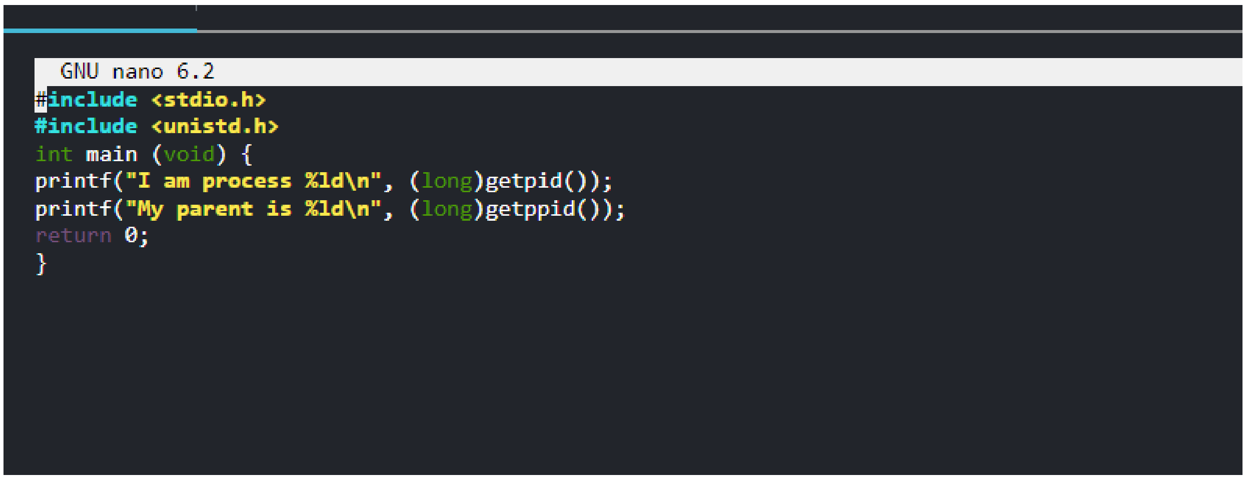
Sistemas Operacionais - 04N

Lab.02a

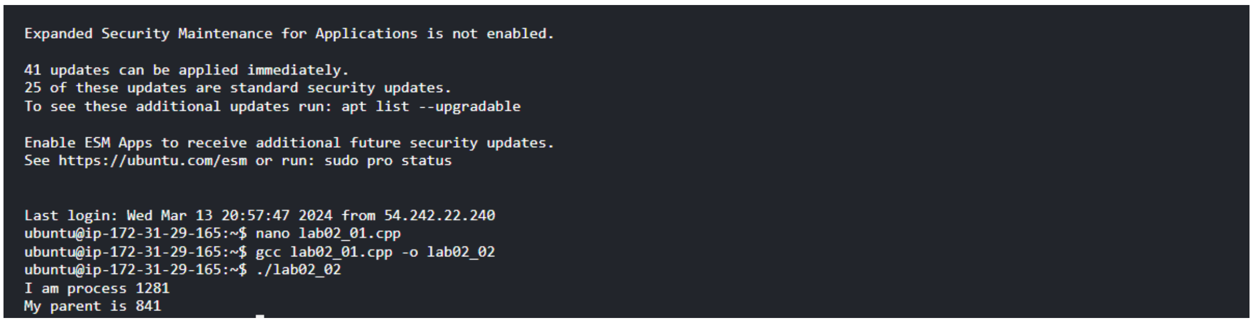
Guilherme Garcia Lima RA: 10409637

Isabela de Castro Jorge RA: 10409732

1)

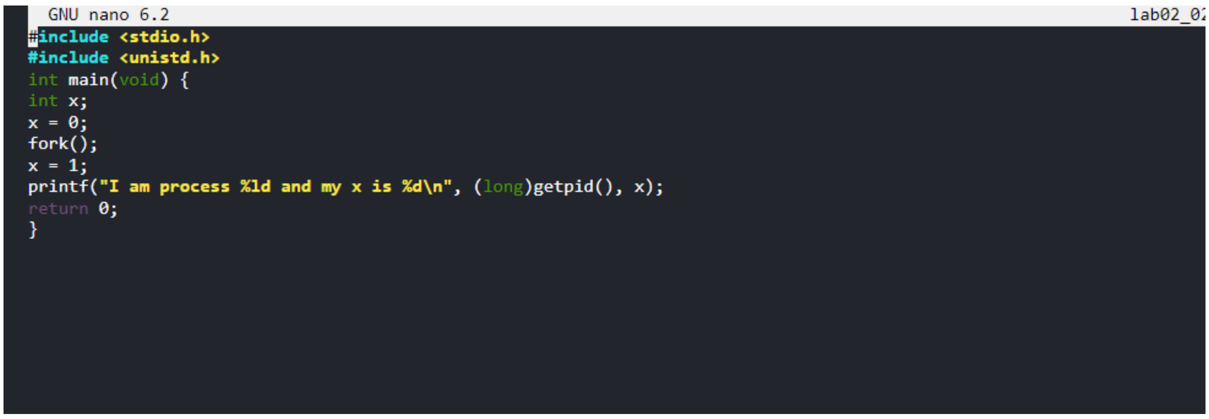


```
  GNU nano 6.2
#include <stdio.h>
#include <unistd.h>
int main (void) {
printf("I am process %ld\n", (long)getpid());
printf("My parent is %ld\n", (long)getppid());
return 0;
}
```

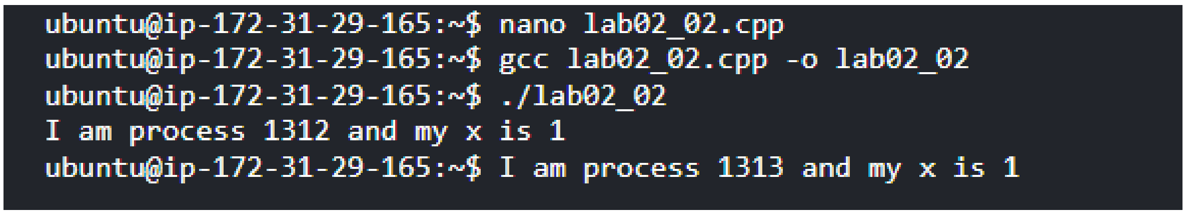


```
Expanded Security Maintenance for Applications is not enabled.

41 updates can be applied immediately.
25 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Wed Mar 13 20:57:47 2024 from 54.242.22.240
ubuntu@ip-172-31-29-165:~$ nano lab02_01.cpp
ubuntu@ip-172-31-29-165:~$ gcc lab02_01.cpp -o lab02_02
ubuntu@ip-172-31-29-165:~$ ./lab02_02
I am process 1281
My parent is 841
```

2)

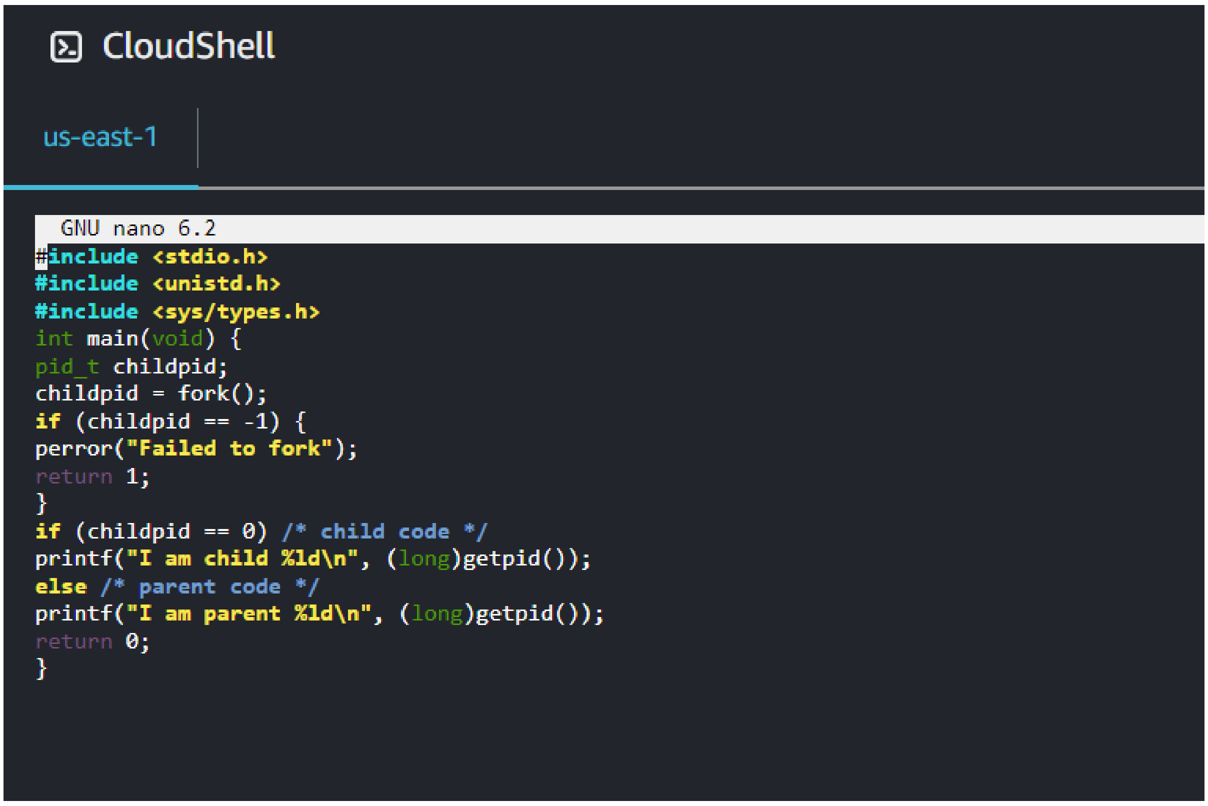


```
  GNU nano 6.2                                                          lab02_02
#include <stdio.h>
#include <unistd.h>
int main(void) {
int x;
x = 0;
fork();
x = 1;
printf("I am process %ld and my x is %d\n", (long)getpid(), x);
return 0;
}
```

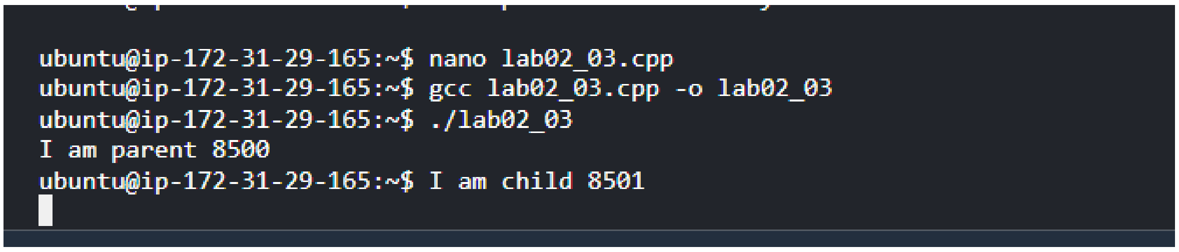


```
ubuntu@ip-172-31-29-165:~$ nano lab02_02.cpp
ubuntu@ip-172-31-29-165:~$ gcc lab02_02.cpp -o lab02_02
ubuntu@ip-172-31-29-165:~$ ./lab02_02
I am process 1312 and my x is 1
ubuntu@ip-172-31-29-165:~$ I am process 1313 and my x is 1
```

3)

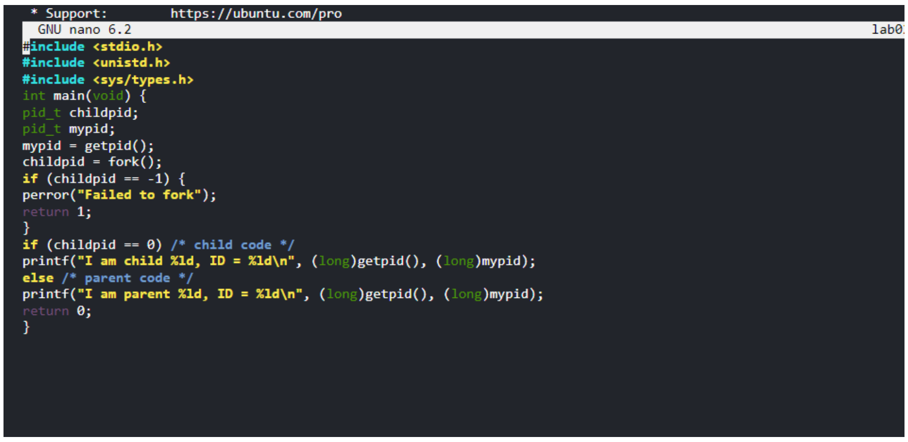


```
  GNU nano 6.2
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(void) {
pid_t childpid;
childpid = fork();
if (childpid == -1) {
perror("Failed to fork");
return 1;
}
if (childpid == 0) /* child code */
printf("I am child %ld\n", (long)getpid());
else /* parent code */
printf("I am parent %ld\n", (long)getpid());
return 0;
}
```

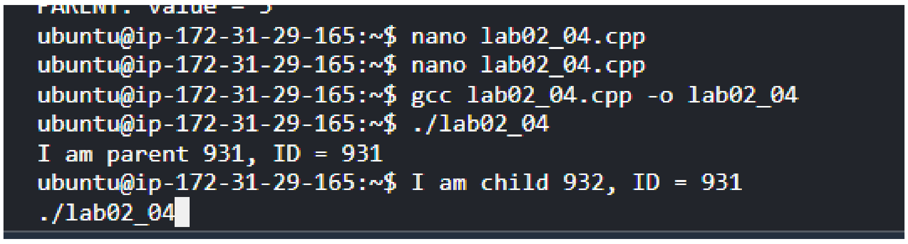


```
ubuntu@ip-172-31-29-165:~$ nano lab02_03.cpp
ubuntu@ip-172-31-29-165:~$ gcc lab02_03.cpp -o lab02_03
ubuntu@ip-172-31-29-165:~$ ./lab02_03
I am parent 8500
ubuntu@ip-172-31-29-165:~$ I am child 8501
```

4)



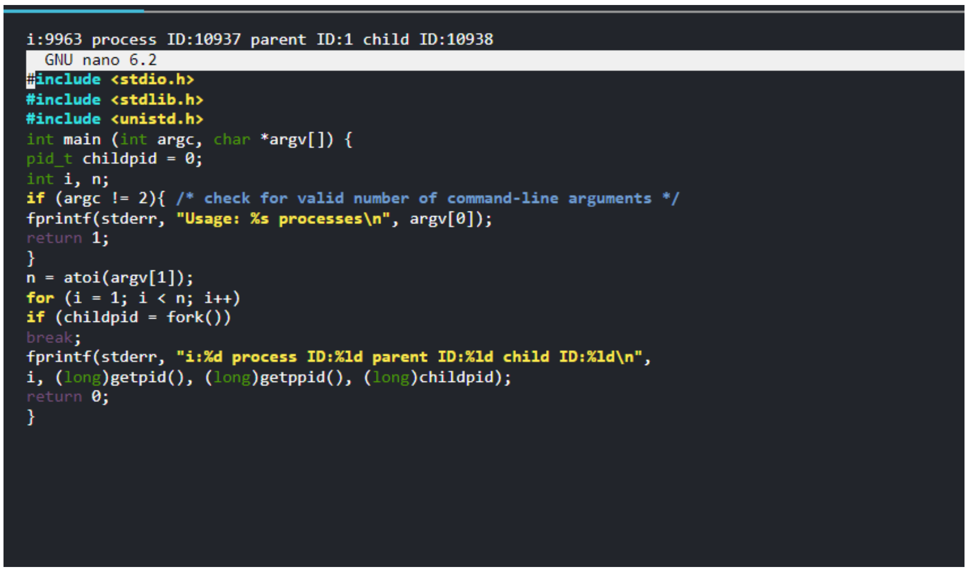
```
 * Support:        https://ubuntu.com/pro
  GNU nano 6.2                                                              lab0
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(void) {
pid_t childpid;
pid_t mypid;
mypid = getpid();
childpid = fork();
if (childpid == -1) {
perror("Failed to fork");
return 1;
}
if (childpid == 0) /* child code */
printf("I am child %ld, ID = %ld\n", (long)getpid(), (long)mypid);
else /* parent code */
printf("I am parent %ld, ID = %ld\n", (long)getpid(), (long)mypid);
return 0;
}
```



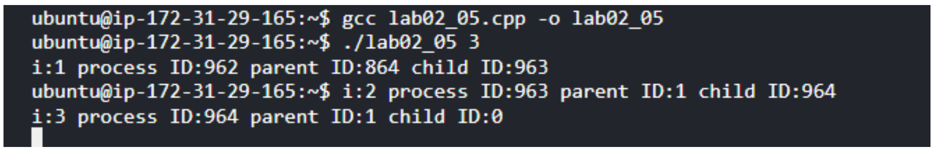
```
PARENT: value = 5
ubuntu@ip-172-31-29-165:~$ nano lab02_04.cpp
ubuntu@ip-172-31-29-165:~$ nano lab02_04.cpp
ubuntu@ip-172-31-29-165:~$ gcc lab02_04.cpp -o lab02_04
ubuntu@ip-172-31-29-165:~$ ./lab02_04
I am parent 931, ID = 931
ubuntu@ip-172-31-29-165:~$ I am child 932, ID = 931
./lab02_04
```

O código pega os IDs tanto do processo filho, quanto do processo pai.

5)



```
i:9963 process ID:10937 parent ID:1 child ID:10938
  GNU nano 6.2
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main (int argc, char *argv[]) {
pid_t childpid = 0;
int i, n;
if (argc != 2){ /* check for valid number of command-line arguments */
fprintf(stderr, "Usage: %s processes\n", argv[0]);
return 1;
}
n = atoi(argv[1]);
for (i = 1; i < n; i++)
if (childpid = fork())
break;
fprintf(stderr, "i:%d process ID:%ld parent ID:%ld child ID:%ld\n",
i, (long)getpid(), (long)getppid(), (long)childpid);
return 0;
}
```



```
ubuntu@ip-172-31-29-165:~$ gcc lab02_05.cpp -o lab02_05
ubuntu@ip-172-31-29-165:~$ ./lab02_05 3
i:1 process ID:962 parent ID:864 child ID:963
ubuntu@ip-172-31-29-165:~$ i:2 process ID:963 parent ID:1 child ID:964
i:3 process ID:964 parent ID:1 child ID:0
```