

Disciplina de Sistemas Operacionais – Trabalho Prático – Simulador de sistema operacional

Data da Entrega: 18/12/2017, até as 23:59h.

Valor do trabalho: 100 pontos, sujeito à penalidade de 50 pontos por dia de atraso na entrega.

Nota: a entrega do trabalho vale como presença para as aulas dos dias 5 e 6/12/2017.

Especificação do trabalho:

- O trabalho deve ser realizado em grupo de um a três alunos.
- Como parte do trabalho, implemente um simulador de componentes de hardware, os quais operam independentemente, em paralelo [Prof. Johann, UFRGS]. É recomendado o uso de *threads* em uma linguagem como C ou Java para simular o paralelismo.
 - Os seguintes componentes devem ser modelados: Processador, Disco e Teclado. Cada um desses componentes deve ser implementado em um arquivo separado, contendo suas funções e seus dados.
 - O objetivo desse trabalho será a simulação de parte de um pequeno sistema operacional com funções mínimas. Em particular, você deve criar as *threads* que representam os componentes que funcionam independentemente: Processador, Disco e Teclado.
 - Todas as *threads* iniciam de mesmo modo, executando um laço sem fim e repetindo a mesma sequência de operações: imprime uma mensagem e espera um pequeno intervalo (função *sleep*). No processador, a mensagem é “executei uma instrução”, e a espera é pequena. No teclado, a mensagem é “usuário digitou algo”, e somente é emitida a cada vez que é digitado “ENTER” no terminal (use *getchar* do C ou similar), sendo nesse caso a espera desnecessária. Finalmente o disco imprime “dei uma volta”, e espera um tempo muito maior que o processador. Até esse ponto você deve ver que ao rodar o programa ele mostra as mensagens intercaladas das três *threads*, sendo impressas repetidamente de forma idêntica.
 - O processador também deverá ler um dado da memória a cada iteração do seu laço, imprimir o que leu, e avançar o endereço de leitura. Se o dado for um elemento de vetor, o que se faz é acessar o vetor em um determinado índice (endereço) e imprimir o elemento obtido em uma iteração. Se você armazenar o endereço em uma variável “PC” e iniciar a memória com o vetor, estará simulando um processador que busca instruções da memória sequencialmente.
 - O disco, apesar de poder funcionar independentemente dos outros componentes, não deve ficar fazendo operações sem ser requisitado. Ele deve aguardar um pedido vindo do processador. Para isso, ele terá uma função que pode ser chamada pelo processador, e um semáforo interno, inicializado com 0. O disco executa *down()* nesse semáforo como primeira instrução dentro do seu laço (a cada iteração), de modo que ficará bloqueado aguardando logo no início da iteração. A função disponibilizada para o processador, quando chamada, deve executar um *up()* nesse

semáforo, liberando o disco para dar uma volta – lembre que a espera do disco é grande, e nesse caso você pode imprimir uma mensagem antes e outra depois da espera para poder observar esse assincronismo quando roda o programa. Tendo realizado isso, você pode verificar o funcionamento fazendo com que o processador chame essa função de pedido de disco a cada vez que ele encontrar um determinado valor na memória. Assim se simula o funcionamento de uma instrução de acesso a disco.

- O trabalho deve ser entregue para o email (newtonsp.unioeste@gmail.com) com um arquivo zipado contendo:
 - Todo o código fonte correspondente ao simulador, com comentários descrevendo o propósito de cada função, procedimento, classe e método.
 - Arquivo README.txt com informações básicas sobre pré-requisitos do simulador (sistema operacional, versão da máquina virtual Java...), além de configurações necessárias e instruções para execução do programa. O arquivo deve permitir que um programador consiga instalar e executar seu simulador com autonomia.
 - Arquivo com nomes dos alunos que fizeram o trabalho e todas as referências utilizadas (livros, relatórios, artigos, sites).
- O simulador deve ser apresentado ao professor no dia 20/12/2017. A ordem de apresentação é inversa à ordem de entrega de trabalho (*deadline*: 18/12). Assim, o grupo que entregar o trabalho primeiro será o último a apresentar. A apresentação deve conter no mínimo dois itens, com uso opcional de slides:
 - Informações básicas do código implementado: linguagem utilizada, tecnologias consideradas, bibliotecas e recursos adicionais empregados, pré-requisitos necessários.
 - Execução do simulador em um computador/*laptop* trazido pelo grupo.
- Após a simulação, o professor fará perguntas ao grupo sobre o trabalho.