

Sistemas Operacionais: Simulador de SO

Nome: Diogo Buzatto - RA: 111809
Nome: Lucas Ferreira - RA: 111519
Nome: Mauricio Sarpa - RA: 112824
Nome: Pablo Novais Ramos - RA:
112070

1 INTRODUÇÃO

Este trabalho apresenta um simulador de Sistema Operacional desenvolvido em C++ com ênfase na aplicação prática de conceitos fundamentais da disciplina, como gerenciamento de processos, memória e dispositivos de entrada/saída. O projeto utiliza os princípios da programação orientada a objetos para estruturar os principais componentes do sistema.

2 OBJETIVOS

- Simular o funcionamento básico de um sistema operacional;
- Aplicar conceitos de orientação a objetos (herança, encapsulamento e polimorfismo);
- Implementar classes para processos, escalonadores, gerenciadores de memória e dispositivos de E/S;
- Analisar o comportamento do sistema com diferentes estratégias de escalonamento e alocação de memória.

3 FUNDAMENTAÇÃO TEÓRICA

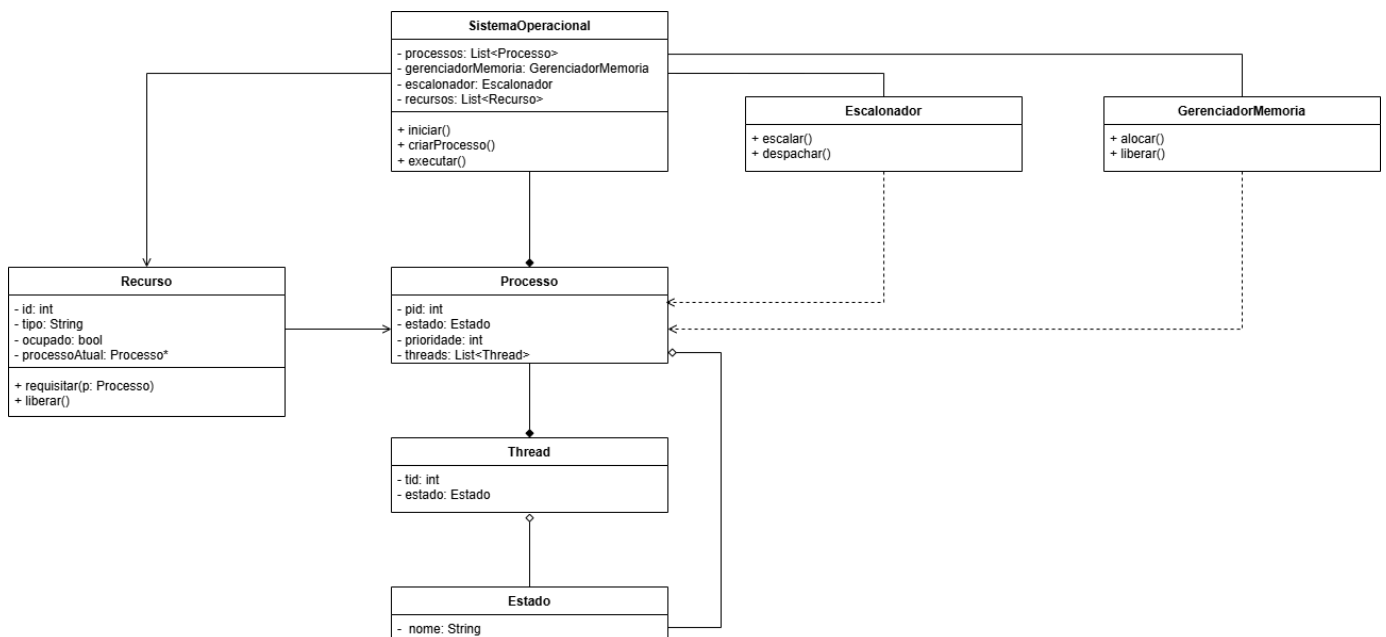
A simulação é baseada nos conceitos descritos por autores como Tanenbaum (2015) e Silberschatz et al. (2018):

- Gerenciamento de Processos: Controle da criação, execução e finalização de processos com diferentes prioridades, tempos de execução e chegada.
- Escalonamento: Implementados dois algoritmos clássicos: Round Robin (com quantum configurável) e Prioridade.
- Gerenciamento de Memória: Suporte à alocação via partições fixas e memória paginada, permitindo comparar a eficiência e fragmentação de cada abordagem.
- Entrada/Saída: Dispositivos simulados que gerenciam filas de espera e controle de acesso por processo.

4 DESENVOLVIMENTO

O sistema é modular, com os seguintes componentes principais:

Diagrama de classes:



- Processo: Contém PID, prioridade, tempo de chegada, execução e estado atual.
- Escalonadores:
 - Round Robin: tempo de CPU fixo por processo (quantum).

- Prioridade: seleciona o processo pronto com menor valor de prioridade.
- Gerenciadores de Memória:
 - Partições Fixas: memória dividida em blocos de tamanho fixo.
 - Paginação: memória dividida em páginas, com alocação dinâmica.
- Dispositivos de E/S: Utilizam fila FIFO. O processo entra na fila, aguarda e “bloqueia” o recurso até ser liberado.
- Sistema Operacional: Instância e gerencia os componentes, executa os ciclos de simulação e exibe o resumo final.

5 RESULTADOS

A simulação é interativa via terminal, com as seguintes etapas:

1. Criação de processos (aleatórios);
2. Escolha do tipo de escalonador e memória;
3. Execução da simulação:
 - Processos são alocados, executados e finalizados;
 - O estado dos processos é atualizado conforme a execução;
 - A memória é liberada ao final de cada processo.
4. Exibição de resumo com tempo total e memória disponível.

6 CONCLUSÃO

O projeto contribuiu significativamente para a compreensão prática dos conceitos de sistemas operacionais. Foi possível observar de forma concreta como o gerenciamento de memória, o escalonamento e o controle de E/S afetam o desempenho do sistema. Além disso, o uso da linguagem C++ reforçou habilidades de programação orientada a objetos e manipulação de ponteiros.

7 MELHORIAS FUTURAS

- Implementar o estado “Bloqueado” para simular espera por E/S;
- Utilizar tempos de chegada progressivos para os processos;
- Calcular métricas por processo, como tempo de espera, turnaround e tempo de CPU utilizado;
- Criar uma interface gráfica para visualização dos estados e da memória.

8 REFERÊNCIAS

- TANENBAUM, A. S.; BOS, H. Modern Operating Systems. 4. ed. Pearson, 2015.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. Operating System Concepts. 10. ed. Wiley, 2018.