



Fundamentos de Sistemas Operacionais

Trabalho 03

Prof. Tiago Alves

Processos e Threads

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

Capítulos 3, 4 e 5 de 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 lance apenas um processo porém, para sua devida operação, o programa deverá disparar threads.
 - A aplicação receberá como **parâmetro de entrada** o nome do arquivo em que serão gravadas as mensagens para registro (log) em persistência.
 - Uma das threads, a **thread de controle**, será responsável por monitorar as entradas do usuário.
 - Quando o usuário digitar CTRL+C na linha de comando, a aplicação deverá ser finalizada elegantemente, avisando para o usuário que a thread de controle aguarda as outras threads para finalizar a aplicação. Esperam-se as seguintes mensagens quando da sinalização de CTRL+C:
 - [aviso]: Termino solicitado. Aguardando threads...
 - [aviso]: Maior numero gerado: 9999221
 - [aviso]: Menor numero gerado: -233333
 - [aviso]: Maior ocupacao de buffer: 13
 - [aviso]: Aplicacao encerrada.
 - O conteúdo das linhas 2, 3 e 4 da sequência de mensagens acima será especificado mais abaixo.
 - Todas as mensagens deverão também ser escritas no arquivo de log cujo nome foi informado como parâmetro de entrada da aplicação.
 - Outra thread, a **thread produtora**, será responsável por produzir informações (um inteiro aleatório) e escrever essa informação em um buffer limitado a 50 posições (array para 50 inteiros).
 - A thread produtora dorme 100 ms entre a produção de cada inteiro.
 - A thread produtora deverá registrar no arquivo de log qual foi o valor gerado. Formato esperado da mensagem:
 - [producao]: Numero gerado: -12345678
 - [producao]: Numero gerado: 98765432
 - Duas outras threads, as **threads consumidoras**, são responsáveis por ler as informações do buffer, 1 inteiro por vez.
 - Cada thread consumidora deverá dormir por 150 ms entre as leituras dos inteiros.
 - A thread consumidora deverá registrar no arquivo de log qual foi o valor lido do buffer. Formato esperado da mensagem:



- [consumo b]: Numero lido: -12345678
- [consumo a]: Numero lido: 98765432
- Durante seu processamento, cada thread consumidora deverá calcular qual foi o maior e o menor número inteiro aleatório lido do buffer.
- A thread de controle, durante a finalização da aplicação, deverá aguardar as threads consumidoras e produtoras se encerrarem.
 - Durante a finalização da aplicação, a thread de controle deverá informar em tela e escrever no arquivo de log:
 - qual foi o maior e o menor número inteiro gerado durante a execução da aplicação;
 - qual foi a maior ocupação de buffer, ou seja, quantos inteiros foram gerados e armazenados para aguardar pelo consumo.

Instruções e Recomendações

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

Cada Problema do Trabalho 03 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_trab03.zip.

Entre os artefatos esperados, listam-se:

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