

# REDES DE COMPUTADORES A

## Atividade 3

### Desenvolvimento da Atividade

A atividade deverá ser desenvolvida em **grupos de no máximo 4 integrantes**.

Em todos os laboratórios o Sistema Operacional Linux encontra-se instalado. Faça uso de qualquer um dos laboratórios para a realização dos projetos.

### 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 Atividade

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

Leia atentamente as informações a seguir e realize as atividades solicitadas:

- Estude os textos Projeto de Software de Clientes e Projeto e Implementação de Servidores;
- Estude, compile e teste os exemplos de programas cliente TCP e servidor TCP com sleep;
- Estude, compile e teste os exemplos de programas cliente TCP e servidor TCP concorrente (com fork);
- Implemente uma aplicação para compartilhamento de mensagens, como descrito a seguir;
- Elabore um relatório detalhado mostrando todo o trabalho realizado, como descrito a seguir.

#### Parte 1

Na primeira parte desta atividade você deve executar o programa servidor TCP com sleep e pelo menos 5 (cinco) instâncias simultâneas do programa cliente TCP básico.

Observe e registre os resultados obtidos no relatório, procurando explicar o que está ocorrendo.

Procure aumentar o número de instâncias simultâneas do programa cliente TCP básico para 20 (vinte). Observe e registre os resultados obtidos no relatório, procurando explicar o que está ocorrendo.

#### Parte 2

Na segunda parte desta atividade você deve executar o programa servidor TCP concorrente (com fork) e pelo menos 5 (cinco) instâncias simultâneas do programa cliente TCP básico.

Observe e registre os resultados obtidos no relatório, procurando explicar o que está ocorrendo.

Dica: Utilize o comando `ps -axf` para visualizar a árvore de processos existentes.

Observe que alguns processos ficam em estado "zumbi" (indicados pela flag de estado "Z" exibida pelo comando `ps` e a mensagem <defunct> na frente do nome do processo).

Mostre como é possível evitar que esses processos fiquem no estado zumbi.

#### Parte 3

Nesta parte serão implementados um cliente TCP e um servidor TCP concorrente que permitirão o compartilhamento de mensagens utilizando sockets TCP.

Ao executar o programa servidor o usuário deve informar o número da porta na qual o programa servidor aguardará por requisições, como no exemplo a seguir:

```
# ./servidor 5000
```

O servidor deve aguardar por requisições enviadas pelos clientes e realizar as operações solicitadas, exibindo na tela mensagens de log contendo informações resumidas das operações solicitadas. As mensagens de log devem exibir informações como o endereço IP e a porta do cliente que solicitou a operação, além da operação solicitada e o resultado retornado pelo servidor informando sucesso ou falha na realização da operação.

Ao executar o programa cliente o usuário deve informar o nome do computador (ou o endereço IP) e a porta onde se encontra o programa servidor, como no exemplo a seguir:

```
# ./cliente localhost 5000
```

ou

```
# ./cliente 192.168.1.2 5000
```

Em seguida, o programa cliente deve exibir um menu com as seguintes opções:

**Opções:**

- 1 - Cadastrar mensagem
- 2 - Ler mensagens
- 3 - Apagar mensagens
- 4 - Sair da Aplicação

O programa cliente só deve ser encerrado, caso o usuário selecione a opção 4.

Caso o usuário selecione a opção 1, o programa cliente deve solicitar o nome do usuário e a mensagem que será cadastrada, como no exemplo a seguir:

**Usuário:** fulano

**Mensagem:** Testando o cadastramento de mensagens.

O nome do usuário deve conter no máximo 19 caracteres.

A mensagem digitada deve conter no máximo 79 caracteres.

Após a leitura do nome do usuário e da mensagem, o programa cliente deve enviar uma requisição ao servidor solicitando o cadastramento de tais informações.

O servidor deve então realizar a operação solicitada, retornando ao cliente uma resposta informando se a solicitação foi atendida corretamente ou não, especificando o motivo da falha caso a requisição não possa ser atendida por algum motivo.

O servidor deve permitir o cadastramento de no máximo 10 mensagens no total.

Caso o usuário selecione a opção 2, o programa cliente deve enviar uma requisição ao servidor solicitando o envio das mensagens existentes.

O servidor deve então realizar a operação solicitada, retornando ao cliente uma resposta informando quantas mensagens cadastradas existem e, caso existam mensagens cadastradas, uma listagem contendo o nome do usuário e a respectiva mensagem.

O programa cliente, após receber a resposta do servidor, deve exibir uma mensagem adequada na tela contendo as informações recebidas, como no exemplo a seguir:

**Mensagens cadastradas: 2**

**Usuário:** fulano

**Mensagem:** Testando o cadastramento de mensagens.

**Usuário:** ciclano

**Mensagem:** Compartilhamento de mensagens funcionando!

Caso o usuário selecione a opção 3, o programa cliente deve solicitar o nome do usuário para o qual as mensagens serão apagadas no servidor:

**Usuário:** fulano

Após obter o nome do usuário, o cliente deve enviar uma requisição adequada ao servidor, que deve então realizar a operação solicitada, retornando ao cliente uma resposta informando quantas mensagens foram apagadas e uma listagem contendo o nome de usuário e as mensagens apagadas, caso existam.

O programa cliente, após receber a resposta do servidor, deve exibir uma mensagem adequada na tela contendo as informações recebidas, como no exemplo a seguir:

**Mensagens apagadas: 1**

**Usuário:** fulano

**Mensagem:** Testando o cadastramento de mensagens.

O servidor deve ser capaz de atender a vários clientes simultaneamente, através da criação de processos filhos para atender a cada novo cliente. O servidor também deve evitar a existência de processos zumbi.

Além disso, cada cliente deve enviar todas as requisições através de uma única conexão TCP (conexão persistente).

## Detalhes da entrega

Cada integrante do grupo deverá postar em seu escaninho no AVA, na pasta **Atividade3**, o **código fonte de todos os programas implementados** e um **relatório em formato digital (.pdf)** contendo:

- Parte 1: descrição dos testes realizados procurando explicar os resultados obtidos (contendo screenshots da execução dos programas e dos resultados obtidos);
- Parte 2: descrição dos testes realizados procurando explicar os resultados obtidos (contendo screenshots da execução dos programas e dos resultados obtidos);
- Parte 3:
  - 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.