



Fundamentos de Redes de Computadores

Trabalho 01

Prof. Tiago Alves

Camada de Aplicação: Servidor Aritmético

Introdução

A disciplina de Fundamentos de Redes de Computadores trata de diversos tópicos relativos a Sistemas Operacionais, 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.

Objetivos

- 1) Exercitar conceitos de programação de sistemas operacionais.
- 2) Estudar a implementação de esquemas de comunicações entre processos (IPC), especificamente a utilização da interface *socket*.

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 de aplicações para o sistema operacional Linux.

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

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



Implementações e Questões para Estudo

- 1) Escreva um programa com duas seções: um **cliente** e um **servidor aritmético**.
 - O **cliente** receberá entradas do teclado e traduzirá as strings digitadas em requisições ao servidor aritmético, por exemplo, receber uma string como “3 + 4” e transformá-la em uma requisição ao servidor aritmético.
 - Formato do pacote de requisição: cada pacote consistirá de três campos: (i) uma string de três caracteres representando a operação matemática a ser executada, ou seja, “add”, “sub”, “div” e “mul”; (ii) o primeiro **inteiro** da operação; (iii) o segundo **inteiro**.
 - Note que o tamanho de cada pacote de requisição dependerá da representação de caracteres e inteiros da linguagem de programação usada nessa atividade. Em C, um caractere é representado por um byte ao passo que um inteiro é representando por 4 bytes.
 - O **servidor**, ao receber um pacote do cliente, deverá interpretá-lo, extrair os parâmetros da requisição, executar a operação solicitada, criar uma mensagem de resposta e transmitir os resultados ao cliente que, por sua vez, imprimirá os resultados na tela.
 - O servidor aritmético deverá implementar as funções: **add**, **subtract**, **multiply** e **divide**.
 - Cada um dessas funções deverá aceitar **dois inteiros** como entrada e retornar **um inteiro** como resultado. Naturalmente, exceções deverão ser tratadas adequadamente: a função **divide** deverá detectar eventual divisão por 0 e retornar um código de retorno que sinalize essa condição.
 - Cada pacote de resultado deverá conter dois campos: o **primeiro campo** é um inteiro que poderá assumir os valores **0** ou **1**, onde **0 representa uma operação bem sucedida** e **1 representa operação com erro** (divisão por 0, por exemplo). O **segundo campo** é o inteiro que conterá o resultado da operação aritmética ou o código de retorno especificando as condições de erro (você deverá criar uma sinalização para o caso divisão por 0).
 - Espera-se uma implementação do par cliente/servidor usando TCP como protocolo de camada de transporte.
 - São esperados os **códigos-fonte** dos projetos do cliente, do servidor e uma **documentação** que mostra como construir e usar o par cliente/servidor.
 - Na documentação, esperam-se também saídas que ilustram o programa operando corretamente.
 - **PONTUAÇÃO EXTRA(+10%)**: Implementar o servidor de forma que seja capaz de atender requisições concorrentes, ou seja, mais de um cliente demandando operações do mesmo servidor.

Instruções e Recomendações

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

Cada resposta a problema desse Trabalho **01** 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_**trab01.zip**.

Entre os artefatos esperados, listam-se:



- códigos-fonte das soluções dos problemas, correspondendo a **80% dos escores de pontuação final** do trabalho;
- documentação mínima da aplicação, correspondendo a **20% dos escores de pontuação final** do trabalho:
 - o qual sistema operacional foi usado na construção do sistema;
 - o qual ambiente de desenvolvimento foi usado;
 - o como construir a aplicação;
 - o como executar a aplicação;
 - o quais são as telas (instruções de uso);
 - o quais são as limitações conhecidas

Não devem ser submetidos executáveis.

Códigos-fonte com erros de compilação/interpretação 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.

Em relação ao uso das linguagens de programação, as equipes deverão atentar-se para a seguinte tabela de limites de escore:

- linguagem compilada C: **100%** de escores da parcela de implementação;
- linguagem compilada C++: **95%** de escores da parcela de implementação;
- linguagem Java: **90%** de escores da parcela de implementação;
- linguagens interpretadas: **90%** de escores da parcela de implementação.