System call select():

- O *select()* é uma função disponível no linux que permite monitorar um conjunto de descritores de arquivos (*file descriptors*) em busca de atividades de leitura, escrita ou exceções.
- É comumente usada em programação de rede para aguardar por eventos em sockets ou outros tipos de descritores de arquivo.
- A função *select()* bloqueia o programa até que ocorra uma atividade em um ou mais dos descritores de arquivo monitorados ou até que o tempo limite seja atingido. Quando ocorre uma atividade nos descritores de arquivo, a função *select()* retorna o número de descritores de arquivo prontos para leitura, escrita ou exceção.
- Com esse dado você pode iterar sobre os descritores de arquivo para realizar operações relevantes, como leitura ou escrita de dados. Criando assim um servidor com multiplexação sem o uso de thread.

Estrutura de dados fd_set:

- O tipo *fd_set* é uma estrutura de dados que representa um conjunto(set) de descritores de arquivo e sua lib exportam as funções *FD_ZERO(), FD_SET(), FD_CLR() e FD_ISSET()* são usadas para manipular o conjunto de descritores de arquivo.
- *FD_ZERO():* Zerar o set de file descriptors.
- *FD_SET():* Adicionar um file descriptor ao set.
- *FD_CLR():* Remover um file descriptor do set
- *FD_ISSET():* Verificar se um file descriptor está contido no set.

Uso no client.c:

Neste código estamos monitorando os fds do cliente e sempre que estiverem dentro do set(*FD_ISSET()*) iremos tratar a troca de mensagens entre o server.

```
58
59  strncpy(client_name, argv[1], BUFSIZE);
60  connect_req(&sockfd, &server_addr, client_name);
61  FD_ZERO(&master);
62  FD_ZERO(&read_fds);
63  FD_SET(0, &master);
64  FD_SET(sockfd, &master);
65
66  last_fd = sockfd;
67
68  while (1) {
69    read_fds = master;
70
71    if (select(last_fd + 1, &read_fds, NULL, NULL, NULL) == -1) {
72      perror("Erro no select()");
73      exit(1);
74    }
75
76    for (i = 0; i <= last_fd; i++) {
77      if (FD_ISSET(i, &read_fds))
78        send_recv(i, sockfd);
79    }
80  }
```

Uso no client.c:

Neste código estamos tratando o fd *master*(server) e o *read_fds* para quando o mesmo for escrito o server seja notificado através do *select()*.

Depois de ser notificado da atualização no *read_fds* iremos iterar sobre todo o conjunto(set) buscando qual fd foi modificado e tratar de acordo: tratando um novo usuário ou indo para a troca de mensagens.

```
104   for (i = 0; i < FD_SETSIZE; i++)
105     client_names[i] = NULL;
106
107   FD_ZERO(&master);
108   FD_ZERO(&read_fds);
109   connect_request(&socket_fd, &my_addr);
110   FD_SET(socket_fd, &master);
111
112   last_fd = socket_fd;
113
114   while (1) {
115     read_fds = master;
116
117     if (select(last_fd + 1, &read_fds, NULL, NULL, NULL) == -1) {
118       perror("Erro no select()");
119       exit(4);
120     }
121
122     for (i = 0; i <= last_fd; i++) {
123       if (FD_ISSET(i, &read_fds)) {
124         if (i == socket_fd)
125           connection_accept(&master, &last_fd, socket_fd, &client_addr, client_names);
126         else
127           send_recv(i, &master, socket_fd, last_fd, recv_buf, client_names);
128       }
129     }
130   }
131
```