UFRGS - INF01151 - Sistemas Operacionais II N – Prof. Marcelo Johann – 27/03/2012

Tarefas para aula em modalidade de ensino à distância

Tema: Sincronização com Threads, Processos e Semáforos
Metodologia: programação em C POSIX
Recursos: livro-texto da série didática, slides na página da disciplina, consultas à Internet

Objetivos: As atividades dessa sexta aula da disciplina, não presencial, são exercícios com o uso de threads, processos, comunicação por memória compartilhada e uso de semáforos para resolução de problemas de sincronização.

Introdução: Você deve entregar as duas atividades listadas abaixo até o próximo sábado, em um arquivo zipado com o seu nome, enviado via e-mail para johann@inf.ufrgs.br, com o *subject* AULA06SISOP2. Exercícios adicionais são listados após, e podem ser explorados de forma parcial ou completa no andamento da disciplina e da avaliação.

Atividade 1:

Você deve recuperar o arquivo chamado *pncnversions.zip*, que contém 4 versões de um programa simples que executa vários produtores e vários consumidores operando sobre um buffer circular limitado. Inicie pelo programa *pncnpersistent.c* – implementação com threads e semáforos POSIX persistentes. Há uma versão chamada *pncnunnamed.c* – não testada, com semáforos sem nome, pode ser ignorada. A outra versão importante é *pncnfork.c* – cria processos separados, e não apenas *threads*, utilizando memória compartilhada. Há também uma versão em Java, que não é importante para essa tarefa.

Nessa atividade você deve estudar o código de *pncnpersistent.c* e *pncnfork.c*, compilar, rodar, e observar o seu funcionamento. Elabore um resumo descrevendo o seu comportamento, o que faz, quais são os problemas de sincronização, como foram resolvidos, e principalmente o que você aprendeu com a experiência, tirando conclusões críticas. Descreva também quais as dificuldades que teve ao entender os exemplos ou dúvidas que teria ao aplicar esses recursos a outro problema. Esse relatório deve ter cerca de uma página.

Atividade 2:

Escolher um dos problemas de sincronismo descritos no "*The Little Book of Semaphores*", de Allen Downey (http://greenteapress.com/semaphores/) e programá-lo em C utilizando threads e semáforos nomeados POSIX.

Exercícios Adicionais:

- Resolver o problema de sincronismo de "*race0.c*" com processos (via *fork*) e semáforos nomeados;
- Testar o uso de semáforos sem nome para algum problema de sincronismo;
- Programar em C a solução final para o problema dos leitores e escritores apresentada no livro de Toscani, com prioridade absoluta para os escritores;

- Fazer um programa do tipo produtor/consumidor, para um único produtor e vários consumidores, usando um "*pipe*" não nomeado para envio dos "itens", e verificar a necessidade (ou não) de sincronismo;
- Fazer uma versão para a solução de produtores e consumidores com vários processos, mas sem criá-los via *fork*. Cada processo deve ser disparado separadamente por uma janela de console. Escolha entre duas opções: códigos fonte separados, ou um mesmo código fonte que diferencia sua função por parâmetro passado na chamada;
- Experimentar o uso de "*pipes*" com nomes;