

# REDES DE COMPUTADORES A

## Projeto 1

### Desenvolvimento do Projeto

O projeto deverá ser desenvolvido em **grupos de no máximo 4 integrantes**.

### Normas de conduta

A interação entre os grupos é estimulada, no entanto qualquer tentativa de plágio de trabalhos será punida com a **nota -Nmax nos trabalhos para todos os envolvidos**.

### Descrição da Projeto

**Objetivo:** Obter conhecimentos básicos para o desenvolvimento de software cliente e servidor em ambientes TCP/IP.

O objetivo deste projeto é implementar um cliente e um servidor para um Protocolo Simples para Transferência de Arquivos (PSTA). O Cliente fornece uma interface na forma de linha de comandos que permite ao usuário:

- Conectar-se a um servidor;
- Listar os arquivos localizados no diretório corrente do servidor;
- Receber um arquivo do servidor;
- Enviar um arquivo para o servidor;
- Encerrar a conexão com o servidor.

O servidor espera por conexões de clientes em uma porta conhecida. Após um cliente conectar-se ao servidor, o servidor espera por comandos. Quando o cliente enviar uma mensagem de encerramento de conexão, o servidor terminará a conexão. O servidor descrito trata-se de um servidor concorrente, ou seja, ele deve ser capaz de atender a diversos clientes simultaneamente.

O projeto consistirá na implementação de dois programas independentes em C que executem em sistemas operacionais Linux, um cliente chamado *cliente\_psta* e um servidor chamado *servidor\_psta*. Ao ser executado o programa cliente exibe um prompt, que permite ao usuário digitar os seguintes comandos:

- conectar <nome do servidor> [<porta do servidor>]
- listar
- receber <arquivo remoto> [<arquivo local>]
- enviar <arquivo local> [<arquivo remoto>]
- encerrar

Todos os comandos digitados pelo usuário serão sequências de caracteres ASCII. Os comandos enviados pelo cliente ao servidor também devem estar na forma de sequências de caracteres ASCII. Ao receber um comando, o servidor deve analisá-lo e executar a ação apropriada, conforme descrito a seguir:

- **conectar <nome do servidor> [<porta do servidor>]**

Este comando permite ao cliente conectar-se a um servidor. O nome do servidor poderá ser dado na forma literal (p.ex. `maquina1.puc-campinas.edu.br`) ou na forma de um endereço IP em notação decimal pontuada (p. ex. `200.136.254.139`). Ao receber este comando, o cliente deve iniciar uma conexão com o servidor.

- **listar**

Quando este comando for enviado ao servidor, este retornará a lista de arquivos no diretório onde ele está sendo executado. A lista será enviada como uma sequência de caracteres ASCII terminada por um delimitador de linha (`\n`). O cliente deverá receber a lista e exibi-la na tela. Dica: veja as páginas do manual para `getcwd`, `opendir`, `readdir`, `closedir`.

- **receber <arquivo remoto> [<arquivo local>]**

Permite ao cliente solicitar ao servidor o arquivo de nome <arquivo remoto>. Este arquivo deverá ser sempre tratado como um arquivo binário. Ao receber o arquivo especificado o cliente deverá salvá-lo no diretório onde está executando com o nome <arquivo local> ou, caso este nome não tenha sido especificado, com o nome <arquivo remoto>.

- **enviar <arquivo local> [<arquivo remoto>]**

Este comando é simétrico ao comando receber. Ele permite o envio de um arquivo local ao cliente para o diretório de trabalho do servidor.

- **encerrar**

Ao receber este comando, o cliente deverá enviá-lo para o servidor e terminar a conexão. Quando o servidor receber este comando ele deverá fechar o seu lado da conexão.

Para implementar a comunicação entre o cliente e o servidor use dois soquetes TCP: um de controle e outro de dados. O soquete de controle será usado para o envio de mensagens de controle (comandos enviados pelo cliente). O soquete de dados será usado para o envio e recepção de mensagens de dados. Neste esquema, o programa *servidor\_psta* age como servidor na porta de controle e o programa *cliente\_psta* age como servidor na porta de dados. A conexão de dados é estabelecida e desfeita para cada comando.

## Detalhes da entrega

Cada grupo deverá entregar o **código fonte de todos os programas implementados** e um **relatório em formato digital (.pdf)** contendo:

- detalhes do ambiente utilizado no desenvolvimento da atividade (contendo fotos do ambiente);
- detalhes de projeto e de implementação da aplicação, apresentando uma descrição de alto nível dos programas implementados (contendo textos explicativos e diagramas) e o formato das requisições e respostas enviadas;
- descrição do processo de compilação (contendo um screenshot que mostre os comandos utilizados na compilação e os resultados obtidos);
- descrição dos testes realizados (contendo screenshots da execução dos programas e os resultados obtidos, demonstrando que o programa realiza as tarefas solicitadas);
- os códigos fonte dos programas implementados.

Além disso, cada grupo deve preparar um **vídeo de no máximo 2 min** demonstrando o funcionamento do seu sistema e **uma apresentação de no máximo 15 min** descrevendo os detalhes do trabalho, que será realizada nas aulas seguintes à entrega.