Relatório – Trabalho Prático de Sistemas Operacionais

Aluno: Thiago Olivio Mendes
RA: 113383
Disciplina: Sistemas Operacionais

1. Introdução

Este trabalho apresenta a implementação de um simulador didático de Sistema Operacional em Python, desenvolvido como atividade prática da disciplina de Sistemas Operacionais. O objetivo foi aplicar os conceitos estudados em sala de aula, modelando processos, threads, gerenciamento de memória, escalonamento, sistema de arquivos e dispositivos de entrada/saída, além de coletar métricas de desempenho.

2. Estrutura Geral do Código

O simulador foi implementado em um único arquivo Python (SO.py), compatível com Python 3.8+ e sem dependências externas.

Principais blocos implementados: PCB, TCB, estados de thread, gerenciador de memória, sistema de arquivos, dispositivos de E/S, escalonadores e coleta de métricas.

3. Escalonamento de Processos

O simulador oferece três algoritmos de escalonamento: FCFS, Round Robin (com quantum configurável) e Prioridade (preemptivo). As trocas de contexto são contabilizadas e incluídas nas métricas finais.

4. Gerenciamento de Memória

Foi implementada paginação por demanda, com contagem de falhas de página e substituição simples em caso de falta de molduras. As falhas de página são registradas e apresentadas nas métricas finais.

5. Entrada e Saída

Foram simulados dois dispositivos: disco (disk) e terminal (tty). Cada dispositivo possui tempo de serviço, fila de espera e operações bloqueantes (io) e não bloqueantes (io_nb).

6. Sistema de Arquivos

Foi implementado um sistema de arquivos simplificado com estrutura hierárquica, criação, escrita, leitura e listagem de arquivos.

7. Métricas Coletadas

Ao final da execução, o simulador apresenta métricas em formato JSON, incluindo tempo de simulação, utilização de CPU, trocas de contexto, falhas de página, turnaround médio e tempo médio em fila de pronto.

8. Workload de Teste

O simulador inclui um workload padrão com dois processos e múltiplas threads, utilizando operações de CPU, memória, E/S e arquivos.

9. Checklist da Especificação

Requisito	Status
PCB/TCB com informações de processos	Implementado
Estados de thread	Implementado
Escalonadores FCFS, RR, Prioridade	Implementado
Contagem de trocas de contexto	Implementado
Paginação de memória + page faults	Implementado
Dispositivos de E/S com bloqueio	Implementado
Operações de sistema de arquivos	Implementado
Coleta de métricas de desempenho	Implementado
Log textual de eventos	Implementado
Relatório/documentação	Implementado

10. Conclusão

O trabalho resultou em um simulador funcional de Sistema Operacional em Python, cobrindo os principais conceitos da disciplina. A implementação é didática, roda em qualquer ambiente Python 3.8+ e serve como base para experimentos e futuras extensões.

11. Instruções de Execução

• Versão Python recomendada: 3.8 ou superior
• Dependências externas: nenhuma
• Execução no Spyder: abrir SO.py e clicar em Run
• Execução via prompt: python SO.py --sched rr --quantum 3
• Se rodar por duplo clique, o programa imprime métricas e aguarda Enter para encerrar.