Introdução

Bom, inicialmente eu segui as vídeo aulas presentes no moodle para criar uma comunicação padrão. Como havia especificações (como não enviar o caracter \o, o servidor não precisar se conectar a mais de um cliente por vez,etc) eu acabei fazendo algumas pequenas modificações na comunicação padrão.

Após isso eu fiz a criação de uma função, func (), no serve, para deixar o código mais organizado.

Nessa função eu verificava o comando recebido pelo servidor, inclusive comandos inválidos, e direcionada a respectiva função, de acordo com o proposto pelo TP.

```
        }

        else if (strncmp(messageReceive, "remove", 6) == 0) {
            bzero(messageSend, BUFSZ);
            strcat(messageSend, remov(messageReceive));
            count = send(csock, messageSend, strlen(messageSend), 0);
            if (count != strlen(messageSend)) {
                logexit("send");
            }
            close(csock);

        } else if (strncmp(messageReceive, "list", 4) == 0) {
            bzero(messageSend, BUFSZ);
            strcat(messageSend, list(messageReceive));
            count = send(csock, messageSend, strlen(messageSend), 0);
            if (count != strlen(messageSend)) {
                logexit("send");
            }
            close(csock);

        } else if (strncmp(messageReceive, "read", 4) == 0) {
            bzero(messageSend, BUFSZ);
            strcat(messageSend, reader(messageReceive));
            count = send(csock, messageSend, strlen(messageSend), 0);
            if (count != strlen(messageSend)) {
                logexit("send");
            }
            close(csock);

        } else if (strncmp(messageReceive, "kill", 4) ==
                   0) { // if msg contains "kill" then server exit and chat
                        // ended.
            close(csock);
            break;
        } else {
            break;
        }
    }
}

```

Eu usei uma matriz 5x5 para evitar o índice 0. Essa matriz era global para poder ser acessada por todas as funções. Cada função recebia e retornava uma string de 500bits.

Um dos desafios enfrentados por mim inicialmente foi que eu não sabia que eu tinha que especificar um tamanho para a string de retorno das minhas funções, e isso me custou muito tempo até eu descobrir que tinha que realizar a alocação dinâmica, até então eu me deparava com segmentation fault.

Eu também tive muita dificuldade em construir a função reader por causa dos tratamentos envolvendo sensores existentes nos equipamentos e sensores não existentes.

Nas minhas funções eu usei a função strtok para quebrar meus parâmetros recebidos. Eu colocava esses tokens em outras variáveis para conseguir usar partes específicas da mensagem recebida.