



Fundamentos de Sistemas Operacionais

Trabalho 05

Prof. Tiago Alves

Processos e Threads: Sincronização

Introdução

A disciplina de Fundamentos de Sistemas Operacionais trata de diversos tópicos desses sistemas que provêm uma forma intuitiva de se utilizar as funcionalidades de computadores digitais sem que seja necessário ao usuário ou programador ter profundo conhecimento das interações entre os diferentes *hardwares* que compõem um computador.

Para construir ou adicionar funcionalidades a esses sistemas computacionais, é necessário conhecimento de linguagens de programação e ferramentas de desenvolvimento. Em nosso curso, o domínio da linguagem C é um pré-requisito para o devido acompanhamento das atividades da disciplina.

Objetivos

- 1) Exercitar conceitos da linguagem de programação C, especialmente aqueles referentes à programação de sistemas operacionais.
- 2) Exercitar aspectos de programação de sistemas operacionais referentes ao gerenciamento de threads e processos.

Referências Teóricas

Mitchell, Mark, Jeffrey Oldham, and Alex Samuel. Advanced linux programming. New Riders, 2001.

Material Necessário

- Computador com sistema operacional programável
- Ferramentas de desenvolvimento GNU/Linux ou similares: compilador GCC, depurador, editor de texto.



Roteiro

- 1) Revisão de técnicas e ferramentas de desenvolvimento usando linguagem C.

Colete o material acompanhante do roteiro do trabalho a partir do Moodle da disciplina e estude os princípios e técnicas de desenvolvimento de aplicações usando linguagem C e sistema operacional Linux.

- 2) Realizar as implementações solicitadas no questionário do trabalho.

Nas referências bibliográficas que acompanham o trabalho, há uma breve apresentação de ferramentas de depuração para o ambiente Linux.

ATENÇÃO: Suas implementações deverão ser construídas a partir de um Makefile.

Implementações e Questões para Estudo

- 1) Escreva um programa em C que implemente uma aplicação com o seguinte comportamento:
 - Receba como entrada da linha de comando um inteiro que indique a quantidade de threads a serem lançadas pela aplicação. A aplicação não deverá permitir o disparo de mais de 10 threads.
 - Cada thread receberá como parâmetro o seu número (1 a 10). O trabalho da thread **1** será **imprimir uma linha** (em stdout) com um caractere **a**; o trabalho da thread **2** será **imprimir uma linha** com **dois** caracteres **b**; ...; o trabalho da thread **10** será **imprimir uma linha** com **dez** caracteres **j**. Cada thread deverá contabilizar a quantidade de linhas impressas durante sua execução e **cada thread deverá aguardar**, no mínimo, **500 ms** entre a impressões em tela.
 - O programa deverá tratar a recepção do sinal SIGINT (Ctrl+C) da seguinte maneira: a ocorrência de SIGINT deverá sinalizar que as threads lançadas devem ser encerradas e o programa ser fechado.
 - Após a recepção do sinal SIGINT, a aplicação deverá imprimir em tela uma mensagem (“Encerrando a aplicação. Aguardando finalização de threads....”).
 - Imediatamente antes de encerrar, a aplicação deverá imprimir em tela uma mensagem que indique o encerramento da mesma e a quantidade de linhas impressas por cada thread (“Aplicação encerrada com sucesso!\n Estatísticas: thread 1: XX linhas\n thread 2: YY linhas\n....”).
 - **Uma thread jamais interromperá a outra thread no momento da impressão em tela.**

Instruções e Recomendações

A submissão das respostas aos problemas dos trabalhos deverá ser feita através do Moodle da disciplina.

Cada Problema do Trabalho 05 deverá ser entregue em um pacote ZIP. A dupla de alunos deverá nomear o pacote ZIP da seguinte forma: nome_sobrenome_matricula_nome_sobrenome_matricula_**trab05.zip**.

Entre os artefatos esperados, listam-se:



- códigos-fonte C das soluções dos problemas;
- documentação mínima da aplicação:
 - o qual sistema operacional foi usado na construção do sistema;
 - o qual ambiente de desenvolvimento foi usado;
 - o quais são as telas (instruções de uso)
 - o quais são as limitações conhecidas

Não devem ser submetidos executáveis.

Códigos-fonte C com erros de compilação serão desconsiderados (anulados).

Os trabalhos poderão ser realizados em duplas; a identificação de cópia ou plágio irá provocar anulação de todos os artefatos em recorrência.