Universidade do Estado de Santa Catarina – Bacharelado em Ciência da Computação
Sistemas Operacionais – Prof. Rafael R. Obelheiro – Semestre: 2020.2

Trabalho 2 – Escalonamento e Gerência de Memória

Sobre o trabalho

- O trabalho é individual.
- Exercícios de implementação devem ser realizados na linguagem C, e serão testados no sistema operacional Linux. As submissões estarão sujeitas a controle de plágio usando ferramentas de detecção de similaridade.
- Exercícios teóricos devem ser submetidos em formato PDF. Para os exercícios de implementação deve ser submetido o código fonte, em formato compilável (ou seja, os arquivos `*.c`, e não listagens em PDF). Submeta um único arquivo ZIP contendo todas as suas respostas (teóricas e de implementação).
- O trabalho deverá ser entregue até **TERÇA-FEIRA, 02 DE MARÇO**. O Moodle aceitará submissões até 23h59min, sendo automaticamente bloqueado após a data limite. É **RESPONSABILIDADE DOS ALUNOS** garantir que o trabalho seja entregue no prazo.
- Em caso de dificuldades ou dúvidas na interpretação do trabalho, entre em contato com o professor (`rafael.obelheiro@udesc.br`).

Exercícios

1. Três programas devem ser executados em um computador monoprocessado. Todos os programas são compostos por dois ciclos de processador e um ciclo de E/S. A entrada e saída de todos os programas é feita sobre a mesma unidade de disco. Os tempos para cada ciclo de cada programa são mostrados abaixo:

Programa	CPU	Disco	CPU	instante de chegada
A	2	4	5	3
B	4	4	2	1
C	9	4	4	0

   Observe que nem todos os processos chegam no mesmo instante. O algoritmo de escalonamento de CPU usado no sistema é o de múltiplas filas com realimentação, onde as filas são escalonadas por prioridade e os processos em cada fila por *round-robin*, de acordo com o seguinte esquema:

   - fila de alta prioridade: $quantum = 2$
   - fila de baixa prioridade: $quantum = 5$

   Os processos iniciam um ciclo de CPU na fila de alta prioridade, e mudam de fila caso não tenham concluído seu ciclo de CPU ao término do *quantum*. A chegada de um processo na fila de alta prioridade preempta um processo de baixa prioridade que esteja em execução; quando isso ocorre, o processo preemptado permanece na mesma posição da fila de baixa prioridade, e ao retomar a CPU ele executa pelo tempo remanescente do *quantum*.

   (a) Construa um diagrama de tempo mostrando qual programa está ocupando o processador e o disco a cada momento, até que todos os programas terminem.

   (b) Determine o tempo médio de espera para o conjunto de processos.

   (c) Determine o tempo médio de retorno para o conjunto de processos.

   (d) Determine a vazão para essa escala, sabendo que cada unidade de tempo equivale a 0,1 s.

   (e) Esse algoritmo está sujeito a inanição? Justifique.

2. Considere um sistema de paginação simples, com endereços virtuais de 16 bits e páginas de 8 KB, onde o processo P1 tem a seguinte tabela de páginas:

	pág. física	válido
0	6	1
1	7	1
2	3	0
3	3	1
4	0	1
5	1	0
6	9	1
7	4	0

   (a) Acesse o endereço `https://bit.ly/2M62kaX` para obter um conjunto de 5 números inteiros aleatórios entre 1000 e 65000. Liste quais foram os números obtidos.

   (b) Considere os números gerados como endereços virtuais gerados por P1, e determine os endereços físicos correspondentes. Caso ocorra falta de página, aloque uma página física livre do conjunto $\{1, 2, 4, 8\}$ (você pode escolher qualquer uma) para a página virtual que gerou a falta, e determine o endereço físico resultante.

   IMPORTANTE: Suas respostas devem demonstrar claramente todos os passos da tradução de endereços. Respostas que contenham apenas o endereço físico resultante serão desconsideradas.

3. Considere um sistema de paginação multinível com endereços virtuais de 36 bits e páginas de 2 KB. Cada entrada de tabela de páginas ocupa 32 bits. O espaço ocupado por uma tabela de páginas deve ser igual ou inferior a uma página.

   (a) Quantos níveis de tabelas de páginas são necessários para atender às especificações acima?

   (b) Qual o tamanho (em número de entradas) das tabelas de páginas em cada um desses níveis? (Dica: nem todas as tabelas podem ter o mesmo tamanho.)

   (c) Em qual dos níveis da hierarquia seria melhor colocar a tabela de páginas com menos entradas? Justifique sua resposta.

   (d) Decomponha o endereço hexadecimal `0xbadc0ffee` nos componentes de número de página e deslocamento de acordo com a estrutura de endereço virtual identificada nos itens anteriores. (Dica: use a representação binária do endereço para fazer a decomposição.)

4. Em um sistema que usa paginação por demanda, um processo pode executar em três páginas físicas (inicialmente vazias).

   (a) Acesse o endereço `https://bit.ly/3ugUQTQ` para obter uma sequência de 15 números inteiros aleatórios. Liste quais foram os números obtidos.

   (b) Considere que esses números representam uma sequência de referências a páginas virtuais. Determine o número de faltas de página para essa sequência quando são usados os algoritmos de substituição FIFO e MRU.

   IMPORTANTE: A ordem de acesso às páginas é altamente relevante neste exercício. Você deve preservar a sequência gerada pelo serviço web.