|  |  |
| --- | --- |
|  | **FIPP**  CIÊNCIA DA COMPUTAÇÃO |

**Gabriel Hideki**

**Henrique dos Santos**

**Thomas Henschel**

**Yuri Gabriel**

**Ciclo de Vida do Processo (Sistemas Operacionais)**

Presidente Prudente - SP

2025

O projeto tem a ideia de simular o ciclo de vida dos processos em um Sistema Operacional, utilizando dois tipos de fila: a circular ou Round-Robin (Fila de Prontos) e a fila do tipo FIFO (Filas de Espera e de Execução).

Na fila de execução, o processo fica com um limite de uso da CPU (quantum) de 10 unidades de tempo. Assim que atingir esse limite, se o processo não tiver finalizado, ele volta para a Fila de Prontos.

Os códigos foram divididos em 3:

* **“Simulacao.cpp”**: Responsável pela criação de processos, por executá-los e por mover os processos entre as filas desejáveis.
* **“TADProcesso.h”**: Tem a finalidade de guardar a estrutura do PCB e as funções para utilizar as filas (inserir e remover, por exemplo).
* **“TADFrontend.h”**: Tem a finalidade de deixar a aparência do programa mais legível e permitir ver o que está acontecendo na simulação mostrando qual processo está sendo executado e quais estão na fila para entrar.

Fluxo da Simulação:

1. Ao se executar a simulação dos processos, o quantum do escalonamento já é definido previamente como 10. São criados 4 filas PCB (Espera, Pronto, Mouse, HD e Concluído), a fila Pronto recebe processos em sua fila em ordem de prioridade (Round Robin).

Cada Processo ao ser criado recebe um tempo aleatório de 10 a 20 unidades de tempo para terminar a execução.

1. Ao se iniciar a execução da CPU, é retirado da fila Pronto de acordo com a ordem de execução, randomiza o tempo total de execução e limita a execução há cada 10 unidade de tempo.
2. Durante a execução da CPU é verificado se o tempo do quantum chegou a 10, caso sim o processo em execução é interrompido e colocado na fila de pronto, para se colocar outro processo (que tenha uma prioridade de nível menor do que o processo que estava em execução) de acordo com a política de escalonamento Round Robin.

Também feito uma randomização se um processo pode sofrer um bloqueio (randomizado para 12% de chance de ocorrer a cada unidade de tempo). Caso ele seja bloqueado ele recebe de 5 até 15 unidades de tempo em bloqueio, que após as unidades de tempo estabelecidas, o processo muda para o estado pronto e é jogado para a fila de Prontos.

1. É verificado a cada ciclo se o processo atual ja terminou seu processo, caso sim, ele é jogado para uma fila de concluídos para ser contabilizado e exibido posteriormente.

Caso ele possua filhos, o processo pai é colocado na fila de espera até que seus filhos sejam executados e terminem.

1. Durante a execução, há uma probabilidade de 10% de o processo atual gerar um novo processo filho.

O processo pai é temporariamente movido para a Fila de Espera, enquanto o novo processo filho é inserido na Fila de Prontos para execução.

O relacionamento entre pai e filho é mantido através dos campos pid e ppid da estrutura PCB.

1. A cada unidade de tempo (ut), o simulador atualiza todas as filas na tela.

Mostra qual processo está em execução, quantos estão prontos, bloqueados, em espera ou concluídos. Além dos processos que estão em dispositivo externo (Mouse e teclado).

1. Ao se apertar a tecla ESC, todos os processos que ainda estiverem nas filas são movidos para a Fila de Concluídos, a simulação termina e aparece os dados de cada processo e dos filhos que foram criados por cada um:

Quantidade de processos finalizados.

Quantidade de processos bloqueados.

Tempo médio de espera (ou bloqueio).

Relação de cada processo pai e seus filhos.

Tempo total de execução de cada processo.

Estruturas de Dados:

typedef struct pcb {

int pid, ppid;

int tempo\_total;

int tempo\_executado;

int tempob\_inicial;

int total\_espera;

int tempo\_bloqueio;

int tempo\_inic;

int prioridade;

int tempo\_bloqueado\_total;

int ja\_bloqueou;

char status;

char bloc;

struct pcb \*prox;

} PCB;

\*Estrutura Responsavel pelo PCB (Controle dos Processos) onde guarda a prioridade, e qual a situação do processo (pronto, espera e bloqueado)

typedef struct filaPCB {

struct pcb \*inicio, \*fim;

char fila;

} filaPCB;

\*Estrutura da fila circular com prioridade.

Antes de usar:

Ao executar o programa tem que deixar em tela cheia, pois se abrir normal pode dar erro na interface e não mostrar os processos adequadamente.

Interface:

Menu Inicial  
Texto

O conteúdo gerado por IA pode estar incorreto.

Indicando os nomes dos alunos que fizeram e o que deve ser feito para começar

Simulação:  
Texto

O conteúdo gerado por IA pode estar incorreto.

O programa exibe quais processos estão em cada fila, informa se eles criaram processos filhos e mostra o tempo total da simulação. Além disso, ele indica o que deve ser feito para finalizar o programa ou para criar um novo processo.