

Lab 2B - Sistemas Operacionais

Nome: Rodrigo Mello de Campos | RA: 10416316

Nome: Vitor Leandro Machado | RA: 10409358

Nome: Rodrigo Lucas Rosales | RA: 10365071

Nome: Gabriel Forster | RA: 10409702

Considere o exemplo fornecido para troca de mensagens entre pai e filho. Altere o programa de modo que a alteração da variável value se reflita no processo pai, ou seja, o último print do programa deve apresentar o seguinte resultado:

Código original:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
```

```
int value = 5;
```

```
int main()
```

```
{
```

```
    pid_t pid;
```

```
    pid = fork();
```

```
    if (pid == 0) { /* child process */
```

```
        printf("Entrei no filho!\n");
```

```
        value += 15;
```

```
        printf ("CHILD: value = %d\n",value); /* LINE A */
```

```
        return 0;
```

```
    }
```

```
    else if (pid > 0) { /* parent process */
```

```
        wait(NULL);
```

```
        printf ("PARENT: value = %d\n",value); /* LINE A */
```

```
        return 0;
```

```
    }
```

```
}
```

Código alterado:

```

[1/4]
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>

int value = 20;

int main()
{
    pid_t pid;

    pid = fork();

    if (pid == 0) { /* child process */
        printf("Entrei no filho!\n");
        value += 15;
        printf ("CHILD: value = %d\n",value); /* LINE A */
        return 0;
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf ("PARENT: value = %d\n",value); /* LINE A */
        return 0;
    }
}

```

Execução do código

```

[ec2-user@ip-172-31-25-65 ~]$ gcc -o lab2b compilar.c
[ec2-user@ip-172-31-25-65 ~]$ ./lab2b
Entrei no filho!
CHILD: value = 20
PARENT: value = 5
[ec2-user@ip-172-31-25-65 ~]$ sudo nano compilar.c
[ec2-user@ip-172-31-25-65 ~]$ gcc -o lab2b compilar.c
[ec2-user@ip-172-31-25-65 ~]$ ./lab2b
Entrei no filho!
CHILD: value = 35
PARENT: value = 20
[ec2-user@ip-172-31-25-65 ~]$ 

```