Sistemas Operacionais - 2020/1 - Escola Politécnica – PUCRS
Prof. Fernando Dotti
Questões da Avaliação G2

Orientações:

1. Entregue um **PDF** na sala de entrega apropriada, no dia da entrega marcado no Moodle.
2. O PDF deve estar legível, podendo ser feito à mão ou em computador.
3. As questões devem ser respondidas em ordem, sendo claramente numeradas na folha de respostas.
4. No caso de programas, coloque o fonte complete no arquivo PDF.

Questões:

1. Suponha que você deve propor estruturas internas para duas variantes de um sistema operacional:
   a. Usando partições fixas de memória
   b. Usando paginação

   Proponha de forma algoritmica a estrutura do bloco de controle de processo (*Process Control Block*) explicando o significado de cada campo.

2. Considere o seguinte algoritmo como tentativa de resolução da seção crítica por software. Assuma que cada linha é atômica,

boolean wantp ← false, wantq ← false	
p	**q**
loop forever	loop forever
p1: non-critical section	q1: non-critical section
p2: wantp ← true	q2: wantq ← true
p3: while wantq	q3: while wantp
p4: wantp ← false	q4: wantq ← false
p5: wantp ← true	q5: wantq ← true
p6: critical section	q6: critical section
p7: wantp ← false	q7: wantq ← false

   Discuta se as seguintes propriedades são satisfeitas pelo mesmo, justificando:
   i. Exclusão Mútua
   ii. Progresso
   iii. Não postergação

3. Suponha que o algoritmo concorrente abaixo, que tem situações em que insere elementos nas listas A e B, retira elementos destas listas, e também ***passa*** elementos de uma lista para outra de forma **atômica**. Ou seja, o elemento transferido é observável em A ou em B. A operação ***passa*** não permite que outro processo observe o estado em que o elemento foi retirado de uma lista e ainda não inserido na outra. Como as listas A e B são acessadas concorrentemente, este algoritmo usa semáforos para proteger o acesso às operações de inserção e retirada de cada lista. Uma estrutura agregando uma lista e um semáforo para cuidar de sua exclusão mútua é criada, como abaixo.

```
struct listaBlq {      // lista com bloqueio
    mutex : Semaforo   // para exclusão mútua
    conteudo : Lista de itens
}

listaBlq(){  // construtor
     return new listaBlq { new Semaforo(1), new Lista() }
}

void insere(l: listaBlq, x: item){
   ...
   // insere x em l.conteudo
   ...
}

item retira(l: listaBlq){
   ...
   // retira item de l.conteudo
   ...
   retorna o item
}

boleano passa(l1 listaBlq, l2 listaBlq) {

   // passa um elemento qualquer de l1 para l2 atomicamente.
   // Outras operações não podem perceber estado da lista l1
   // sem o elemento retirado se este ainda não estiver inserido em l2

}

procedimento procLista(l1 listaBlq, l2 listaBlq){
  Loop {
         …
         passa(l1,l2)
         …
  }
}

main() {
   la := new listaBlq()
   lb := new listaBlq()
   inicia processos que inserem itens em la e lb
   inicia processos que retiram itens de la e lb
   // ou seja, la e lb terão itens no decorrer do funcionamento

   inicia  procLista(la,lb)  como um processo novo
   inicia  procLista(lb,la)  como um processo novo
   // deseja-se que estes processos possam passar elementos de
   // la para lb e de lb para la conforme necessário,
   // concorrentemente
   espera processos acabarem
}
```

Pede-se:

Apresente os algoritmos ***insere, retira e passa***, fazendo controle de concorrência utilizando semáforos. Suponha que uma lista não tem limite máximo de itens. Caso considere que outros semáforos são necessários para manipular uma lista, declare-os adicionalmente na estrutura listaBlq. Caso considere necessário mudar os parâmetros de uma operação, modifique e explique a razão.

4. Suponha que, nos sistemas de arquivos abaixo, a estrutura de diretorio esta carregada em memória e se deseja abrir um arquivo e acessar seu quinto bloco de dados. Descreva: (1) que estruturas necessitam ser trazidas de disco para a memória quando o arquivo é aberto (se houver) e (2) depois disso, quantos acessos a disco são necessários para se chegar a seu quinto bloco, justificando.
   a. Em um sistema de alocação encadeada
   b. Em um sistema FAT
   c. Em um sistema de alocação indexada