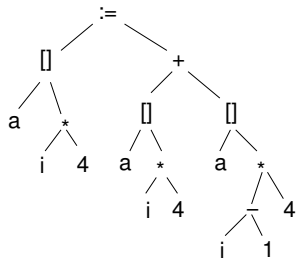
# Prova 3 de EA879 — Turma U, novembro de 2008

**1** Um compilador gerou, para um comando de um programa em linguagem C com inteiros de 4 bytes, a seguinte árvore sintática abstrata:




(a) Qual o comando em C?

(b) Apresente a representação do código intermediário correspondente a essa árvore sintática abstrata usando a notação pós-fixa.

(c) Quais são as possibilidades de otimização nesse fragmento de código?

(d) Os projetistas da linguagem C adotaram arranjos com base 0 (isto é, o índice do primeiro elemento do arranjo é 0 em vez de 1) como uma estratégia de geração de código mais eficiente. Por que isso é verdade?

**2** Um sistema de memória virtual foi configurado para trabalhar com endereços virtuais de 21 bits (cada endereço referencia uma palavra de 1 byte) e páginas de 2 KiBytes (1 Ki = 1024). Para execução de cada processo, o sistema aloca 4 quadros. Cada entrada na tabela ocupa 4 bytes.

(a) Qual é, em KiBytes, o espaço de memória ocupado por cada processo, considerando os quadros alocados pelo sistema operacional e a tabela de páginas completa?

(b) Qual a página que contém o endereço 0x004A9E (19102 em decimal) e para qual endereço físico esse endereço é mapeado se a página for alocada ao quadro 2?

(c) A execução de um processo gera a seguinte seqüência de referências a páginas, na qual o índice indica se as operações executadas na página foram de apenas leitura (L) ou se houve pelo menos uma operação de escrita (E) na memória: $2_L\ 0_L\ 1_E\ 2_L\ \dagger\ 9_E\ 0_L$. O símbolo † indica o momento no qual ocorreu uma interrupção de relógio do sistema operacional. A próxima referência é a uma página que não está na memória. Qual dessas páginas será retirada para dar espaço a essa nova página se o algoritmo de troca de páginas adotado for *não usada recentemente* (NUR ou NRU)?

(d) Para os mesmos dados do item anterior, qual página seria retirada da memória se o algoritmo de troca de páginas utilizado fosse *menos recentemente usada* (MRU ou LRU)?

**3** Em um sistema operacional que adota o escalonamento por alternância (*round robin*) com quantum de 40 ms, o processo $P_A$ iniciou sua execução às 19h00, o processo $P_B$ às 19h01 e o processo $P_C$ às 19h02. O tempo total de CPU para a execução de cada processo é, respectivamente, de 2 minutos, 3 minutos e 1 minuto.

(a) Qual o horário de término de cada processo?

(b) Como a resposta ao item anterior é alterada se cada processo gasta em média 50% em tempo de espera para operações de entrada e saída?

(c) Qual seria o horário de término dos processos (sem tempo de espera para entrada e saída) se o escalonamento preemptivo por prioridades fosse adotado, sendo $P_C$ o processo de maior prioridade e $P_A$ o de menor prioridade?

(d) Explique o que é o problema da inversão de prioridades e dê um exemplo de como esse problema poderia ocorrer no caso do item anterior.

**4** Uma solução proposta para o problema da região crítica entre dois processos (0 e 1) utiliza um arranjo booleano de duas posições, `flag`, em memória compartilhada, e os seguintes procedimentos para entrar e sair da região crítica:

```
void entre(int processo) {
   int outro = 1 - processo;
   flag[processo] = true;
   while (flag[outro]) /* espera */ ;
}
void sai(int processo) {
   flag[processo] = false;
}
```

(a) Qual é o problema dessa solução? Dê um exemplo de uma condição de execução no qual esse problema possa ocorrer.

(b) Na solução de problema de comunicação interprocessos, em qual situação o programador deve utilizar a solução de Peterson em vez de uma solução baseada em espera bloqueada?

(c) Se as chamadas de sistema `sleep` e `wakeup` estão disponíveis para qualquer programador, por que um usuário qualquer não pode implementar semáforos caso o sistema operacional não ofereça essa funcionalidade?

(d) É possível usar semáforos para sincronizar threads em um sistema operacional que implementa threads no nível do usuário? Explique.