

Relatório de APC — Simulador de Restaurante

Vitor Fernandes Dullens – 16/0148260

Explicação detalhada sobre as variáveis globais utilizadas:

char restaurante[17][41]: matriz que guarda o arquivo tipo texto(.txt) que representa o restaurante. Nela que são alterados os lugares ocupados e desocupados e que a animação é feita.

long long int tempo_estadia[20]: nesse vetor, é armazenado um tempo padrão muito grande, para que facilite a manipulação. E posteriormente ele é utilizado para guardar o tempo de estadia de cada mesa do restaurante.

int s = 0, v = 0: são contadores, inicializados com 0, que armazenam o local que a mesa foi utilizada, “s” para “salão” e “v” para “varanda”, ou seja, se o usuário for escolher a primeira mesa do salão, s=1, segunda mesa, s=2, e assim por diante.

int varanda[4], salao[12]: esse vetor armazena valores, que são utilizados posteriormente para verificar se a mesa está ocupada ou não.

Explicação detalhada sobre as funções utilizadas:

void Zerar_Vetor(): utilizei essa função para que fosse mais fácil a manipulação dos vetores, sem q seja pego nenhum lixo de memória que coincida com os valores utilizados. Nela eu zero dois vetores, pois em outras funções verifico se o valor é 0 ou 1, e o vetor “tempo_estadia” eu atribuo valores muito altos, para facilitar a manipulação posteriormente.

void ler_arquivo(): utilizada apenas no início do programa, ela é responsável por ler o arquivo tipo texto(.txt) e atribui-lo a uma matriz, como foi solicitado nos planos do trabalho.

void sleep(int ms): utilizada para dar o delay ao mostrar algo na tela, fazendo com que seja possível mostrar ao usuário informações temporárias e facilitando o processo de animação, tornando-a mais fluída e visível ao usuário.

void mostrar_arquivo(): utilizada apenas para “printar” a matriz na tela, utilizando 2 loops de repetição e varrendo por toda a matriz.

void limpar_tela(): utilizando o comando(“clear”) do terminal do Linux, dentro do programa, para que fosse possível mostrar as versões atualizadas da matriz sem que a tela ficasse visualmente poluída.

char sim_nao(char letra): verifica se a pessoa digitou o carácter correto quando foi solicitado, se o carácter não for nenhum dos desejados, mostra-se um aviso e como deve ser digitado. Essa função se repete ate que o usuário digite o comando certo, nesse caso, um carácter entre esses: “s”, “S”, “n” e “N”; E o retorna para a “**int main()**”.

int um_quatro(int qnt): verifica se a pessoa digitou o inteiro correto quando foi solicitado, se o inteiro não for nenhum dos desejados, mostra-se um aviso e como deve ser digitado. Essa função se repete ate que o usuário digite o comando certo, nesse caso, um interio entre esses: “1”, “2”, “3” e “4”; E o retorna para a “**int main()**”.

int um_dois(int mesa): verifica se a pessoa digitou o inteiro correto quando foi solicitado, se o inteiro não for nenhum dos desejados, mostra-se um aviso e como deve ser digitado. Essa função se repete ate que o usuário digite o comando certo, nesse caso, um interio entre esses: “1” e “2”; E o retorna para a “**int main()**”.

int controle_restaurante(char continuar): na "**int main()**", é digitado um carácter, que é verificado na função "**char sim_nao(char letra)**", se o carácter escolhido for "n" ou "N", quer dizer q a pessoa deseja encerrar o programa, com isso, a função retorna "0", que devido a um "if" na "**int main()**", encerra o programa instantaneamente. Caso o carácter seja "s" ou "S", a função retornara "1", que faz com que o programa se mantenha no loop, ou seja, continua a ser executado.

int Check_Full(int mesa): utilizada para verificar se as mesas estão ocupadas ou não, e qual mesa deve ser ocupada se existirem mesas disponíveis. Se o local escolhido estiver ocupado é mostrado na tela e a função retorna um valor que é interpretado na "**int main()**" e dará sequencia para verificar se o usuário deseja trocar de lugar ou não.

void desocupar(int mesa): nessa função quando a mesa é ocupada ela recebe o tempo de estadia próprio, representado pelo vetor "tempo_estadia", que grava o tempo até q a mesa seja liberada. Após fazer isso, existe um loop que varre todo o vetor e verifica se em alguma das mesas o tempo estimado já foi alcançado, se isso ocorrer, é mostrado ao usuário qual mesa foi desocupada por 3 segundos e os locais que antes representavam as pessoas na mesa são alterados para representar a mesa vazia.

int lotacao_restaurante(int mesa): essa função é responsável por, ocupar a mesa. Exemplo: s=1, primeira mesa do salão é ocupada, v=1, primeira mesa da varanda ocupada. E atribuir "1" a posição da mesa, o que representa que ela está ocupada. Essa informação depois é utilizada na "**int Check_Full(int mesa)**" para verificar se todas as mesas já foram ocupadas ou não, e qual mesa o usuário deve se sentar. A partir da informação do local que foi ocupado, é possível fazer a animação e representar as pessoas sentadas em tais lugares.

void pessoas_mesa(int qnt, int mesa): essa função faz algo muito básico, a partir da posição da mesa que deve ser ocupada, o local da mesa e a quantidade de pessoas de cada mesa, é alterado na matriz "**char restaurante[17][41]**" a representação das mesas, ou seja, se a mesa for a primeira do salão e o usuário deseja que 1 pessoa se sente lá, as "/" e "\" são substituídos por "-" e "*", o "*" representa uma pessoa e a "-" é uma cadeira vazia.

void animacao(int mesa): responsável pela animação do personagem("0"), ele percorre a matriz e para no local designado para a mesa escolhida. Atravez de loops e utilizando o efeito da função "**void limpar_tela()**" e a "**void sleep(int ms)**" é possível representar a movimentação do personagem. Ao chegar no local designado é dado um "break" no loop e em seguida as mesas são alteradas pela função "**void pessoas_mesa(int qnt, int mesa)**".

int main(): essa função chama todas as outras em seus devidos momentos. No início para verificar se a pessoa quer entrar no controle do restaurante e se digitou os comandos adequados. Posteriormente zera os vetores e mostra a matriz restaurante. O grande loop consiste em perguntar o número de pessoas de cada mesa e a opção de mesa. A partir dessas informações é verificado se há espaço no local escolhido, se não, é perguntado se o usuario deseja mudar o local. Posteriormente, se o usuário decidir prosseguir, as animações ocorrem, a mudança da matriz é feita, para representar as pessoas e as mesas são desocupadas se o tempo for atingido. Em seguida o loop volta ao início e tudo se repete. Vale ressaltar que existe uma flag chamada "pula" que se o usuario decidir que não quer trocar a opção de mesa se a escolhida estiver lotada ele pula certas etapas do loop e volta a perguntar se o usuario deseja continuar controlando o restaurante.