COM120 – SISTEMAS OPERACIONAIS
EXERCÍCIO PRÁTICO 01 – EP01

Data da aula: 15/09/2021

Entregar até 21/09/2021 – 23:59 no Moodle

Analise o programa abaixo:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
	int pid=0;
	int pidpai, pidfilho;
	pidpai = getpid();
	pid=fork();
	if (pid != 0)
	{
		printf( "Sou processo pai!!! Meu PID é %d\n", pidpai );
	}
	else
	{
		pidfilho = getpid();
		printf("Sou processo filho!!! Meu PID é %d e o PID do meu pai é %d\n", pidfilho,
			pidpai);
	}
	return 0;
}
```

Este programa cria um processo filho que imprime a mensagem “Sou processo filho!!!”.
Para compilar o programa use:

```
gcc -Wall codigoFonte.c -o programaExecutavel
```

Para executar o programa use:

```
./programaExecutavel
```

Exercício 1:
Crie um programa em C que cria um processo filho através do fork(). O processo original deve exibir na tela a frase “Sou o processo pai” 30 vezes, enquanto o processo filho deve exibir a frase “Sou o processo filho” 50 vezes na tela.
Execute o programa algumas vezes e comente no final próprio arquivo fonte o que aconteceu.

Exercício 2:
Modifique o exercício 1 para que o processo pai espere o processo filho terminar de executar antes de começar a exibir a frase “Sou o processo pai” na tela. Para isso, você pode utilizar a função waidpid definida no cabeçalho **#include <unistd.h>**. Exemplo de uso da função waitpid:

int childStatus;
waitpid(pid, &childStatus, 0);

Onde ***pid*** deve ser o PID do processo que desejamos esperar. Após a execução desta função o inteiro ***childStatus*** irá armazenar o estado do processo que estávamos esperando. Obs.:
Antes do childStatus existe um “e comercial”.
Observação: inclua **#include <sys/wait.h>** para evitar warnings de declaração da função waitpid.
Execute o programa algumas vezes e comente no final próprio arquivo fonte o que aconteceu.

Exercício 3:
Considere o programa abaixo. Quandos processos serão criados (considerando o processo inicial)?

```
int main(int argc, char **argv)
{
	int pid = fork();
	if(pid == 0)
	{
		fork();
		fork();
	}
	else
	{
		fork();
	}
}
```

Execute o programa algumas vezes e comente no final próprio arquivo fonte o que aconteceu.

Observação: Todo arquivo de programa fonte deve ter identificação no cabeçalho do programa (nome, número de matrícula, disciplina, exercício, data, etc), comentários a respeito de métodos de cálculo e outras coisas mais, isso é o mínimo para podermos entender o seu programa.
Salve os arquivos fonte em PDF para entregar, o Moodle só irá aceitar .pdf.