



TP02 - Sistema Operacional

Aluno: Nathann Zini dos Reis

Matrícula: 19.2.4007

Q.1 - Threads vs Fork

Threads e Forks basicamente funcionam de modo a executar em paralelo um código com o comportamento específico.

Um fork é considerado "filho" do código em que foi chamado e basicamente é um processo cujo código é idêntico ao do "pai", porém é executado alguma função específica para cada filho que é criado. É importante ressaltar que é feita uma cópia de tudo para o filho: do código, da memória... enfim é um novo processo. Também, as informações entre o pai e os filhos não são compartilhadas. Cada processo tem seus próprios dados. A menos que é usada alguma primitiva de comunicação entre processos (IPC).

Já uma thread é uma divisão do processo para executar pequenos processos paralelamente ao código principal. Nesse caso é compartilhado a memória do processo com suas threads (um processo pode ter várias threads) e todos os dados.

Para executar o código no C-Compiler online disponibilizado pelo professor no enunciado do trabalho basta copiar o código pronto dentro da interface do site ou digitar um código próprio e pressionar "run" para que o mesmo seja executado.

Q.2 - O que foi feito

Foram feitos 2 códigos distintos porém bem similar entre si e com a mesma ideia de funcionamento. Ambos os códigos tem a finalidade de controlar um saldo de Unidades de Dinheiro (UD), em que é possível acrescentar 100UD, retirar 100UD e imprimir o saldo atual.

No primeiro código foi utilizado a ideia de Fork, com a IPC pipe para fazer a intercomunicação entre os processos permitindo o compartilhamento de informações. E foi separado em 3 processos pesados, ou forks, distintos: Um responsável pela impressão, outro pela decrementação e, por fim, o último responsável pela incrementação.

No segundo código, por sua vez, foi utilizado a ideia de threads para realizar as mesmas funções que o código anterior. Foram criado 3 subprocessos, ou threads, responsável pelas mesmas finalidades do código anterior.