

Trabalho 1 – Programação socket

1.1 Orientações gerais

- O trabalho deverá ser realizado em dupla.
- Trabalhos iguais a outros trabalhos terão nota zero.

1.2 Objetivos

Objetivo geral do trabalho é desenvolver uma aplicação cliente/servidor utilizando a biblioteca de programação *socket* na linguagem C.

Os objetivos específicos do trabalho são:

- Familiarizar-se com a programação utilizando a API *socket*.
- Implementar um protocolo simples para envio e recebimento de mensagens.
- Enviar e receber dados em uma aplicação que utiliza a arquitetura Cliente/Servidor.

1.3 Descrição do trabalho

Monitorar e controlar o acesso a determinados ambientes é fundamental. Explorando este contexto, será desenvolvido neste trabalho **uma aplicação composta de um servidor e diversos clientes** para monitorar a entrada e saída de pessoas de um ambiente conforme ilustrado na figura 1.

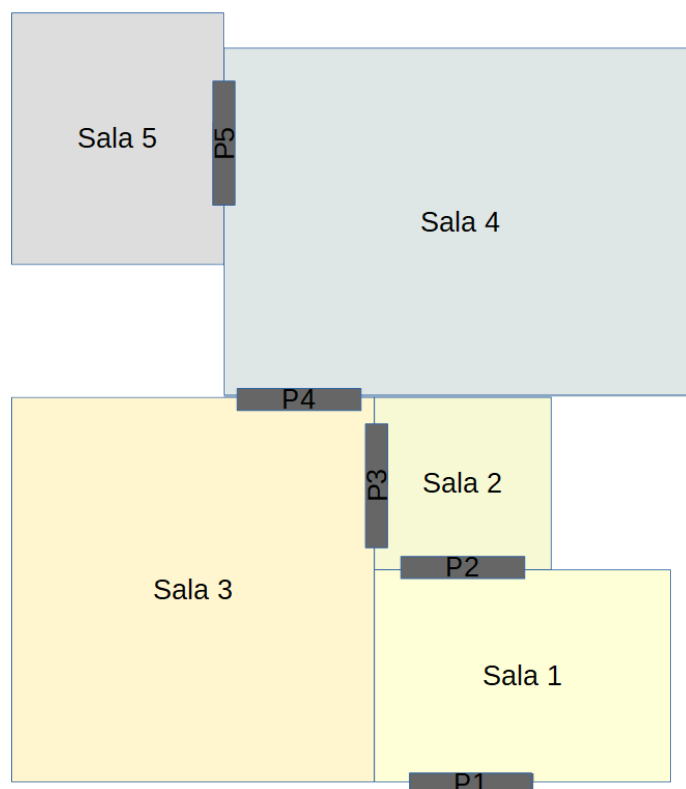


Figura 1: Ambiente a ser monitorado

1.1.1 Cliente

A aplicação cliente representará será o software instalado junto as portas. Sua função será:

1. Ler os dados fornecidos pelo usuário (pelo teclado);
2. Enviar estes dados para o servidor;
3. Receber as informações de autorização do servidor e;
4. Imprimir (na tela) o nome do usuário e se ele está autorizado ou não a passar pela porta.

1.1.2 Servidor

A aplicação servidor será o software instalado na central de controle do ambiente. Sua função será:

1. Armazenar as credências dos usuários em uma base de dados, conforme descrito na seção “Credenciais de usuários”;
2. Receber as requisições vindas dos clientes;
3. Consultar a base de dados de usuários;
4. Armazenar a tentativa de acesso um arquivo de registro, conforme descrito na seção “Arquivo de registro”;
5. Enviar para os clientes informações de autorização.

1.1.3 Credenciais de usuários

As credencias do usuário serão armazenadas em um arquivo TXT, onde cada linha conterà três informações: código do usuário, nome do usuário e nível de acesso. Por exemplo:

```
1458, Tony Stark, 5
4287, Steve Rogers, 4
4298, Thor, 2
6528, Natasha Romanoff, 3
7458, Bruce Banner, 1
```

Usuários com nível acesso 1 podem acessar, apenas, a sala 1. Usuários com nível de acesso 2 podem acessar as salas 1 e 2, e assim por diante, de forma que o usuário com nível de acesso 5 pode acessar todas as salas.

1.1.4 Arquivo de registro

O arquivo de registro de tentativas de acesso será um arquivo no formato TXT. Cada tentativa de acesso reportada pelos software clientes deverão ser registradas em uma linha deste arquivo. Cada linha do arquivo deve conter as seguintes informações: data e hora no formato “dd/mm/aaaa – hh:mm:ss”, identificação da porta, identificação do usuário, autorizado ou negado. Exemplo:

```
04/09/2018 – 13:20:14, p1, 1458, autorizado
04/09/2018 – 14:11:35, p2, 7458, negado
```

1.1.5 Protocolo

Funcionamento básico do protocolo:

- O servidor é inicializado.
- Os clientes são inicializados e ficam aguardando a digitação do código do usuário.
- Ao digitar um código de usuário, o software cliente:
 1. Cria uma mensagem.
 2. Conecta-se ao servidor.

3. Envia a mensagem para o servidor.
 4. Aguarda pela resposta de autorização do servidor. Caso o cliente não receba uma mensagem de autorização dentro de 10 segundos, ele reenviará a mensagem para o servidor.
 5. Verifica se o conteúdo do campo autorização.
 - 5.1 Se o valor for "A", imprime na tela a frase "Acesso autorizado. Bem-vindo Nome do Usuário".
 - 5.2 Se o valor for "N", imprime na tela a frase "Acesso negado. Nome do Usuário suas credencias não permitem acesso a esta sala".
 6. Desconecta-se do servidor.
 7. Aguarda a digitação de um novo código.
- O servidor, ao receber uma mensagem irá:
 1. Consultar a credencia informada na base de credenciais.
 2. Armazenar a tentativa de acesso no arquivo de registro.
 3. Enviar o mesmo pacote para o cliente, preenchendo o campo autorização com "A" ou "N", dependendo da porta que se deseja atravessar e do nível de acesso do usuário.

1.1.6 Mensagem

A mensagem será composta pelos seguintes campos:

- Identificação da porta: um número inteiro de 16 bits. A identificação da porta deve ser fornecida pelo usuário, via linha de comando, ao inicializar o cliente. Exemplo:

```
./cliente_porta 1
```

- Data e hora: utilizar o tipo "time_t" (ver exemplo no código apresentado aula). A data e hora deve ser obtida pelo sistema.
- Autorização. Este campo será do tipo "char". Apenas o servidor irá preencher este campo.
- Nome do usuário. Este campo será uma *string* com 50 caracteres. Apenas o servidor irá preencher este campo.

1.4 Entrega do trabalho

O trabalho deverá ser postado na sala da disciplina no Moodle, utilizando a tarefa "*Envio do trabalho 1 – Programação Socket*". É importante notar que a data limite para entrega dos trabalhos será: **Quinta-feira, 25 de setembro de 2018. As duplas que não enviarem o trabalho dentro do prazo receberão nota zero.**

Deverá ser enviado um único arquivo compactado, em formato "zip" contendo:

- Um documento no formato TXT apresentando: os autores trabalho e um minitutorial do programa desenvolvido.
 - O minitutorial deve detalhar, no mínimo:
 - Como compilar os arquivos.
 - Como executar o software cliente.

- Como executar o software servidor.
- Como visualizar o arquivo de registro.
- Os códigos fontes dos trabalhos.

1.5 Pontuação

O valor total do trabalho será de 100 pontos, distribuídos da seguinte forma:

- 75 pontos para o desenvolvimento dos códigos fontes.
- 05 pontos para o minitutorial.
- 20 pontos para apresentação do trabalho.

1.6 Apresentação do trabalho.

Cada grupo deverá agendar um horário para apresentar o trabalho. A apresentação será realizada fora do horário de aula. O grupo ficará responsável por mostrar o trabalho funcionando e esclarecer eventuais dúvidas durante a apresentação.