

Organização de Computadores – DEI/ISEP

Exame da parte prática – Época de Melhoria

19 de Julho de 2006 – 1h30m

Conselhos para o exame:

- Tentar não usar expressões `case` ou `if`
- Dividir a definição das funções em várias cláusulas
- Usar *pattern matching* e/ou *guards* para seleccionar uma cláusula da função
- Manter o corpo da cláusula pequeno e simples
- Usar sub-funções
- Ler as duas páginas do enunciado

Avisos:

- Nota mínima na parte prática: 40%
- Todas as perguntas valem 1/3 da nota da parte prática
- Resolver cada exercício numa folha (**folha!**) separada devidamente identificada
- O teste deve ser resolvido individualmente
- É permitida a consulta de quaisquer documentos em papel que pertençam ao próprio
- Não é permitido o uso de computadores

Perguntas:

1. Faça uma função chamada `f texto/1` que aceite como argumento um texto (uma string) e formate esse texto do modo que está especificado na tabela seguinte:

Entrada:	Saída:
,	&
;	/DIV
um ou mais espaços consecutivos	apenas um espaço

2. Implemente um servidor que controla um braço robótico num armazém automático. O servidor deve ser capaz de tratar as seguintes mensagens:

- {colocar_peca,Pid,Linha,Coluna} – ordena ao braço robótico que coloque a peça na gaveta indicada por n^o da linha e coluna. Se a gaveta estiver ocupada, deve devolver uma mensagem de erro ao processo que requisitou a colocação da peça. Caso contrário, devolve uma mensagem de sucesso.
- {retira_peca,Pid,Linha,Coluna} – ordena ao braço robótico que retire a peça que se encontra na gaveta indicada por n^o de linha e coluna. Se não existir nenhuma peça nessa gaveta, devolve uma mensagem de erro. Se a peça existir, retira-a e avisa o processo do sucesso da operação.

Considere que o braço robótico possui um processo registado com o nome `braco`. Inicialmente o armazém está completamente vazio. Não importa o n^o de gavetas do armazém. Apenas tem de controlar os pedidos de colocação e retirada de peças das gavetas. Considere também que os processos clientes apenas enviam valores de `Linha` e `Coluna` válidos para o armazém em causa.

3. Simule o uso de memória partilhada para comunicação entre processos Erlang. Implemente um servidor que gere a criação e utilização de zonas de memória partilhada.

As zonas de memória partilhada são representadas por uma lista. O tamanho dessa lista é enviado no pedido de criação, juntamente com uma chave que identifica univocamente essa zona. A criação e partilha da chave é da responsabilidade dos clientes.

Os restantes processos acedem à zona de memória partilhada, indicando a chave e posição na lista a ler/escrever.

Sempre que um processo quer escrever um valor numa dada posição da memória partilhada, deve primeiro conseguir obter o semáforo que lhe garante exclusão mútua no acesso. O semáforo é atribuído pelo servidor, de acordo com a ordem de chegada dos pedidos. Após a escrita do valor, o processo deve libertar o semáforo. Existe apenas um semáforo por cada zona de memória partilhada.

As leituras não necessitam de exclusão mútua no acesso. O servidor devolve ao processo o valor que se encontra actualmente na posição da lista requerida.

Para tornar o sistema robusto é necessário considerar que se um processo terminar enquanto detém o semáforo que garante acesso exclusivo a uma dada zona, o servidor deve libertar esse semáforo de exclusão mútua e atribuí-lo ao próximo processo na fila de espera (se existir).

Implemente apenas o servidor. Assuma que os processos não enviam endereços inválidos ao servidor.