**Manual De Intrução Do Usuário:**

1. Pré-requisitos:

* Possuir um compilador C++ (*como g++*) instalado no seu sistema operacional.
* Ter o arquivo de programa de entrada (*.txt*).

1. Como compilar:

Obs.: Esse programa foi feito no Linux, porém funciona perfeitamente também no Windows.

* gcc t2.c -o t2
* *t2.c* é o nome do arquivo c
* *t2* é o nome do arquivo exe criado
* ./ t2.exe

1. Como funciona o programa após a execução:

* O programa irá ler todos os arquivos da pasta de arquivos *.txt* e mostrará cada arquivo que foi lido.
* Irá mostrar uma tabela com ID de cada programa encontrado, seu tempo de chegada incial(padrão), custo do ci(padrão) e o pi inicial(padrão).
* Pergunta ao usuário se deseja carregar todos os programas encontrados ou selecionar individualmente.
* Conforme cada programa carregado, pergunta se o usuário deseja alterar seu tempo de chegada (Arrival Time), custo computacional(ci), e período(pi). Caso deseje alterar irá perguntando o valor que quer colocar no Arrival Time,no ci,e no pi. Após alterar os parâmetros e desejar não fazer mais alterações irá mostrar a tabela com os novos parâmetros.
* Depois de mostrar a tabela com alteração ou sem alteração irá inicializar o escalonamento EDF, mostrando a unidade de tempo atual e as informações a seguir a cada passo:
* O programa em execução caso haja um.
* O estado de cada programa(ready,running,blocked,done).
* A fila de programas ready(ordenando por deadline).
* A fila de programas bloqueados(aguardando).
* A fila de programas finalizados.
* Saídas de chamadas de sistema (impressão do acumulador, solicitação de entrada).
* Mensagens sobre perda de deadline e recalculação do deadline.
* Ao terminar a execução de todos os programas finaliza a simulação.
* Conceitos do Simulador:
* **ID do Programa:** Um identificador único para cada programa carregado.
* **Arrival Time (Tempo de Chegada):** O momento em que o programa se torna disponível para execução.
* **Ci (Custo Computacional):** O número total de unidades de tempo de CPU necessárias para executar todas as instruções do programa em cada período.
* **Pi (Período):** O intervalo de tempo entre as ativações (ou repetições) de um programa periódico.
* **Deadline Absoluto (Di Absoluto):** O momento em que a execução do programa deve ser concluída (calculado como Arrival Time + Pi).
* **Estado do Programa:** Indica a situação atual do programa:
* **ready:** Pronto para ser executado.
* **running:** Atualmente em execução na CPU.
* **blocked:** Aguardando a conclusão de uma operação (simulada por um tempo aleatório após um syscall).
* **done:** Concluiu sua execução.
* **countCI:** Um contador que acompanha quantas instruções do programa já foram executadas dentro do período atual.
* **Accumulator (acc):** Um registrador virtual utilizado para operações aritméticas e de transferência de dados.
* **Program Counter (pc):** Um contador que indica a próxima instrução a ser executada dentro de um programa.
* **EDF (Earliest Deadline First):** O algoritmo de escalonamento utilizado. A tarefa com o deadline absoluto mais próximo tem a maior prioridade.
* A cada instrução o simulador assume que leva uma unidade de tempo a ser executada (exceto *syscall* que pode levar mais tempo).
* O tempo de bloqueio após um *syscall* é gerado aleatoriamente entre 1 e 3 unidades de tempo.
* Quando o deadline absoluto de um programa pronto ou em execução é menor ou igual ao tempo atual ocorre a perda de deadline. Nesse caso, o programa retorna à fila de prontos com um novo deadline.