

Trabalho Final: Simulação de Arquitetura Multicore com Gerenciamento de Memória e Escalonamento de Processos

Michel Pires, Centro Federal de Educação Tecnológica de Minas Gerais

Data de Entrega: 06/12/2025

Valor: 20 pontos (10 Artigo + 10 Etapa de escalonamento + 10 Etapa de gerencia de memória)

1 Apresentação Geral

Este trabalho tem como objetivo consolidar os conhecimentos referentes às arquiteturas computacionais modernas e aos mecanismos de gerenciamento de processos e memória, por meio do desenvolvimento de um **simulador de arquitetura multicore**. A proposta dá continuidade ao projeto anterior, baseado em uma arquitetura de von Neumann com pipeline MIPS, ampliando-o para contemplar um ambiente com múltiplos núcleos de processamento, gerência de memória segmentada e diferentes estratégias de escalonamento de processos.

Cada equipe deverá ser composta por **quatro alunos**, responsáveis pelo planejamento, implementação, validação e documentação do simulador. O resultado final deverá ser apresentado sob a forma de um **artigo técnico no formato IEEE**, seguindo o modelo disponível em:

<https://pt.overleaf.com/latex/templates/ieee-conference-template/grfzhnscsfqn>

2 Descrição Geral do Sistema

O simulador a ser desenvolvido deverá representar uma arquitetura **multicore**, composta por um conjunto de n núcleos (definido pela equipe), que compartilham uma **memória principal unificada**. O sistema deve ser capaz de:

1. Ler do disco um **lote inicial de programas** previamente definido. Não é permitida chegada de novos processos durante a execução;
2. Carregar todos os programas na memória principal, realizando o mapeamento de endereços segundo um modelo inspirado em Tanenbaum, onde:
 - A memória é acessada por palavras de x bits;
 - Parte dos bits representa o endereço do **bloco** e parte o **deslocamento** interno;
 - O gerenciamento de memória deve prever políticas de substituição (ex.: FIFO, LRU);
3. Executar os programas de forma concorrente entre os núcleos, sob diferentes políticas de **escalonamento**, como:
 - **First Come, First Served (FCFS)**;

- **Shortest Job Next (SJN)**;
 - **Round Robin (RR)**;
 - **Prioridade** (preemptiva ou não preemptiva).
 - **Outros** ...
4. Coletar métricas de desempenho para comparação entre os diferentes modos de execução:
- Tempo médio de espera;
 - Tempo médio de execução;
 - Utilização média da CPU;
 - Eficiência e taxa de throughput global.

Ao final da execução do lote, o simulador deve apresentar:

- Um relatório resumido de desempenho de cada política de escalonamento;
- A utilização de memória ao longo do tempo;

3 Entrega e Produto Final

A entrega deverá conter os seguintes itens:

1. **Implementação completa do simulador multicore**, devidamente comentada e publicada em repositório público (*GitHub*);
2. **Artigo científico** descrevendo o trabalho desenvolvido, no formato IEEE Conference Template, contendo no mínimo as seções:
 - Resumo;
 - Introdução;
 - Referencial Teórico (arquiteturas multicore, escalonadores, gerenciamento de memória);
 - Metodologia e Implementação;
 - Resultados e Discussão;
 - Conclusão e Trabalhos Futuros;
 - Referências.

Recomenda-se que o artigo inclua figuras esquemáticas da arquitetura implementada, fluxogramas dos módulos e pseudocódigos representando as políticas de escalonamento e alocação de memória.

4 Execução e Simulação

Para fins comparativos, deve-se utilizar como *baseline* a arquitetura *single-core* previamente desenvolvida. A nova versão multicore deverá ser avaliada em condições experimentais equivalentes, permitindo a análise direta do impacto da paralelização e das diferentes estratégias de escalonamento e gerenciamento de memória sobre o desempenho global do sistema. Em ambas as arquiteturas, devem ser observadas as seguintes diretrizes de execução:

- **Carga inicial de programas:** todos os programas pertencentes ao lote devem ser completamente carregados na memória principal antes do início da execução. Após essa etapa, não será permitida a chegada de novos processos ao sistema, garantindo um ambiente de simulação controlado e determinístico;
- **Distribuição de tarefas:** a alocação das tarefas entre os núcleos deve ocorrer de acordo com a política de escalonamento selecionada. Devem ser considerados dois cenários experimentais distintos:
 1. **Cenário não preemptivo:** todas as tarefas são executadas até a conclusão, sem interrupções, respeitando a ordem determinada pelo escalonador;
 2. **Cenário preemptivo:** todas as tarefas são passíveis de interrupção, sendo executadas conforme um *quantum* de tempo definido pelo escalonador. O simulador deve gerenciar o contexto de execução, preservando o estado dos processos interrompidos para retomada posterior.
- **Execução do ciclo de processamento:** o simulador deve executar o ciclo completo de instruções até que todas as tarefas do lote inicial sejam concluídas, garantindo a sincronização entre os núcleos e a atualização consistente dos estados de memória compartilhada;
- **Registro de métricas:** todas as execuções devem gerar arquivos de *log* contendo métricas de desempenho relevantes, tais como tempo médio de espera, tempo médio de retorno, utilização média da CPU, eficiência por núcleo e *throughput* total do sistema. Essas métricas deverão ser utilizadas para análise comparativa entre as políticas de escalonamento e entre as arquiteturas *single-core* e *multicore*.

Os resultados obtidos devem ser apresentados e discutidos no artigo técnico, destacando a influência das políticas de escalonamento e da gerência de memória no comportamento global do sistema. Recomenda-se, sempre que possível, a inclusão de gráficos ou tabelas comparativas para evidenciar o impacto da arquitetura multicore em relação à *baseline*. Durante a execução não é permitida a entrada de novos programas. O encerramento da simulação ocorrerá apenas quando todas as tarefas do lote inicial forem concluídas.

5 Critérios de Avaliação

A nota final será composta pelos seguintes elementos:

Implementação (20 pontos: 10 escalonamento + 10 gerencia de memória)

- Correção e completude do simulador;
- Organização modular do código;
- Adequação à teoria de Tanenbaum sobre memória e escalonamento;
- Validação experimental e consistência dos resultados.

Artigo Científico (10 pontos)

- Clareza e coerência técnica da escrita;
- Rigor conceitual e fundamentação teórica;
- Qualidade das análises e interpretação dos resultados;
- Uso correto do modelo IEEE e das referências científicas.

6 Apresentação e Submissão

A entrega do artigo deverá ser realizada via AVA até o dia **06/12/2025**. O repositório contendo o simulador e instruções de compilação deve ser referenciado no artigo e permanecer público até a data de avaliação.

Cada equipe poderá ser convidada a apresentar brevemente (5 a 10 minutos) os principais resultados obtidos, destacando a comparação entre políticas de escalonamento e desempenho sob diferentes parâmetros de memória.

7 Considerações Finais

Este trabalho busca proporcionar aos alunos uma experiência prática no estudo das interações entre hardware e software, destacando como o gerenciamento eficiente de processos e memória impacta o desempenho de sistemas multiprocessados. A ênfase recai sobre a observação empírica e análise quantitativa dos resultados simulados, articulando teoria e prática em um contexto de arquitetura moderna.

Data de Entrega: 06/12/2025

Valor Total: 30 pontos (20 Implementação (Escalonamento + Memória) + 10 Artigo Científico)