Fundamentos de Sistemas Operacionais
Trabalho 02

Prof. Tiago Alves

Comunicação Entre Processos - IPC

Introdução

A disciplina de Fundamentos de Sistemas Operacionais trata de diversos tópicos desses sistemas que provêem 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 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 em que um processo pai crie dois processos filhos.
    - A arquitetura da sua aplicação deverá contemplar os seguintes requisitos:
        - O pai deverá conectar um pipe a cada um de seus processos filhos. Cada processo filho deverá escrever em seu próprio pipe e o processo pai deverá ler de todos os pipes que o ligam aos seus filhos.
    - Um dos filhos será o preguiçoso:
        - Ele gerará mensagens com informações de tempo (timestamps) com precisão de milésimos de segundos.
        - As mensagens deverão seguir o seguinte formato:
            - `0:00.123: Mensagem 01 do filho dorminhoco`
            - `0:02.456: Mensagem 02 do filho dorminhoco`
        - Após o envio da mensagem, o filho deverá dormir por um intervalo de tempo aleatório (0, 1 ou 2 segundos) antes de enviar uma nova mensagem.
    - O processo filho ativo deverá ler as entrada digitadas pelo usuário na linha de comando.
        - Cada mensagem lida deverá ser informada ao processo pai da seguinte forma:
            - `0:00.120: Mensagem 01 do usuario: <ababab>`
            - `0:05.388: Mensagem 02 do usuario: <alo mundo>`
        - Note o requisito de imprimir um timestamp na mensagem.
    - O processo pai controlará os filhos da seguinte maneira:
        - O processo pai monitorará a chegada de mensagens através dos pipes. Pesquise sobre a chamada de sistema `select()`, que poderá ser útil na atividade de monitoramento.
        - O processo pai lerá as mensagens dos pipes e as escreverá no arquivo `output.txt` na ordem em que foram lidas dos pipes. Ele deverá adicionar a timestamp do momento em que a mensagem foi lida do pipe.
            - As mensagens do arquivo `output.txt` possuirão duas timestamps: uma do processo pai e outra do processo filho.
            - Espera-se que o arquivo `output.txt` contenha uma mistura de linhas dos processos filhos.
                - `0:02.123: 0:00.120: Mensagem 01 do usuario:`

```
<ababab>
```

    - `0:03.233: 0:00.123: Mensagem 01 do filho dorminhoco`
- O processo pai deverá terminar a execução da aplicação após 30 segundos desde o início de sua execução.
  - O processo pai deverá encerrar sua execução apenas após seus processo filhos finalizarem sua execução.
- Recomendações
  - Atenda exatamente o formato em que as mensagens serão trocadas entre os processos e como serão gravadas em arquivo. Note que o processo pai deverá incluir seu timestamp próprio como cabeçalho da mensagem lida do processo filho e que será escrita no arquivo de saída.
  - Respeite o nome do arquivo de saída.
- Sugestões de preparação e de execução da implementação
  - Estude como gerar números inteiros aleatórios usando o sistema operacional Linux e monte uma função capaz de escolher entre os inteiros 0, 1 e 2 de forma equiprovável. Cada um dos inteiros terá 33,3% chances de ser escolhido.
  - Estude como instruir um processo a dormir.
  - Estude como obter informações de tempo com precisão de milisegundos, precisão necessária na construção dos cabeçalhos das mensagens formatadas.
  - Estude como usar pipes para trocar informação entre processos relacionados.
  - Monte um esquema que permita a criação de processos filhos diferenciados, ou seja, permita que o pai informe aos filhos qual é o seu tipo: dorminhoco ou ativo.
  - Estude como gerar saídas em arquivos.
  - **Implemente** uma versão preliminar do processo pai que seja capaz de encerrar depois de 30 segundos de execução.
  - **Implemente** o processo filho dorminhoco capaz de trocar as mensagens com o processo pai.
  - **Implemente** o processo filho ativo capaz de trocar as mensagens com o processo pai.
  - **Adicione** a capacidade de gravação de mensagens formatadas em persistência ao processo pai.
  - Realize a integração dos processos.

Instruções e Recomendações

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

Cada Problema do Trabalho 02 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_**trab02.**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.