

Universidade Federal do Rio de Janeiro
Período de 2023.3

Professor(a): **Rafaela Correia Brum**

Disciplina: Arquitetura de Computadores e Sistemas Operacionais

Integrantes: Luiz Brasil, Milton Salgado e Vitor Vicente

RELATÓRIO - TP2

Implementação de um Escalonador utilizando o Algoritmo Round-Robin com Feedback

Observações

1. Introdução

O presente relatório descreve o trabalho prático 2 realizado pelo nosso grupo da disciplina, cujo objetivo é a implementação de um programa em linguagem C que simula a execução de um escalonador de processos, assemelhando-se ao funcionamento de um sistema operacional real.

Um escalonador de processos desempenha um papel crucial em sistemas operacionais, gerenciando a ordem de execução dos processos que competem pelos recursos do sistema, especialmente o tempo de CPU. Utilizando algoritmos específicos, ele determina a sequência em que os processos serão executados, baseando-se em critérios como prioridade, tempo de execução e outras características. O programa desenvolvido simula a execução dos processos, exibindo na tela o processo que ocupa a CPU a cada unidade de tempo (u.t.), uma medida de tempo arbitrária conhecida pelo sistema computacional.

Conforme novos processos requisitam o uso da CPU, eles são organizados de maneira a proporcionar uma competição justa. Isso evita uma competição injusta entre os processos,

impedindo que um processo espere indefinidamente por um recurso (starvation). Tal situação poderia resultar em um sistema operacional não responsivo e inconsistente.

O relatório abordará detalhadamente a estrutura do simulador de escalonamento de processos, destacando a implementação do algoritmo Round Robin com Feedback. Cada parte do código será apresentada e explicada, fornecendo uma compreensão clara das decisões de projeto e das funcionalidades implementadas. O objetivo é proporcionar uma visão educativa e esclarecedora sobre o funcionamento interno do simulador, adequado para alunos iniciantes no curso de Computação.

2. Funcionamento do Escalonador

O simulador de escalonamento de processos implementa a estratégia de escalonamento conhecida como Round Robin com feedback. Esta abordagem combina a simplicidade do algoritmo Round Robin com a flexibilidade de ajustar dinamicamente as prioridades dos processos, permitindo que processos que demandam mais tempo de CPU recebam fatias maiores.

2.1. Round Robin: A Base do Escalonamento

O algoritmo Round Robin é a base do escalonador, onde cada processo é atribuído uma fatia de tempo de CPU fixa antes de passar para o próximo na fila. Ao final da execução, o processo corrente é movido para o final da fila de espera. Isso cria uma distribuição equitativa do tempo de CPU entre os processos, evitando que um processo monopolize os recursos.

2.2. Round Robin com Feedback: Adaptação Dinâmica

No entanto, o Round Robin com feedback introduz uma abordagem mais dinâmica. Cada processo é inicialmente atribuído a uma fila com uma prioridade padrão, neste caso, uma fila de alta prioridade. Os processos na fila são programados usando o algoritmo Round Robin, recebendo fatias de tempo fixas de CPU em uma ordem circular. Se um processo não for concluído durante sua fatia de tempo, ele é movido para uma fila de baixa prioridade.

2.3. Adaptação de Prioridades

A capacidade de ajustar dinamicamente as prioridades é crucial no Round Robin com feedback. Processos que frequentemente exigem mais tempo de CPU são promovidos para filas de alta prioridade, garantindo que recebam fatias de tempo mais substanciais. Esse monitoramento dinâmico do comportamento dos processos permite uma adaptação contínua ao ambiente de execução.

2.4. Interrupções por Operações de E/S

Os processos podem ser interrompidos ao realizar operações de E/S, uma vez que ficam bloqueados até receberem uma resposta do processo responsável por essa operação. No simulador, dependendo do tipo de E/S realizada, o processo pode retornar a qualquer uma das duas filas iniciais existentes. O simulador suporta três tipos de E/S: Disco, fita magnética e impressora.

Este mecanismo de interrupção por operações de E/S adiciona uma camada de realismo à simulação, refletindo o comportamento dinâmico de sistemas operacionais reais durante a execução de processos.

Os próximos segmentos do relatório detalhará as funções responsáveis pelo funcionamento do escalonador, abordando as estruturas de dados utilizadas, e em seguida, as decisões de projeto e fornecendo exemplos práticos da execução do simulador.

3. Definições do Escalonador

A definição de algumas macros fundamentais molda o cenário de execução do simulador, fornecendo parâmetros cruciais para a dinâmica e comportamento dos processos simulados.

- **MAXIMO_PROCESSOS:** Essa macro estabelece o número total de processos a serem criados e executados no simulador. Em nossa simulação, esse valor foi configurado para 5, limitando o conjunto de processos ativos durante a execução. O PID (identificador do processo) varia incrementalmente de 0 a MAXIMO_PROCESSOS - 1, atribuindo identificadores únicos a cada processo em execução.
- **QUANTUM:** A macro QUANTUM determina a quantidade de unidades de tempo que compõem um quantum. Nossa simulação adotou um valor de 2 para evitar outputs extensos e impraticáveis para inclusão detalhada no relatório. O ajuste desse valor influencia diretamente a granularidade do escalonamento, impactando a frequência com que a CPU muda de processo.
- **TEMPO_MAXIMO_INICIO:** Esta macro define o intervalo de tempo durante o qual novos processos podem chegar ao sistema. Em nossa configuração, esse intervalo é definido como 4 unidades de tempo, controlando a dinâmica de entrada de novos processos e sua distribuição ao longo da simulação.
- **TEMPO_MINIMO_CPU e TEMPO_MAXIMO_CPU:** Essas macros estabelecem os limites inferior e superior para o tempo de serviço de um processo. Em nossa simulação, esses

valores foram configurados como 5 e 10, respectivamente, influenciando a variabilidade na duração da execução de cada processo.

- **Filas de Prioridade:** Novos processos entram na fila de alta prioridade, enquanto processos que sofreram preempção são redirecionados para a fila de baixa prioridade. Processos que retornam de operações de E/S voltam à fila ditada pela regra específica do dispositivo em questão.
- **Operações de E/S:** As operações de E/S são essenciais para simular o comportamento realista do sistema operacional. Três tipos de dispositivos foram implementados na simulação:
 - **Disco:** A duração dessa operação é definida pela macro TEMPO_DISCO (configurada para 2 u.t. na simulação). Ao encerrar, o processo que a chamou entra na fila de baixa prioridade.
 - **Fita Magnética:** A duração é definida por TEMPO_FITA (configurada para 3 u.t.). Ao encerrar, o processo retorna à fila de alta prioridade.
 - **Impressora:** O tempo de execução é definido por TEMPO_IMPRESSORA (configurado para 4 u.t.). Ao encerrar, o processo retorna à fila de alta prioridade.
- **Execução de Operações de E/S:** Ao início da simulação, um processo pode necessitar de 0 a 2 operações de E/S, com a quantidade e tempo de início sendo aleatórios e definidos dentro do intervalo do tempo de execução do programa. O tempo de chegada do processo de E/S é calculado em relação ao tempo decorrido desde o início da execução do processo, contabilizando apenas o tempo em que o processo esteve na CPU.

4. Header "auxiliar.h"

Arquivo de cabeçalho que contém as funções auxiliares implementadas.

4.1. Função quicksort

A função realiza a ordenação rápida de um vetor de operações de E/S, utilizando o algoritmo de particionamento.

4.2. Função particiona

Responsável por particionar o vetor de operações de E/S durante o processo de ordenação.

4.3. Função troca

Realiza a troca de posição entre dois elementos do vetor de operações de E/S.

4.4. Função controla_erro_alocacao

Trata erros de alocação de memória, garantindo um fluxo controlado em caso de falhas.

5. Header "escalonador.h"

Arquivo de cabeçalho que contém as funções necessárias para o funcionamento do escalonador.

5.1. Função escalonador

Simula o escalonamento de processos, utilizando as filas de alta e baixa prioridade, de disco, de fita e de impressora.

5.2. Função imprime_informacao_processos

Imprime informações dos processos, auxiliando na visualização do estado do sistema.

5.3. Função verifica_novos_processos

Verifica a chegada de novos processos no sistema em determinado momento.

5.4. Função imprime_tempos_turnaround

Imprime os tempos de turnaround dos processos, fornecendo métricas importantes de desempenho.

5.5. Funções verifica_processador e verifica_es

Verificam a existência de processos nas filas de prioridade e nas filas de dispositivos de entrada/saída, respectivamente.

5.6. Função envia_para_es

Envia um processo para uma das filas de E/S (disco, fita ou impressora).

5.7. Função atualiza_tempo_turnaround

Atualiza o tempo de turnaround de um processo, refletindo seu desempenho durante a execução.

6. Header "fila.h"

Arquivo de cabeçalho contendo as estruturas de dados e funções de manipulação de filas.

6.1. Estruturas de Dados

Define as estruturas de dados necessárias para a implementação das filas, incluindo a estrutura de nó (No) e a estrutura de fila (Fila).

6.1.1. Estrutura No (_No):

A estrutura _No é utilizada para representar um nó em uma lista encadeada. Cada nó contém dois principais campos:

- Processo (processo):
 - Tipo: Processo
 - Descrição: Armazena o processo associado ao nó. Esta estrutura Processo provavelmente contém informações relacionadas a um processo em um sistema operacional, como o identificador do processo (pid), tempos de CPU, status, entre outros.
- Próximo Nó (prox):
 - Tipo: struct _No* (ponteiro para o próximo nó)
 - Descrição: Mantém um ponteiro para o próximo nó na lista encadeada. Esse campo é usado para conectar os nós consecutivos na lista, permitindo a travessia da lista encadeada.

Em resumo, a estrutura _No é fundamental para a implementação de uma lista encadeada, armazenando informações sobre um processo específico e mantendo um ponteiro para o próximo nó na lista.

6.1.2. Estrutura Fila (_Fila):

A estrutura _Fila é projetada para representar uma fila utilizando uma lista encadeada. Ela possui dois campos principais:

- Início da Fila (inicio):
 - Tipo: No* (ponteiro para o início da fila)
 - Descrição: Mantém um ponteiro para o primeiro nó (elemento) na fila. Esse ponteiro é atualizado à medida que novos elementos são enfileirados ou elementos são removidos, indicando sempre o início atual da fila.

- Fim da Fila (fim):
 - Tipo: No* (ponteiro para o fim da fila)
 - Descrição: Mantém um ponteiro para o último nó (elemento) na fila. Este ponteiro é atualizado à medida que novos elementos são enfileirados, indicando sempre o final atual da fila.

Em resumo, a estrutura `_Fila` fornece uma abstração para operações de fila, permitindo a manipulação eficiente de elementos na ordem em que foram enfileirados. A lista encadeada é uma escolha comum para implementar filas, pois permite fácil inserção no final e remoção no início da fila.

6.2. Funções para Manipulação de Filas

6.2.1. *aloca_fila*

Aloca dinamicamente o espaço para uma fila.

6.2.2. *inicializa_fila*

Inicializa uma fila recém-criada, garantindo que os ponteiros de início e fim estejam corretamente configurados.

6.2.3. *fila_vazia*

Verifica se a fila está vazia.

6.2.4. *aloca_no*

Aloca dinamicamente o espaço para um nó.

6.2.5. *enfileira*

Adiciona um processo ao final da fila.

6.2.6. *enfileira_inicio*

Adiciona um processo no início da fila.

6.2.7. *desenfileira*

Remove e retorna o primeiro processo da fila.

6.2.8. *esvazia_fila*

Libera a memória ocupada pelos nós de uma fila.

6.2.9. imprime_fila

Imprime os processos presentes na fila.

6.2.10. imprime_fila_formatada

Imprime os processos de uma fila de forma formatada, exibindo o nome da fila.

6.2.11. imprime_filas

Imprime os processos presentes em todas as filas do sistema.

7. Header "processo.h"

Arquivo de cabeçalho contendo as estruturas de dados e funções de manipulação e controle de processos.

7.1. Estruturas de Dados

Define as estruturas de dados essenciais para a representação de processos, incluindo informações sobre operações de E/S.

7.1.1. Enumeração TipoES:

A enumeração TipoES é utilizada para representar os diferentes tipos de operações de E/S. Ela possui os seguintes valores:

- DISCO:
 - Valor: 0
 - Descrição: Indica que a operação de E/S é do tipo disco.
- FITA:
 - Valor: 1
 - Descrição: Indica que a operação de E/S é do tipo fita.
- IMPRESSORA:
 - Valor: 2
 - Descrição: Indica que a operação de E/S é do tipo impressora.

Essa enumeração é útil para diferenciar os diversos tipos de operações de E/S associadas a um processo.

7.1.2. Estrutura OperacaoES:

A estrutura OperacaoES é utilizada para armazenar informações relacionadas às operações de E/S. Ela possui os seguintes campos:

- Tipo de E/S (tipo_es):
 - Tipo: TipoES
 - Descrição: Indica o tipo de operação de E/S (disco, fita ou impressora).
- Duração da Operação de E/S (duracao_es):
 - Tipo: int
 - Descrição: Representa o tempo total necessário para concluir a operação de E/S.
- Tempo de Início da Operação de E/S (tempo_inicio):
 - Tipo: int
 - Descrição: Indica o tempo em que a operação de E/S deve começar.
- Tempo Restante para Finalizar a Operação de E/S (tempo_restante):
 - Tipo: int
 - Descrição: Mantém o tempo restante para a conclusão da operação de E/S.

Essa estrutura é fundamental para representar as características das operações de E/S associadas a um processo específico.

7.1.3. Enumeração StatusProcesso:

A enumeração StatusProcesso é utilizada para representar os diferentes estados de um processo. Ela possui os seguintes valores:

- PRONTO:
 - Valor: 0
 - Descrição: Indica que o processo está pronto para ser executado.

- EXECUTANDO:
 - Valor: 1
 - Descrição: Indica que o processo está sendo executado.
- ENTRADA_SAIDA:
 - Valor: 2
 - Descrição: Indica que o processo está em uma operação de entrada/saída.

Essa enumeração é útil para rastrear o estado atual de um processo no sistema operacional.

7.1.4. Estrutura Processo:

A estrutura Processo é utilizada para armazenar informações relacionadas a um processo no sistema operacional. Ela possui os seguintes campos:

- Identificador do Processo (pid):
 - Tipo: int
 - Descrição: Identifica de forma única o processo.
- Tempo de Início do Processo (tempo_inicio):
 - Tipo: int
 - Descrição: Indica o tempo em que o processo foi iniciado.
- Tempo de CPU do Processo (tempo_cpu):
 - Tipo: int
 - Descrição: Representa o tempo total de CPU necessário para concluir o processo.
- Tempo de Fim do Processo (tempo_fim):
 - Tipo: int
 - Descrição: Indica o tempo em que o processo foi concluído.
- Tempo de Turnaround do Processo (tempo_turnaround):

- Tipo: int
 - Descrição: Representa o tempo total que um processo leva desde a submissão até a conclusão.
- Tempo de Quantum Restante do Processo (tempo_quantum_restante):
 - Tipo: int
 - Descrição: Mantém o tempo de quantum restante para o processo em execução.
- Tempo de CPU Restante do Processo (tempo_cpu_restante):
 - Tipo: int
 - Descrição: Mantém o tempo de CPU restante para a conclusão do processo.
- Tempo de CPU Atual do Processo (tempo_cpu_atual):
 - Tipo: int
 - Descrição: Mantém o tempo de CPU atual durante a execução do processo.
- Operações de E/S (operacoes_es):
 - Tipo: OperacaoES* (ponteiro para um vetor de operações de E/S)
 - Descrição: Mantém informações sobre as operações de E/S associadas ao processo.
- Quantidade de Operações de E/S (num_operacoes_es):
 - Tipo: int
 - Descrição: Representa o número total de operações de E/S associadas ao processo.
- Índice da Operação de E/S Atual (operacao_es_atual):
 - Tipo: int
 - Descrição: Indica o índice da operação de E/S atual no vetor de operações de E/S.

- Status do Processo (status_processo):
 - Tipo: StatusProcesso
 - Descrição: Indica o estado atual do processo (pronto, executando ou em operação de entrada/saída).

Essa estrutura é fundamental para representar as características e o estado de um processo no contexto de um sistema operacional multitarefa.

7.2. Funções de Manipulação de Processos

7.2.1. aloca_processo

Aloca dinamicamente o espaço para um processo.

7.2.2. aloca_processos

Aloca dinamicamente o espaço para um vetor de processos.

7.2.3. aloca_operacoes_es

Aloca dinamicamente o espaço para um vetor de operações de E/S.

7.2.4. seleciona_tempo_es

Seleciona o tempo de uma operação de E/S com base no tipo de dispositivo.

7.2.5. seleciona_tipo_es

Retorna uma string representando o tipo de dispositivo de E/S.

7.2.6. seleciona_status_processo

Retorna uma string representando o status do processo.

7.2.7. cria_processo

Cria um processo com o PID fornecido.

7.2.8. inicializa_processos

Inicializa a quantidade especificada de processos, retornando um vetor de processos inicializados.

7.2.9. executa_processo

Executa um processo, alterando seu estado e valores relacionados.

7.2.10. processo_finalizado

Verifica se um processo finalizou sua execução.

7.2.11. tempo_inicio_es

Verifica se é o momento de iniciar uma operação de E/S para um processo.

7.2.12. executa_es

Executa uma operação de E/S para um processo.

7.2.13. es_finalizada

Verifica se uma operação de E/S finalizou.

7.2.14. tempo_quantum_completo

Verifica se um processo atingiu o tempo de quantum.

8. Saída do Escalonador

Visto que a construção de processos e de operações de entrada e saída é feita de forma aleatória no sistema, a saída abaixo dificilmente poderá ser a mesma reproduzida durante a apresentação do trabalho. Mas, para ilustrar o funcionamento adequado do escalonador, não poderíamos esquecer de colocá-la aqui. Segue abaixo o exemplo transcrito da saída do terminal:

===== PROCESSOS =====

PID	Tempo de servico	Tempo de inicio	E/S (Tempo de inicio)
P0	9	3	Nenhuma operacao de E/S.
P1	10	0	Fita (1)
P2	6	4	Fita (3), Fita (3)
P3	5	0	Fita (3)

===== INSTANTE 0 =====

O processo P1 entrou na fila de alta prioridade.

O processo P3 entrou na fila de alta prioridade.

O processo P1 executou por 1 u.t.

O processo P1 foi para a fila de E/S (Fita).

Fila de alta prioridade: P3

Fila de baixa prioridade:

Fila de disco:

Fila de fita: P1

Fila de impressora:

===== INSTANTE 1 =====

O processo P3 executou por 1 u.t.

O processo P1 executou 1 u.t. da sua E/S de Fita, faltam 2 u.t.

Fila de alta prioridade: P3

Fila de baixa prioridade:

Fila de disco:

Fila de fita: P1

Fila de impressora:

===== INSTANTE 2 =====

O processo P3 executou por 1 u.t.

O processo P3 sofreu preempcao, vai pra fila de baixa prioridade.

O processo P1 executou 1 u.t. da sua E/S de Fita, faltam 1 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P3

Fila de disco:

Fila de fita: P1

Fila de impressora:

===== INSTANTE 3 =====

O processo P0 entrou na fila de alta prioridade.

O processo P0 executou por 1 u.t.

O processo P1 executou 1 u.t. da sua E/S de Fita, faltam 0 u.t.

O processo P1 finalizou sua E/S de Fita, vai para a fila de alta prioridade.

Fila de alta prioridade: P0 P1

Fila de baixa prioridade: P3

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 4 =====

O processo P2 entrou na fila de alta prioridade.

O processo P0 executou por 1 u.t.

O processo P0 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade: P1 P2

Fila de baixa prioridade: P3 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 5 =====

O processo P1 executou por 1 u.t.

Fila de alta prioridade: P1 P2

Fila de baixa prioridade: P3 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 6 =====

O processo P1 executou por 1 u.t.

O processo P1 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade: P2

Fila de baixa prioridade: P3 P0 P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 7 =====

O processo P2 executou por 1 u.t.

Fila de alta prioridade: P2

Fila de baixa prioridade: P3 P0 P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 8 =====

O processo P2 executou por 1 u.t.

O processo P2 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade:

Fila de baixa prioridade: P3 P0 P1 P2

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 9 =====

O processo P3 executou por 1 u.t.

O processo P3 foi para a fila de E/S (Fita).

Fila de alta prioridade:

Fila de baixa prioridade: P0 P1 P2

Fila de disco:

Fila de fita: P3

Fila de impressora:

===== INSTANTE 10 =====

O processo P0 executou por 1 u.t.

O processo P3 executou 1 u.t. da sua E/S de Fita, faltam 2 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P0 P1 P2

Fila de disco:

Fila de fita: P3

Fila de impressora:

===== INSTANTE 11 =====

O processo P0 executou por 1 u.t.

O processo P0 sofreu preempcao, vai pra fila de baixa prioridade.

O processo P3 executou 1 u.t. da sua E/S de Fita, faltam 1 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P1 P2 P0

Fila de disco:

Fila de fita: P3

Fila de impressora:

===== INSTANTE 12 =====

O processo P1 executou por 1 u.t.

O processo P3 executou 1 u.t. da sua E/S de Fita, faltam 0 u.t.

O processo P3 finalizou sua E/S de Fita, vai para a fila de alta prioridade.

O processo P1 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade: P3

Fila de baixa prioridade: P1 P2 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 13 =====

O processo P3 executou por 1 u.t.

Fila de alta prioridade: P3

Fila de baixa prioridade: P1 P2 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 14 =====

O processo P3 executou por 1 u.t.

O processo P3 finalizou sua execucao.

Fila de alta prioridade:

Fila de baixa prioridade: P1 P2 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 15 =====

O processo P1 executou por 1 u.t.

O processo P1 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade:

Fila de baixa prioridade: P2 P0 P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 16 =====

O processo P2 executou por 1 u.t.

O processo P2 foi para a fila de E/S (Fita).

Fila de alta prioridade:

Fila de baixa prioridade: P0 P1

Fila de disco:

Fila de fita: P2

Fila de impressora:

===== INSTANTE 17 =====

O processo P0 executou por 1 u.t.

O processo P2 executou 1 u.t. da sua E/S de Fita, faltam 2 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P0 P1

Fila de disco:

Fila de fita: P2

Fila de impressora:

===== INSTANTE 18 =====

O processo P0 executou por 1 u.t.

O processo P0 sofreu preempcao, vai pra fila de baixa prioridade.

O processo P2 executou 1 u.t. da sua E/S de Fita, faltam 1 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P1 P0

Fila de disco:

Fila de fita: P2

Fila de impressora:

===== INSTANTE 19 =====

O processo P1 executou por 1 u.t.

O processo P2 executou 1 u.t. da sua E/S de Fita, faltam 0 u.t.

O processo P2 finalizou sua E/S de Fita, vai para a fila de alta prioridade.

O processo P1 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade: P2

Fila de baixa prioridade: P1 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 20 =====

O processo P2 executou por 1 u.t.

Fila de alta prioridade: P2

Fila de baixa prioridade: P1 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 21 =====

O processo P2 executou por 1 u.t.

O processo P2 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade:

Fila de baixa prioridade: P1 P0 P2

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 22 =====

O processo P1 executou por 1 u.t.

O processo P1 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade:

Fila de baixa prioridade: P0 P2 P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 23 =====

O processo P0 executou por 1 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P0 P2 P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 24 =====

O processo P0 executou por 1 u.t.

O processo P0 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade:

Fila de baixa prioridade: P2 P1 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 25 =====

O processo P2 executou por 1 u.t.

O processo P2 finalizou sua execucao.

Fila de alta prioridade:

Fila de baixa prioridade: P1 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 26 =====

O processo P1 executou por 1 u.t.

Fila de alta prioridade:

Fila de baixa prioridade: P1 P0

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 27 =====

O processo P1 executou por 1 u.t.

O processo P1 sofreu preempcao, vai pra fila de baixa prioridade.

Fila de alta prioridade:

Fila de baixa prioridade: P0 P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 28 =====

O processo P0 executou por 1 u.t.

O processo P0 finalizou sua execucao.

Fila de alta prioridade:

Fila de baixa prioridade: P1

Fila de disco:

Fila de fita:

Fila de impressora:

===== INSTANTE 29 =====

O processo P1 executou por 1 u.t.

O processo P1 finalizou sua execucao.

Nenhuma fila com processos, o processador esta ocioso.

Nao ha mais processos, e portanto, o escalonamento foi finalizado com sucesso.

===== FIM DO ESCALONAMENTO =====

Tempos de turnaround de cada um dos processos:

P0: 25 u.t

P1: 29 u.t

P2: 21 u.t

P3: 14 u.t

Turnaround medio: 22.25 u.t.

9. Conclusão

Este relatório detalhou as funções presentes nos headers "fila.h" e "processo.h", fornecendo uma visão mais abrangente das operações realizadas no simulador de escalonamento de processos. Cada função foi descrita para fornecer uma compreensão mais profunda do seu propósito e funcionamento. O código busca oferecer uma experiência educativa e esclarecedora para estudantes de Computação.

10. Referências

- Brian W Kernighan and Dennis M Ritchie. The C programming language. 2002.
- Herbert Schildt. C: Completo e Total. 1997.
- Carlos A Maziero. Sistemas operacionais: conceitos e mecanismos. Livro aberto, 2014.

- William Stallings. Operating systems internals and design principles. Prentice-Hall, Inc., 1998.