

**Projeto de pesquisa - Criando ambientes virtuais de conversação  
com uso system call select()**

**Universidade de Brasília**

**Curso: Engenharia de Software**

**Disciplina: Fundamentos de Redes de Computadores**

**Professor: Fernando W. Cruz**

**Período: 2023/1**

**Participantes:**

**Antonio Rangel Chaves - 180098021**

**Antonio Ferreira de Castro Neto - 190044799**

**Enzo Gabriel Guedes Queiroz Saraiva - 160119006**

**Brasília, 2023**

# Projeto de pesquisa - Criando ambientes virtuais de conversação com uso `system call select()`

## Introdução

Este projeto é uma solução de software que visa atender a necessidade de conversas entre pessoas remotamente, em salas de bate-papo. O projeto fornece algumas possibilidades para o bate-papo, como: A criação de salas, na qual o tamanho é personalizado pelo usuário, a possibilidade de sair da sala e entrar em outra, com a opção de listar quais são os participantes de cada sala e também a exclusão de salas. Para resolver tais problemas propostos, foi necessário usar `sockets TCP` e a `system call select` que são discutidos posteriormente.

## Metodologia utilizada

Durante a realização do projeto de pesquisa, adotamos uma metodologia de trabalho colaborativa para realizar a implementação e documentação. Inicialmente, consumimos os conteúdos disponibilizados na plataforma Moodle pelo professor, além de buscar informações adicionais na internet para aprofundar nosso conhecimento sobre os conceitos de Sockets TCP/IP e a System Call Select().

Pelo fato do grupo ser formado por três integrantes, dividimos os requisitos do projeto entre nós, atribuindo responsabilidades específicas a cada membro. Para garantir uma comunicação eficiente, utilizamos a plataforma Discord para realizar encontros virtuais e alinhar o progresso individual de cada um, além de trabalharmos juntos em momentos oportunos no desenvolvimento de alguns requisitos mais complexos do chat. Também foi utilizado um grupo no Whatsapp para contato rápido e prático.

Ao finalizar a etapa de implementação, elaboramos um relatório detalhado do projeto, além da organização dos slides da apresentação. Cada integrante ficou responsável por preparar e ensaiar sua parte da apresentação.

Através dessa metodologia de trabalho colaborativa e das ferramentas de comunicação utilizadas, projetamos desenvolver a pesquisa de forma eficiente visando atender todos os requisitos para um bom aprendizado.

## Descrição da solução

A solução desenvolvida tem como base 2 structs, a struct das salas e dos clientes. Essas structs são essenciais para o desenvolvimento das salas de chat, visto que vão definir a estrutura das salas e dos clientes.

A struct da sala tem 5 atributos: um objeto do tipo `fd_set` para definir o cesto de sockets referente a essa determinada sala, o limite de participantes da sala, a quantidade de participantes presentes na sala, um boolean para indicar se a sala está ativa ou não, e por fim um ponteiro apontando para os clientes da sala.

```
typedef struct {
    fd_set sala_fd;
    int limite;
    int quantidade_clientes;
    bool ativo;
    cliente *clientes;
} sala;
```

## Projeto de pesquisa - Criando ambientes virtuais de conversação com uso system call select()

A structs de cliente tem 3 atributos: um descritor de arquivo (socket) referente ao cliente, uma string representando o nome e um boolean indicando se está ativo ou não.

```
typedef struct {
    int cliente_sd;
    char nome[MAX_STR_SIZE];
    bool ativo;
} cliente;
```

Além das structs, temos 6 funções principais, que garantem o correto funcionamento do sistema:

- *void prepara\_servidor()*: Preparar o servidor para poder ser utilizado no programa, fazendo a limpeza dos cestos na master e em todas as salas de chat. É utilizado o macro FD\_ZERO() nessa função.
- *void entrar\_na\_sala(...)*: Essa função é utilizada para entrar numa das salas de chat. Nela é utilizado o macro FD\_SET() para inserir o socket no cesto, além de checar se a sala atingiu sua capacidade máxima. Também é enviado uma mensagem ao cliente usando o send(), indicando em qual sala ele se encontra. A função recebe como parâmetro: o descritor do arquivo socket, o id da sala o qual quer entrar, o nome do cliente e o tamanho do nome.
- *int cria\_sala(...)*: É responsável por criar uma nova sala de chat. É passado como parâmetro um número representando a quantidade máxima de clientes permitidos na sala e o socket o qual está criando a sala. Além disso, é enviado uma mensagem de volta ao socket que criou a sala informando o id da sala utilizando o send().
- *void lista\_salas()*: Essa função envia ao cliente uma lista das salas de chat ativas. Para cada sala ativa, é criada uma mensagem contendo o número da sala, seu status de atividade e a quantidade de clientes conectados em relação ao limite estabelecido. Essa mensagem é enviada ao cliente usando a função send().
- *void envia\_msg(...)*: É responsável por enviar uma mensagem para todos os clientes conectados em uma sala específica, exceto para o cliente que enviou a mensagem e o servidor. Para cada cliente na sala, a função constrói a mensagem e a envia utilizando a função send(). Isso permite que a mensagem seja distribuída para todos os outros participantes da sala. Recebe como parâmetro o socket que está enviando a mensagem, o socket do servidor, o id da sala e o id do cliente.
- *void sair\_da\_sala(...)*: A função em questão remove um cliente de uma sala de chat específica. Ela marca o cliente como inativo, reduz a contagem de clientes na sala e remove o descritor de arquivo do cliente dos conjuntos relevantes. Se a sala ficar vazia após a saída do cliente, ela é desativada. Além disso, utiliza o macro FD\_CLR() para retirar o socket do cesto. Recebe

## Projeto de pesquisa - Criando ambientes virtuais de conversação com uso system call select()

como parâmetro o socket que está saindo da sala, o id da sala e do cliente, além de um inteiro auxiliar para retirar da master.

Além dessas supracitadas, temos algumas outras funções auxiliares como a função de menu que vai permitir uma melhor navegabilidade para o usuário, uma função de listar participantes de uma determinada sala e entre outras funções como de validação de nome ou validação de entrada para conservar os limites máximos de sala e de tamanho do nome definidos no programa.

```
void prepara_servidor();
void sair_da_sala(int socket, sala salas[], cliente clientes[], int total_clientes);
void lista_salas();
int cria_sala(int limite, int socket);
void envia_msg(int socket_descritor_arquivo, int server_sd, int sala_id, int cliente_id);
void entrar_na_sala(int socket_descritor_arquivo, int sala_id, cliente cliente);
void menu(int socket, cliente cliente);
int validar_entrada(int sala, int socket);
void validar_nome(int socket, char *nome, int *tam_nome);
void desconectar_cliente(int socket, sala salas[], cliente clientes[], int total_clientes);
void lista_participantes_sala(int socket, int sala);
```

### Conclusão

A solução elaborada possibilita a criação de salas de chat em um ambiente virtual através do emprego da chamada de sistema select(), utilizando sockets. Além disso, é viável diversas funcionalidades como criar sala, listar salas, ingressar numa sala, sair da sala, enviar mensagens para os participantes da sala além de exibir uma lista dos usuários conectados à sala e facilitar a troca de clientes em cada sala. Os requisitos do trabalho foram praticamente completos, com algumas ressalvas que não foram realizadas, como a exclusão de salas, ou a validação de entrada no momento de criar ou listar salas, para sabermos se é um caractere ou número sendo inserido. Como incremento, foi implementado validações de nome, ao inserir o nome do cliente a ser conectado e um menu interativo para facilitar a navegabilidade

### Relatos da Equipe

#### Antonio Ferreira

Gostei do trabalho, e achei condizente com a matéria e é uma boa forma de entendermos como funciona um serviço de chat baseado em sockets, além de termos um vislumbre de como funcionam os grandes serviços de chat no mercado como o Whatsapp. Os exercícios ao longo do semestre me ajudaram muito a entender conceitos para esse projeto de pesquisa, além de colocar a mão na massa diretamente para entender como um servidor de chat opera “por trás dos panos”. A minha dificuldade foi mais voltada para o uso da linguagem C, mas ao mesmo tempo essa linguagem me mostrou as minúcias do funcionamento desse sistema

## **Projeto de pesquisa - Criando ambientes virtuais de conversação com uso `system call select()`**

além do uso correto do `select()`. Portanto, acredito que uma nota justa para mim seria 8,5.

### **Antonio Rangel**

Achei interessante usar a linguagem C para fazer uma conexão pela rede, é mais complicado que o python 3 por exemplo, mas é bom ter essa dificuldade para entender como funciona de fato o *select* e os *file descriptors*. Durante a implementação eu e os colegas tivemos que mudar alguns atributos das structs como o atributo sala na struct cliente, que armazena em qual sala está o determinado cliente, e isso resultou em refatoração. Tive que pesquisar sobre o funcionamento das funções `recv` e `select` para que eu pudesse fazer o uso correto. No geral foi um trabalho com um tema bem próximo ao cotidiano e que nos ensinou o conteúdo da disciplina.

### **Enzo**

No começo estava meio perdido quanto ao projeto, não sabendo como aplicar os conhecimentos, que haviam sido pouco práticos até o momento, e pelo longo tempo sem usar C, mas depois que engatei e consegui ser bastante produtivo, entender bastante sobre o funcionamento de funções como `recv` e `send`, que são muito úteis para programas sistemas de redes e por fim consegui ajudar muito a equipe. Gostei bastante de fazer o projeto pois aprendi coisas interessantes sobre sistemas de redes, sobre o uso do `select` também. Em resumo foi bem legal pois coloquei conhecimento na prática. Acredito que uma nota justa para mim seria 9.

## **Referências bibliográficas**

- Tedesco, Kennedy. Uma Introdução a TCP, UDP e Sockets. Disponível em: [https://www.treinaweb.com.br/blog/uma-introducao-a-tcp-udp-e-sockets#:~:text=O%20que%20%C3%A9%20um%20Socket,0.1%3A4477%20\(IPv4\).](https://www.treinaweb.com.br/blog/uma-introducao-a-tcp-udp-e-sockets#:~:text=O%20que%20%C3%A9%20um%20Socket,0.1%3A4477%20(IPv4).) Acesso em: 29/06/2023.
- Kerrisk, Michael - Linux manual page. Disponível em: <https://man7.org/linux/man-pages/man2/select.2.html>. Acesso em: 01 jul. 2023.