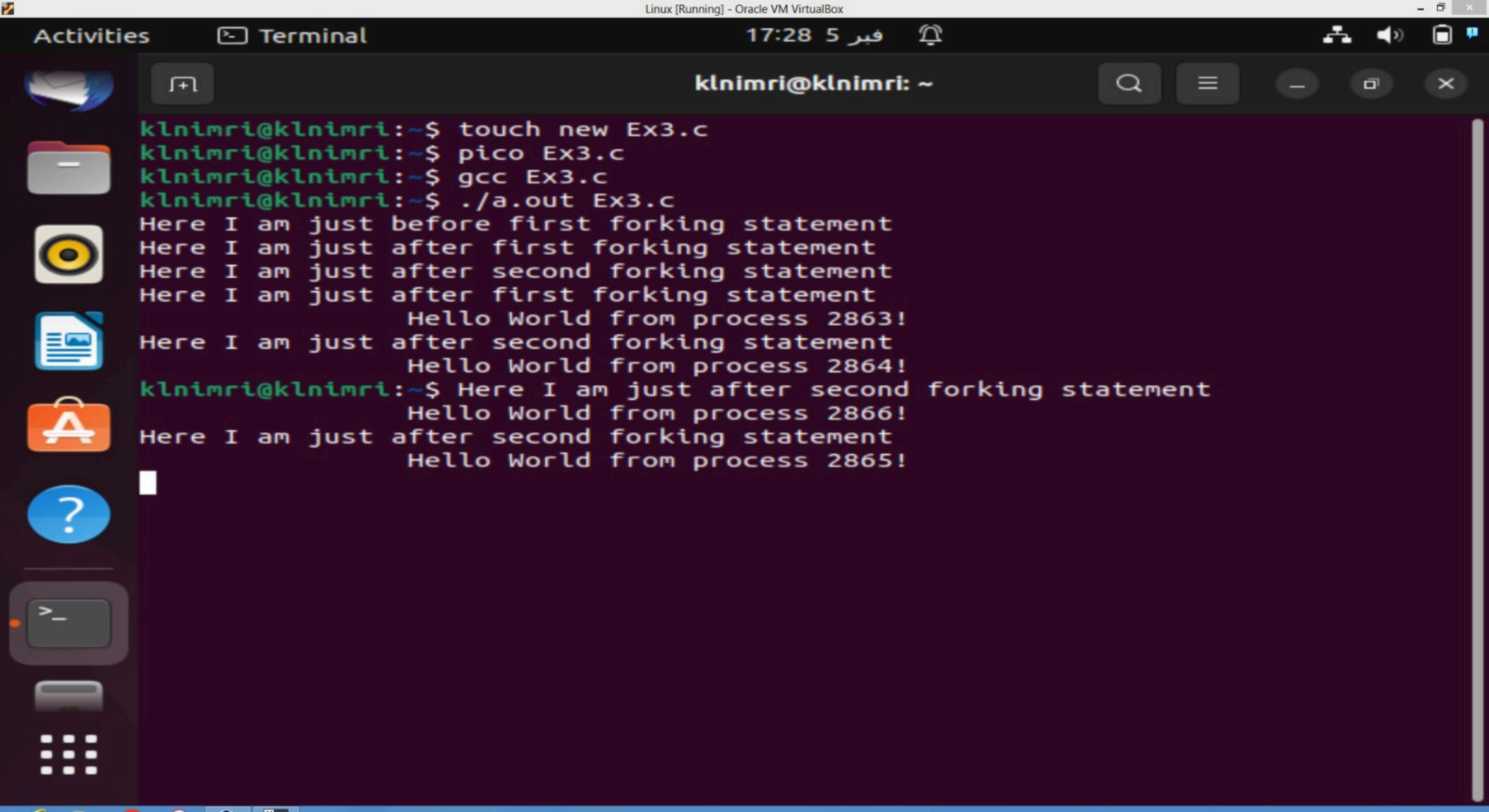
```
  GNU nano 6.2                              Ex2.c *
#include <stdio.h>
#include <unistd.h> /* contains fork prototype */
int main(void)
{
int pid;
printf("Hello World!\n");
printf("I am the parent process and pid is : %d.\n",getpid());
printf("Here i am before use of forking\n");
pid = fork();
printf("Here I am just after forking\n");
if (pid == 0)
printf("I am the child process and pid is:%d.\n",getpid());
else
printf("I am the parent process and pid is: %d.\n",getpid());
}
```



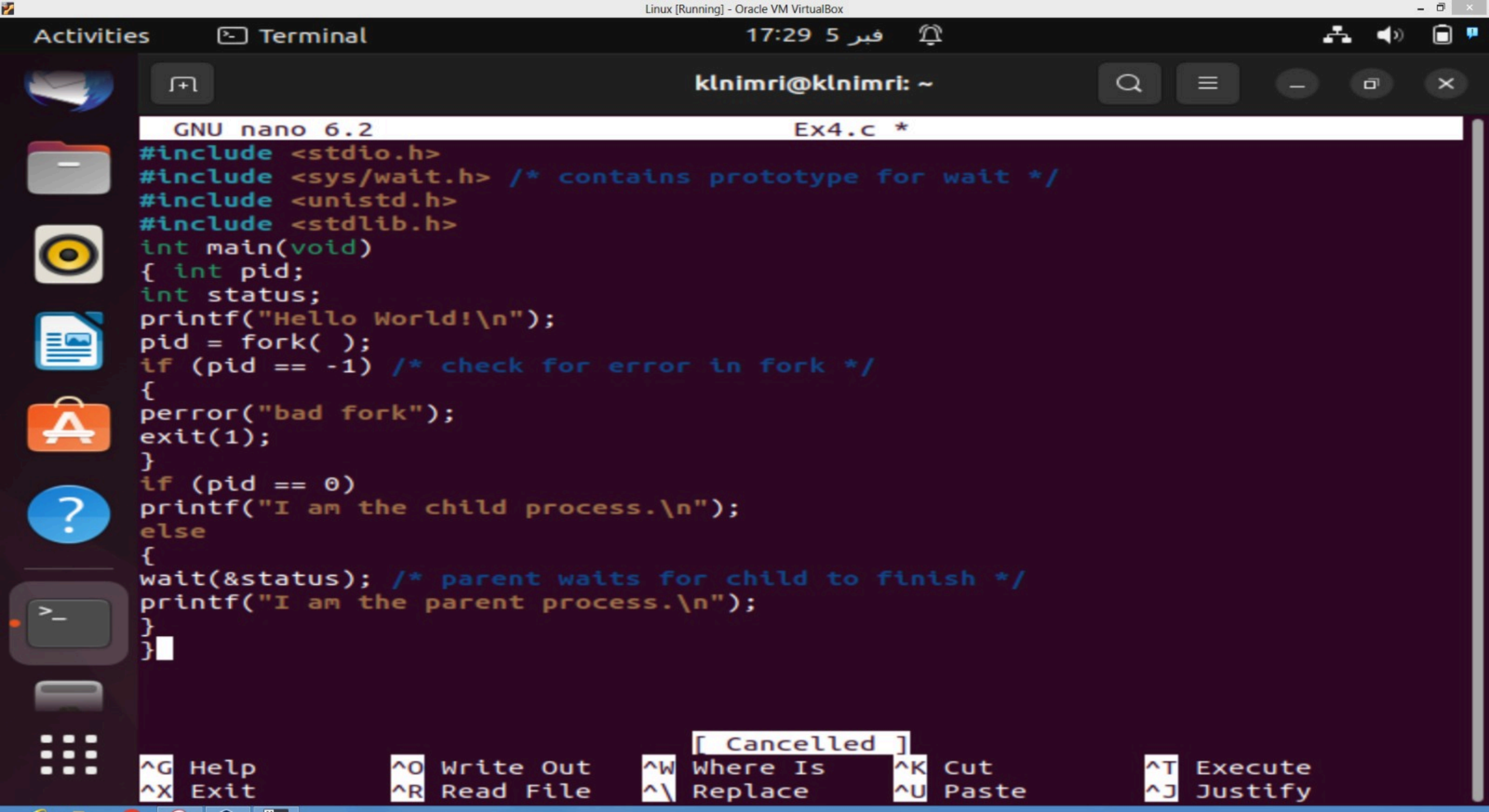
```
klnimri@klnimri:~$ pico Ex2.c
klnimri@klnimri:~$ gcc Ex2.c
klnimri@klnimri:~$ ./a.out Ex2.c
Hello World!
I am the parent process and pid is : 2831.
Here i am before use of forking
Here I am just after forking
I am the parent process and pid is: 2831.
Here I am just after forking
I am the child process and pid is:2832.
klnimri@klnimri:~$
```



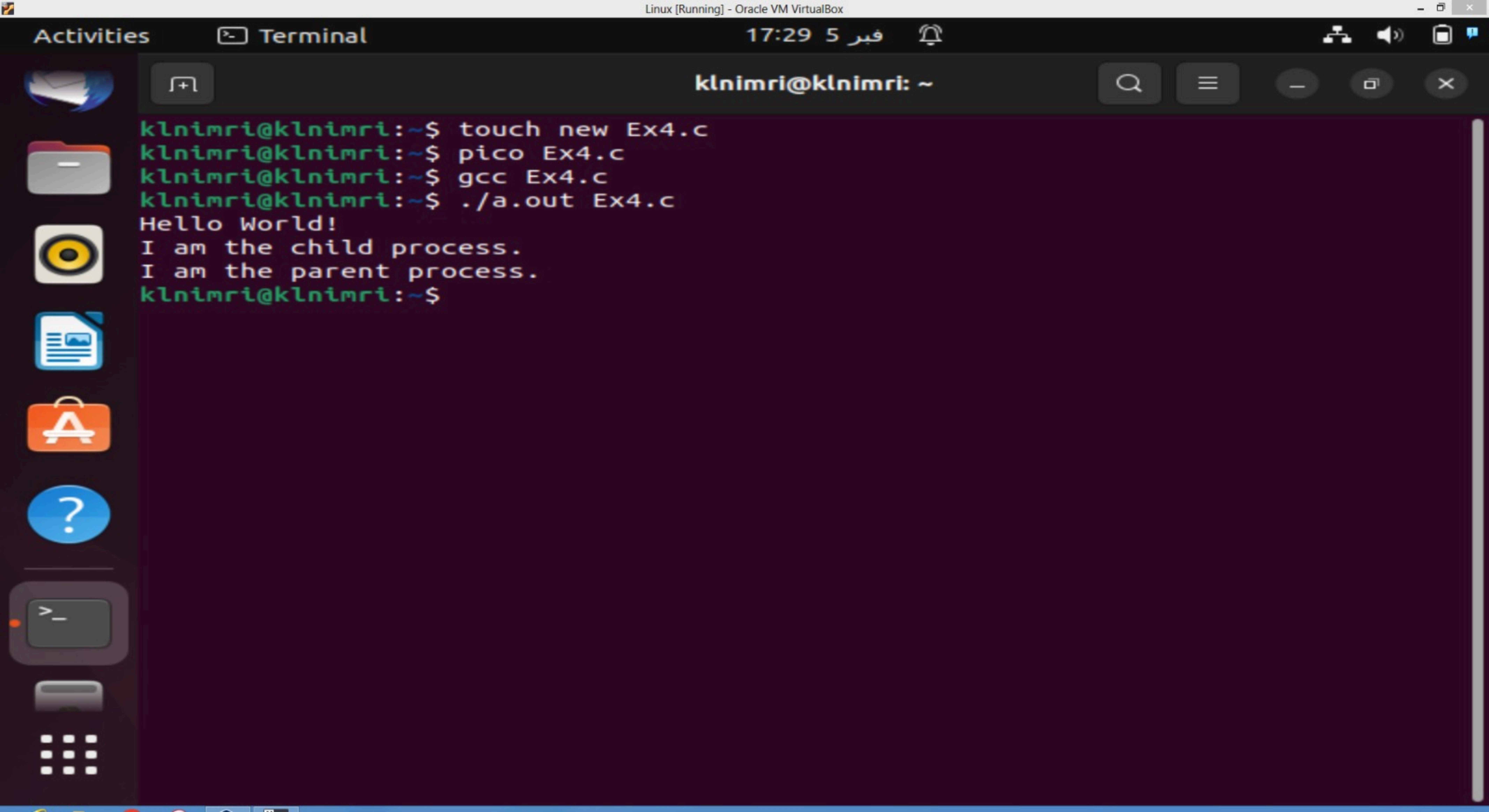
```
  GNU nano 6.2                        Ex3.c *
#include <stdio.h>
#include <unistd.h> /* contains fork prototype */
int main(void) {
printf("Here I am just before first forking statement\n");
fork();
printf("Here I am just after first forking statement\n");
fork();
printf("Here I am just after second forking statement\n");
printf("\t\tHello World from process %d!\n", getpid());
}





                                  [ Cancelled ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```



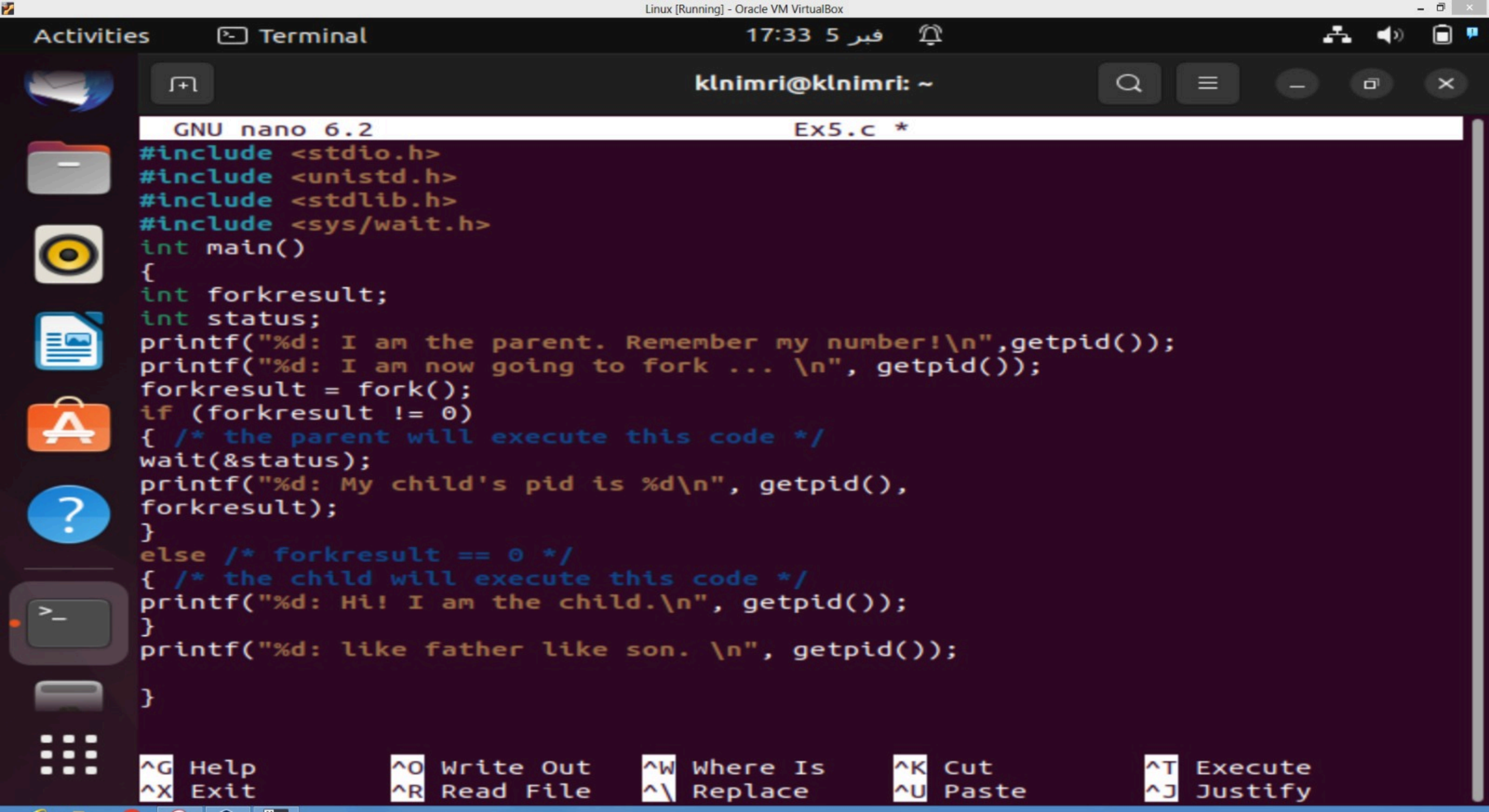
```
klnimri@klnimri:~$ touch new Ex3.c
klnimri@klnimri:~$ pico Ex3.c
klnimri@klnimri:~$ gcc Ex3.c
klnimri@klnimri:~$ ./a.out Ex3.c
Here I am just before first forking statement
Here I am just after first forking statement
Here I am just after second forking statement
Here I am just after first forking statement
                Hello World from process 2863!
Here I am just after second forking statement
                Hello World from process 2864!
klnimri@klnimri:~$ Here I am just after second forking statement
                Hello World from process 2866!
Here I am just after second forking statement
                Hello World from process 2865!
```



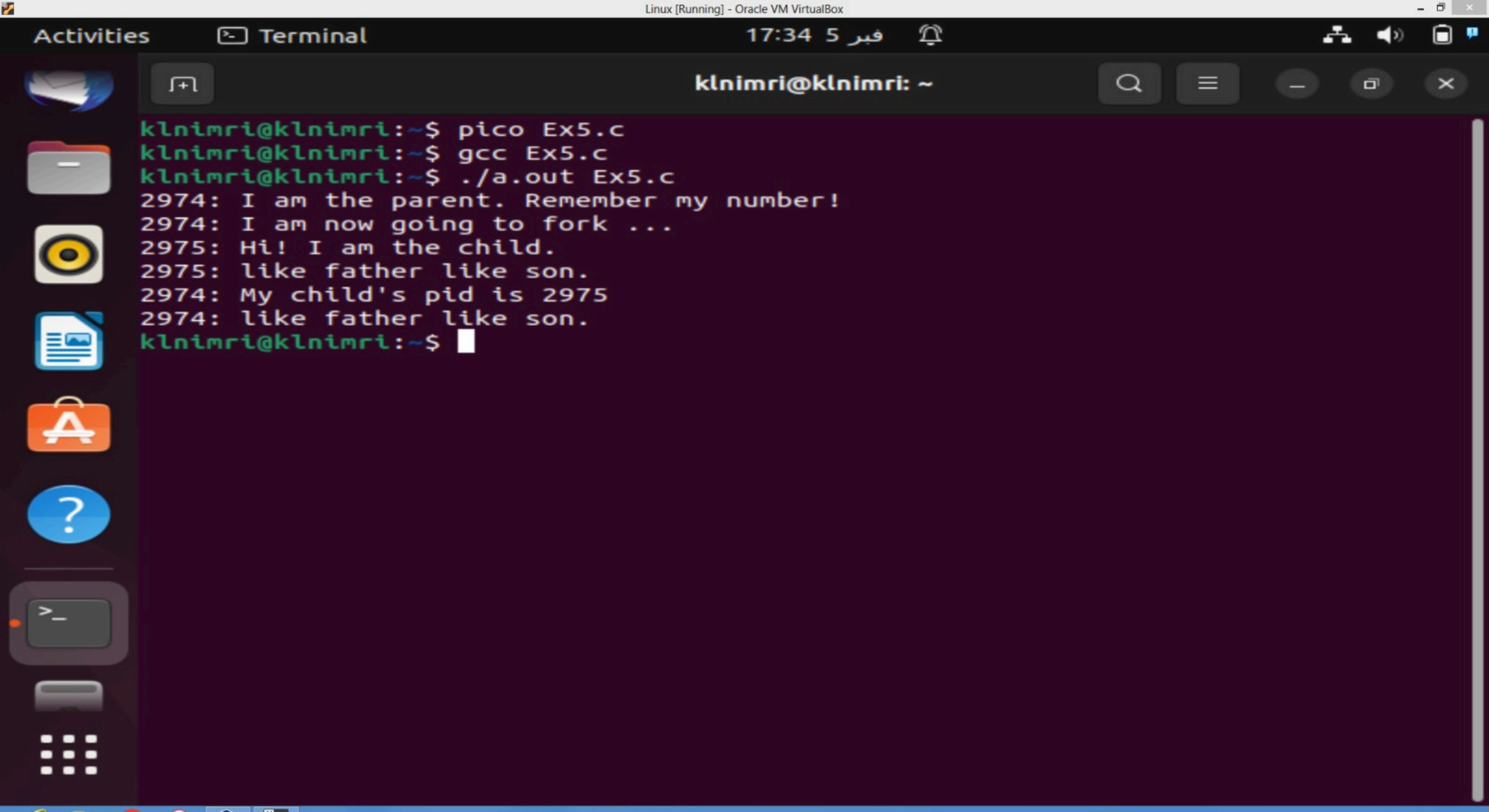
```
GNU nano 6.2                                    Ex4.c *
#include <stdio.h>
#include <sys/wait.h> /* contains prototype for wait */
#include <unistd.h>
#include <stdlib.h>
int main(void)
{ int pid;
int status;
printf("Hello World!\n");
pid = fork( );
if (pid == -1) /* check for error in fork */
{
perror("bad fork");
exit(1);
}
if (pid == 0)
printf("I am the child process.\n");
else
{
wait(&status); /* parent waits for child to finish */
printf("I am the parent process.\n");
}
}

                                        [ Cancelled ]
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
```



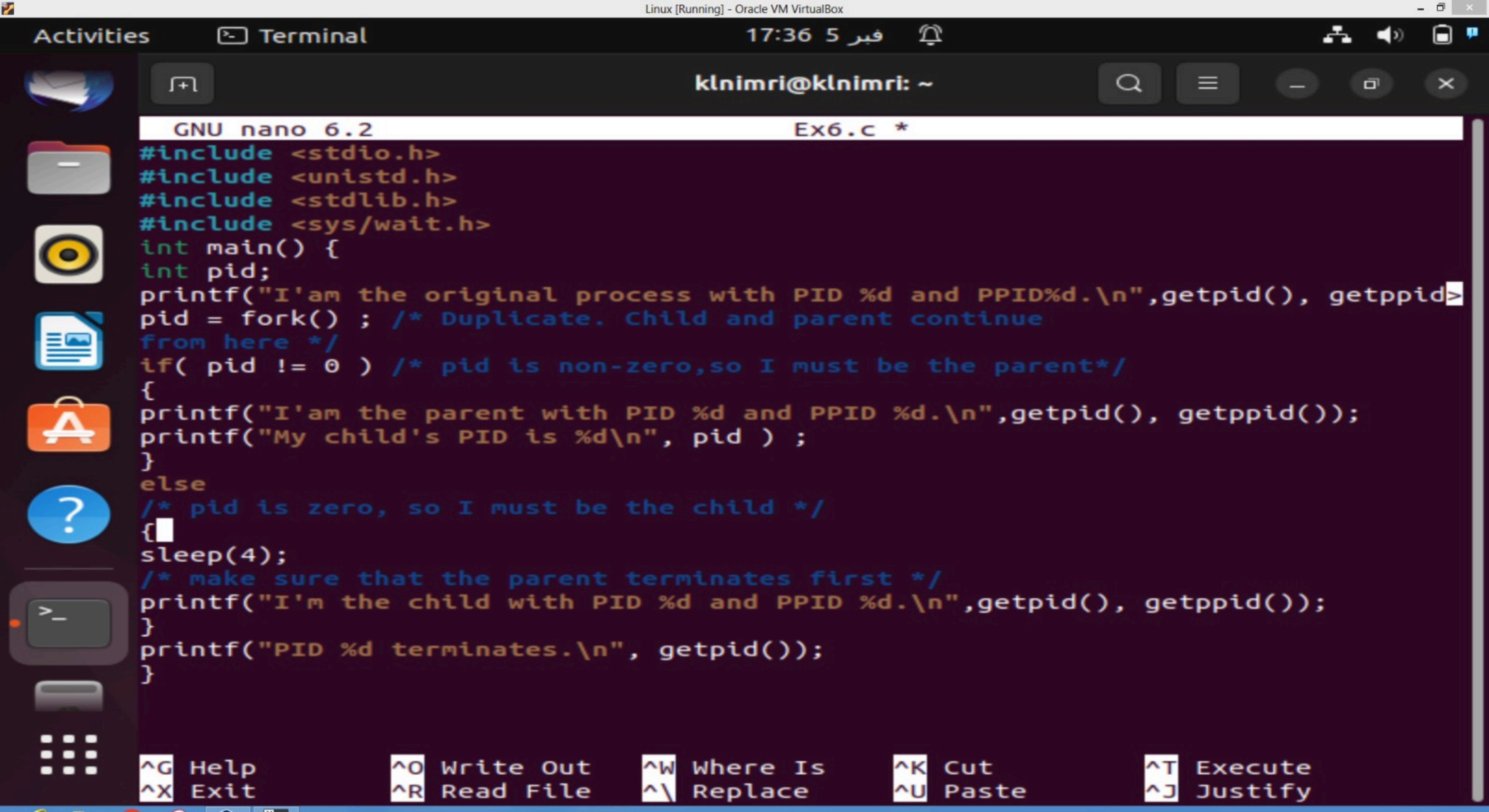
```
klnimri@klnimri:~$ touch new Ex4.c
klnimri@klnimri:~$ pico Ex4.c
klnimri@klnimri:~$ gcc Ex4.c
klnimri@klnimri:~$ ./a.out Ex4.c
Hello World!
I am the child process.
I am the parent process.
klnimri@klnimri:~$
```



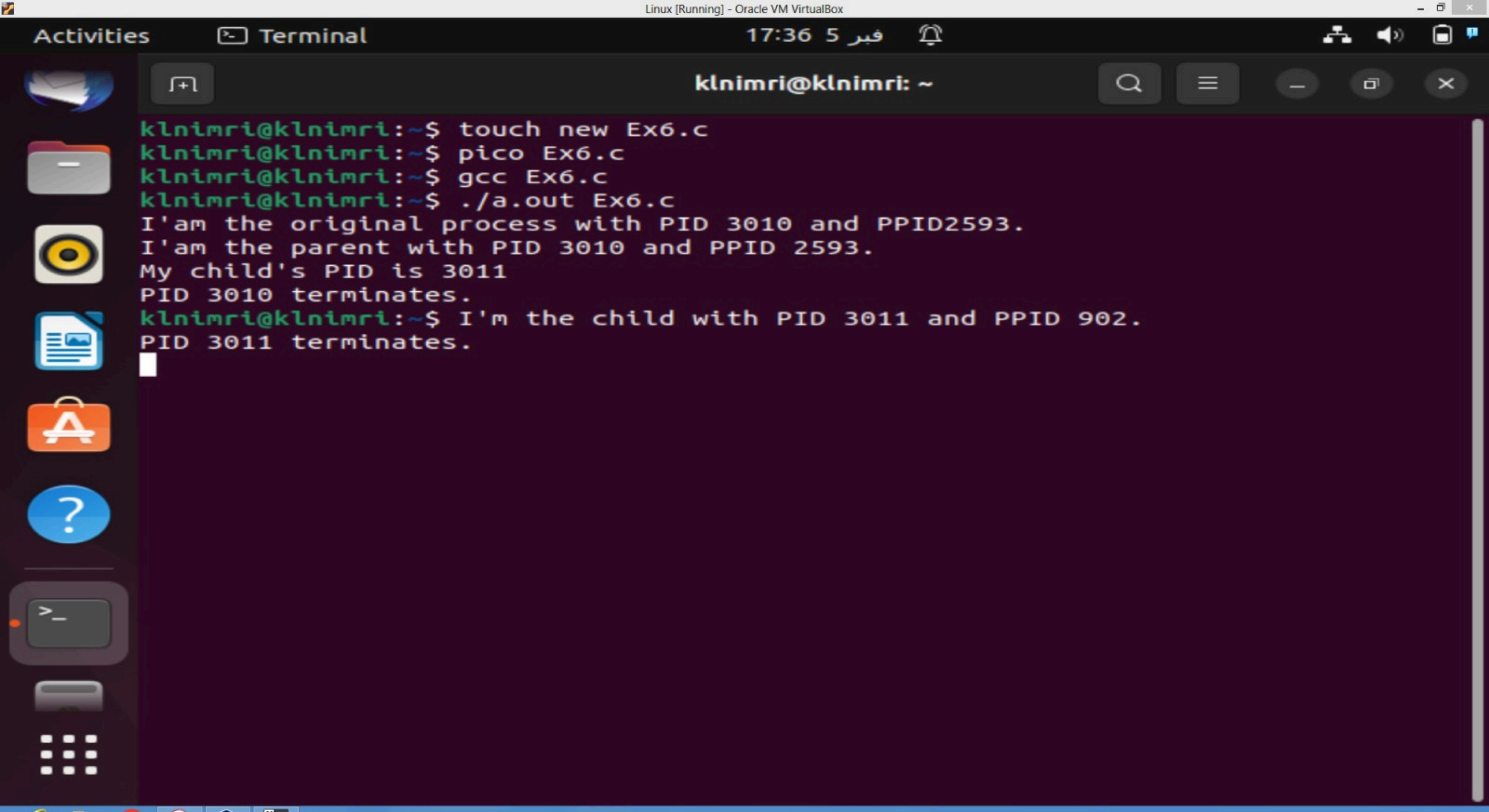
```
  GNU nano 6.2                              Ex5.c *
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
int main()
{
int forkresult;
int status;
printf("%d: I am the parent. Remember my number!\n",getpid());
printf("%d: I am now going to fork ... \n", getpid());
forkresult = fork();
if (forkresult != 0)
{ /* the parent will execute this code */
wait(&status);
printf("%d: My child's pid is %d\n", getpid(),
forkresult);
}
else /* forkresult == 0 */
{ /* the child will execute this code */
printf("%d: Hi! I am the child.\n", getpid());
}
printf("%d: like father like son. \n", getpid());

}

^G Help       ^O Write Out   ^W Where Is    ^K Cut         ^T Execute
^X Exit       ^R Read File   ^\ Replace     ^U Paste       ^J Justify
```



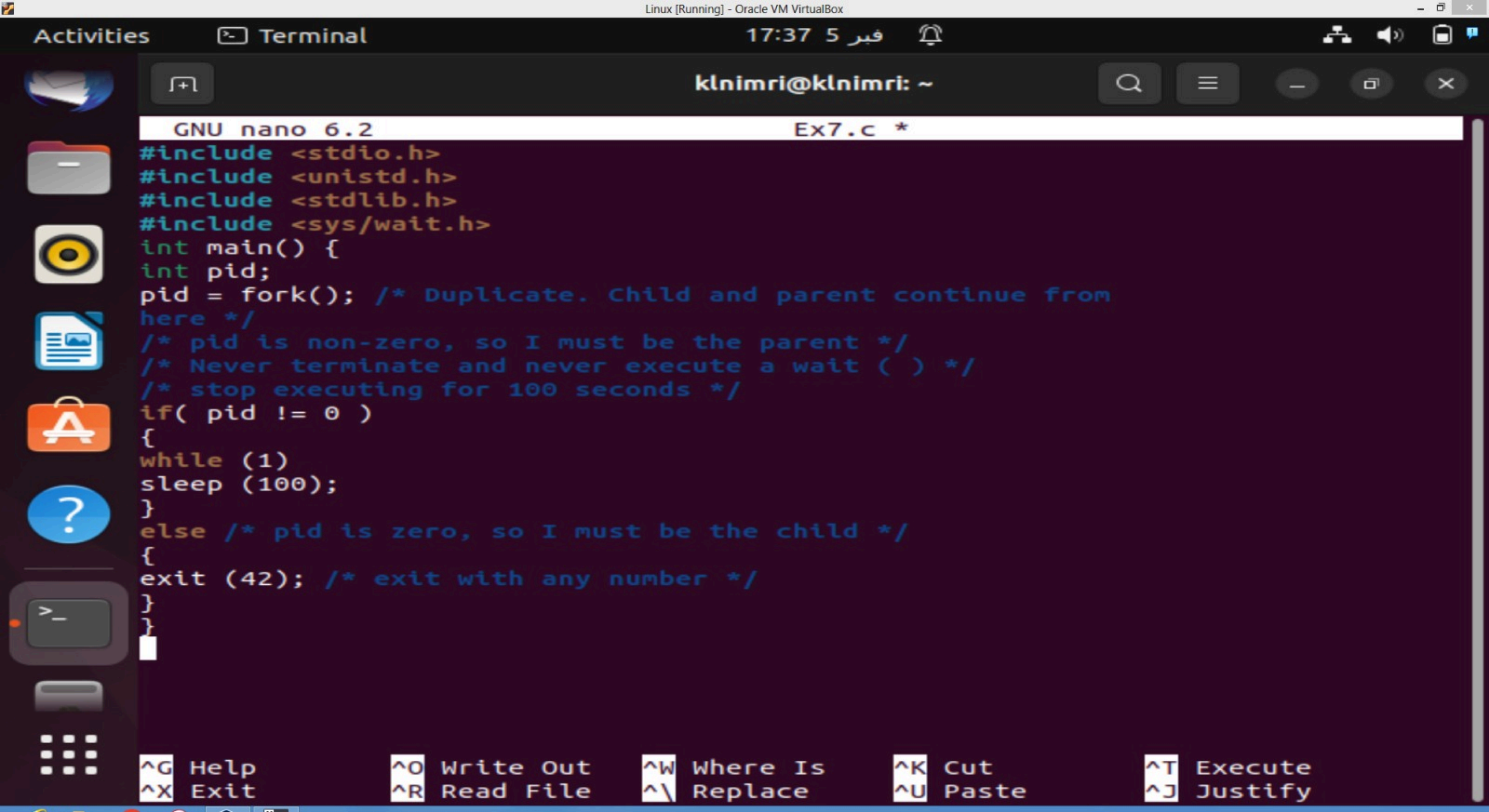
```
klnimri@klnimri:~$ pico Ex5.c
klnimri@klnimri:~$ gcc Ex5.c
klnimri@klnimri:~$ ./a.out Ex5.c
2974: I am the parent. Remember my number!
2974: I am now going to fork ...
2975: Hi! I am the child.
2975: like father like son.
2974: My child's pid is 2975
2974: like father like son.
klnimri@klnimri:~$ 
```



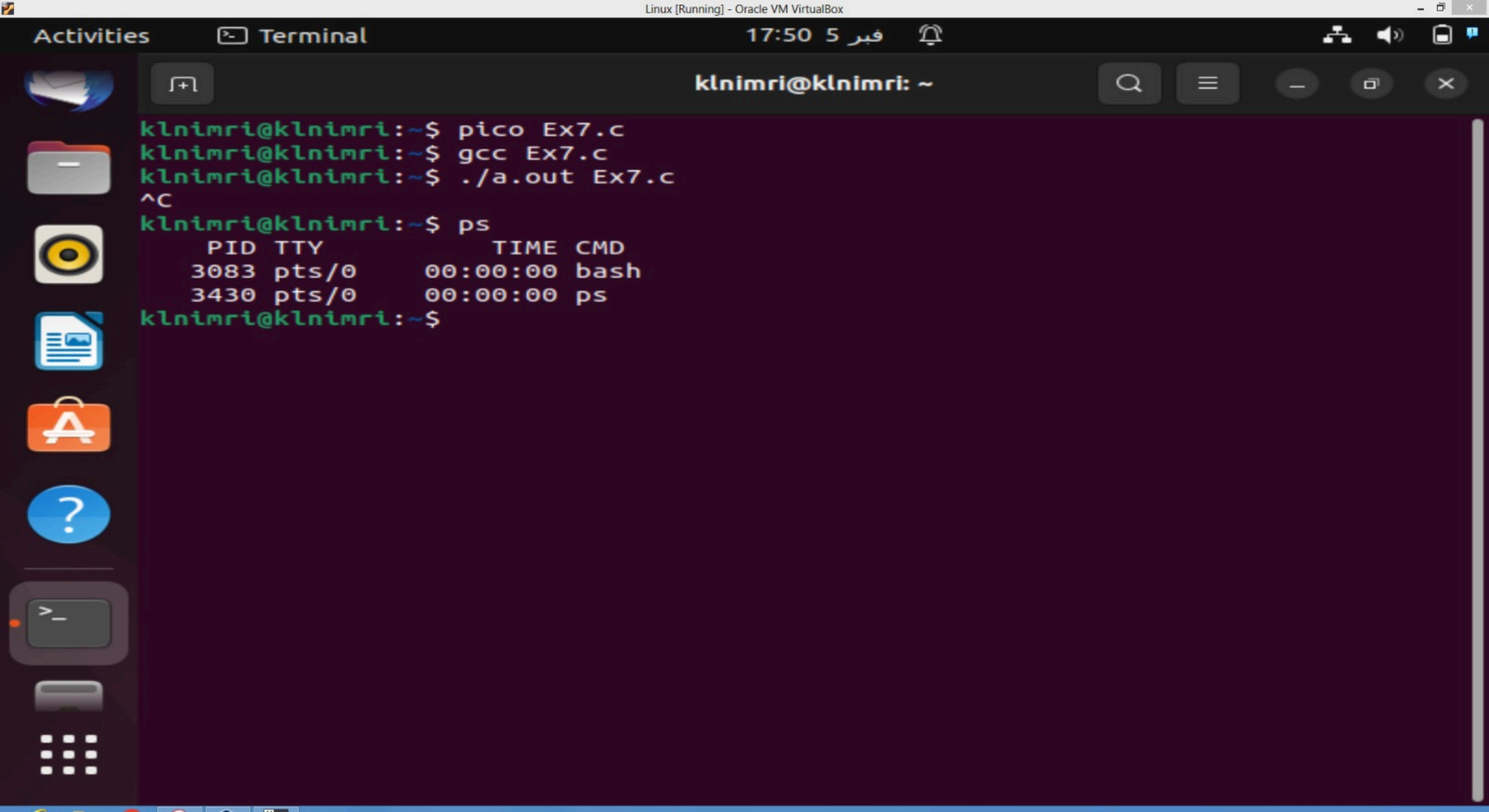
```
  GNU nano 6.2                         Ex6.c *
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
int main() {
int pid;
printf("I'am the original process with PID %d and PPID%d.\n",getpid(), getppid>
pid = fork() ; /* Duplicate. Child and parent continue
from here */
if( pid != 0 ) /* pid is non-zero,so I must be the parent*/
{
printf("I'am the parent with PID %d and PPID %d.\n",getpid(), getppid());
printf("My child's PID is %d\n", pid ) ;
}
else
/* pid is zero, so I must be the child */
{
sleep(4);
/* make sure that the parent terminates first */
printf("I'm the child with PID %d and PPID %d.\n",getpid(), getppid());
}
printf("PID %d terminates.\n", getpid());
}

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```



```
klnimri@klnimri:~$ touch new Ex6.c
klnimri@klnimri:~$ pico Ex6.c
klnimri@klnimri:~$ gcc Ex6.c
klnimri@klnimri:~$ ./a.out Ex6.c
I'am the original process with PID 3010 and PPID2593.
I'am the parent with PID 3010 and PPID 2593.
My child's PID is 3011
PID 3010 terminates.
klnimri@klnimri:~$ I'm the child with PID 3011 and PPID 902.
PID 3011 terminates.
```



```
GNU nano 6.2                              Ex7.c *
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
int main() {
int pid;
pid = fork(); /* Duplicate. Child and parent continue from
here */
/* pid is non-zero, so I must be the parent */
/* Never terminate and never execute a wait ( ) */
/* stop executing for 100 seconds */
if( pid != 0 )
{
while (1)
sleep (100);
}
else /* pid is zero, so I must be the child */
{
exit (42); /* exit with any number */
}
}
```



```
klnimri@klnimri:~$ pico Ex7.c
klnimri@klnimri:~$ gcc Ex7.c
klnimri@klnimri:~$ ./a.out Ex7.c
^C
klnimri@klnimri:~$ ps
    PID TTY          TIME CMD
   3083 pts/0    00:00:00 bash
   3430 pts/0    00:00:00 ps
klnimri@klnimri:~$
```