**Trabalho 1.1 - FORK**

Dupla: Rita de Cássia Lino Lopes - T03

Társila Samille Santos da Silveira - T01

**Passos para compilar:**

1. Abrir terminal
2. Digitar o seguinte comando:

gcc fork.c -o fork

**Passos para executar:**

1. Após compilar o programa
2. No terminal, digitar o seguinte comando:

./fork

**Dados do Sistema Operacional:**

Ubuntu 20.04 LTS

**Versão do GCC:**

9.3.0

Abaixo há uma cópia do código completo:

#include <sys/types.h>

#include <sys/wait.h>

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <time.h>

int main() {

time\_t begin, end;

char s\_begin[50], s\_end[50];

pid\_t filho1, filho2, neto1, neto2;

begin = time(NULL);

strftime(s\_begin,sizeof(s\_begin),"%d/%m/%Y %H:%M:%S",localtime(&begin));

printf("O pai nasceu em %s com PID = %d\n", s\_begin, getpid());

sleep(14);

filho1 = fork();

if (filho1 < 0) {

printf("Erro ao criar processo!!!\n");

return 1;

}

if (filho1 == 0) {

begin = time(NULL);

strftime(s\_begin,sizeof(s\_begin),"%d/%m/%Y %H:%M:%S",localtime(&begin));

printf("O filho 1 nasceu em %s com PID = %d\n", s\_begin, getpid());

sleep(12);

neto1 = fork();

if (neto1 < 0) {

printf("Erro ao criar processo!!!\n");

}

if (neto1 == 0) {

begin = time(NULL);

strftime(s\_begin,sizeof(s\_begin),"%d/%m/%Y %H:%M:%S",localtime(&begin));

printf("O neto 1 nasceu em %s com PID = %d\n", s\_begin, getpid());

sleep(12);

end = time(NULL);

strftime(s\_end,sizeof(s\_end),"%d/%m/%Y %H:%M:%S",localtime(&end));

printf("O neto 1(%d) morreu com %d anos, tempo de vida: %s - %s \n", getpid() , (int) difftime(end, begin), s\_begin, s\_end);

exit(0);

}

sleep(18);

end = time(NULL);

strftime(s\_end,sizeof(s\_end),"%d/%m/%Y %H:%M:%S",localtime(&end));

printf("O filho 1(%d) morreu com %d anos, tempo de vida: %s - %s \n", getpid() , (int) difftime(end, begin), s\_begin, s\_end);

exit(0);

}

sleep(2);

filho2 = fork();

if (filho2 < 0) {

printf("Erro ao criar processo!!!\n");

return 1;

}

if (filho2 == 0) {

begin = time(NULL);

strftime(s\_begin,sizeof(s\_begin),"%d/%m/%Y %H:%M:%S",localtime(&begin));

printf("O filho 2 nasceu em %s com PID = %d\n", s\_begin, getpid());

sleep(14);

neto2 = fork();

if (neto2 < 0) {

printf("Erro ao criar processo!!!\n");

}

if (neto2 == 0) {

begin = time(NULL);

strftime(s\_begin,sizeof(s\_begin),"%d/%m/%Y %H:%M:%S",localtime(&begin));

printf("O neto 2 nasceu em %s com PID = %d\n", s\_begin, getpid());

sleep(18);

end = time(NULL);

strftime(s\_end,sizeof(s\_end),"%d/%m/%Y %H:%M:%S",localtime(&end));

printf("O neto 2(%d) morreu com %d anos, tempo de vida: %s - %s \n", getpid() , (int) difftime(end, begin), s\_begin, s\_end);

exit(0);

}

sleep(16);

end = time(NULL);

strftime(s\_end,sizeof(s\_end),"%d/%m/%Y %H:%M:%S",localtime(&end));

printf("O filho 2(%d) morreu com %d anos, tempo de vida: %s - %s \n", getpid() , (int) difftime(end, begin), s\_begin, s\_end);

exit(0);

}

sleep(44);

end = time(NULL);

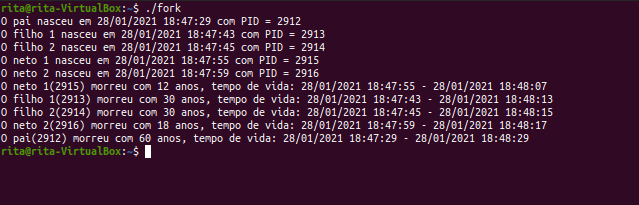
strftime(s\_end,sizeof(s\_end),"%d/%m/%Y %H:%M:%S",localtime(&end));

printf("O pai(%d) morreu com %d anos, tempo de vida: %s - %s \n", getpid() , (int) difftime(end, begin), s\_begin, s\_end);

return 0;

}

A figura 1 mostra o resultado da execução do programa:



**Figura 1: resultado da execução do programa**