



**Departamento de Engenharias, Arquitetura e  
Computação**

Disciplina de Sistemas Operacionais

**Prof<sup>a</sup>.: Ivan L. Süptitz**

**Trabalho – Entrega em 25/06/2020**

**Leia com atenção:**

1. Este trabalho vale 25% da nota final, deve ser realizado grupos de dois ou três integrantes;
2. A formação dos grupos deve ocorrer no dia da especificação do trabalho
3. Deve ser entregue um arquivo compactado com o projeto completo do programa e um relatório de desenvolvimento. Deve ser postado na tarefa correspondente do virtual até as 19h00 do dia da entrega.
4. A apresentação ocorre durante a aula do mesmo dia da entrega, quando cada integrante do grupo deve demonstrar e explicar todas as funcionalidades implementadas.
5. Trabalhos entregues, mas não apresentados, serão desconsiderados.
6. Trabalhos apresentados, mas sem os arquivos corretamente entregues no virtual (ou zip corrompido) serão desconsiderados.

**Objetivo:**

Implementar um simulador de gerenciamento de memória.

**Características básicas:**

- A linguagem para implementação do simulador é livre, bem como o sistema operacional no qual o simulador irá executar;
- Considere a existência de uma MMU que disponibiliza registrador base e registrador de limite;
- Considere a existência de uma memória de 1Mb (um milhão de bits) sendo que 50% estará ocupada pelo Sistema Operacional. Essa memória deve ser simulada por meio de um arquivo texto no qual cada bit será um caracter. Desta forma, logo ao abrir o simulador, o arquivo “memoria.txt” deve ser criado já com metade dele preenchido com o caracter X e a outra metade com espaços vazios. As operações de escrita em memória devem se refletir em escrever os ‘bits’ no respectivo endereço solicitado.
- Considere a existência de um único processador, com escalonamento FIFO de forma que somente um processo pode estar fazendo as requisições de memória por vez. Ele vai deixar o processador somente quando terminar a execução, fizer uma E/S ou quando fizer uma operação de acesso ilegal de memória.

**Funcionalidades:**

- O simulador deve implementar os seguintes algoritmos de gerenciamento de partições variáveis (usuário vai escolher qual deles ao iniciar o programa):
  - First-Fit;
  - Best-Fit;
  - Worst-Fit;
  - Circular-Fit;
- A entrada dos dados ocorre por meio de um arquivo texto, conforme modelo disponibilizado no virtual:

- Cada linha do arquivo representa um processo;
- Os campos são separados uns dos outros por meio do caracter pipe “|”
- O primeiro campo é o identificador do processo;
- O segundo campo indica a quantidade de memória que o processo está requisitando (em bits);
- Do 3º ao último campo, são as instruções do processo:
  - “-“ indica uma operação na CPU que não envolve acesso a memória, portanto pode ser ignorada;
  - ES indica que o processo está pedindo uma operação de Entrada ou Saída. Esse é o momento em que ele deixa o processador e devemos tratar o processo seguinte. Observe que o processo ainda não encerrou e, pela regra do FIFO, ele volta ao final da fila e depois de todos serem tratados, volta a receber o processador;
  - lw,20 indica uma operação de leitura de uma palavra (4 bits) no endereço lógico de memória 20;
  - sw,0001,500 indica uma operação de gravação da palavra 0001 no endereço lógico de memória 500;
- Como saída, o programa deve exibir o um log (em tela ou arquivo) indicando o passo a passo da execução:
  - início e fim do espaço destinado ao processo;
  - valores lidos e escritos (qual valor e em qual posição física da memória);
  - início e término da execução de cada processo;
  - saída por E/S ou acesso ilegal;
  - lista de lacunas (páginas livres).

### **Avaliação**

**(4,0 pts)** Correto funcionamento do programa: avaliado com o log gerado como saída e com o arquivo memoria.txt;

**(3,0 pts)** Correta utilização dos conceitos vistos em aula: lista de partições livres, tradução de endereço lógico em físico com tratamento de acesso ilegal e algoritmos de gerenciamento das partições;

**(1,5 pts)** Relatório: entregue em PDF com a documentação completa do programa: descrição do funcionamento, estruturas de dados utilizadas, print das telas e lógica de funcionamento.

**(1,5 pts)** Apresentação e demonstração do trabalho (nota individual de cada integrante do grupo).

**Bom trabalho**