AS INSTRUÇÕES PARA GERAR OS EXECUTÁVEIS ENCONTRAM-SE NO ARQUIVO MAKEFILE

INTRODUÇÃO

A cantina do ICEX sempre sofre com o excessivo fluxo de estudantes que por lá passam para se alimentar. Tal excesso pode ocasionar vários problemas, como filas muito grandes e a falta de bandejas para os usuários se servirem. Portanto, é necessário encontrar-se um modelo que distribua melhor o fluxo de pessoas, diminuindo o tamanho das filas e aumentando a eficiência do serviço de bandejas, a fim de minimizar o tempo de espera/atendimento dos clientes.

DESENVOLVIMENTO

Foram utilizadas, para o caso base, duas filas (implementadas por ponteiros) e uma pilha (implementada por arranjos). É necessário ressaltar, porém, que a implementação da(s) pilha(s) leva em consideração apenas o topo da mesma, pois o conteúdo dos itens da pilha não importa, mas sim o número de itens presentes, portanto as funções relacionadas à pilha alteram apenas o valor que o topo da pilha assume. Isso é válido para todas as configurações testadas.

Para o caso de duas filas 1 (filas do caixa), optei por utilizar dois caixas. Ficou claro, no entanto, que não há a necessidade de filas 1, já que os clientes irão direto para a fila 2 (fila de bandejas). Neste caso, a fila 2 foi implementada por ponteiros e a pilha, por arranjo.

De maneira semelhante, optei por utilizar duas pilhas ao invés de duas filas para pilhas, já que isso não faria diferença no resultado. Neste caso, foram utilizadas 2 pilhas, implementadas por arranjo, e as filas por ponteiros.

Na configuração ideal, foram utilizados 2 caixas, o que exclui a necessidade de implementação da fila 1(fila do caixa), 2 pilhas de 40 bandejas, implementadas por arranjo, o que também exclui a necessidade de uma fila 2 (fila das bandejas). Ao final, utilizando-se duas mesas para cada alimento, a implementação não é feita através de uma estrutura específica, mas sim de variáveis que assumem as pessoas que por elas estão passando.

IMPLEMENTAÇÃO

Antes de explicar linha por linha, deixo claro que as funções de filas utilizadas (enfileira, desenfileira, etc), foram implementadas exatamente como nos slides dados em sala de aula, portanto acredito não haver a necessidade de explicação aprofundada de como elas funcionam; aquelas que não estão presentes nos slides serão explicadas.

Linhas 12-26: Apenas declaração de variáveis e inicialização das filas e da pilha. As funções InitFila são responsáveis por inicializar as filas que serão utilizadas, e o fazem por alocar a Frente da fila dinamicamente, de forma que o Tras também aponte para a mesma célula "criada". Já a função InitPilha, por não ser importante o conteúdo da pilha, mas sim o número de bandejas que o topo da mesma representa, apenas define o topo da pilha como o número máximo de bandejas permitido.

Linha 28: Inicialização do loop. O loop irá parar após o minuto 240, tempo máximo de 4 horas pedido.

Linha 29: Função de inicialização das pessoas que chegam à fila (É definido aqui a identidade das pessoas, referente à variável Ident, que representa o seu tempo de entrada na fila 1, dessa forma é possível possuir pessoas variadas de acordo com seu tempo de entrada. A variável Count é iniciada como 0, e definirá, ao final, o tempo total de atendimento da pessoa. As variáveis JaArroz, JaFeijao, etc... servem como marcadores para que, ao servirem-se das comidas, não se sirvam novamente da mesma).

Linhas 30 e 31: Funções de enfileiramento. Como chegam 2 pessoas por minuto à fila, são realizados dois enfileiramentos em sequência.

Linha 33: Após passar pelo caixa, a primeira pessoa da fila é desenfileirada.

Linha 34: O apontador referente ao caixa passa a apontar para a pessoa desenfileirada. Esse procedimento serve apenas para facilitar a visualização da pessoa presente no caixa.

Linha 35: O Count da pessoa, referente ao seu tempo total de atendimento, é acrescido. Esse é um procedimento necessário para separarem-se as interações que gastam tempo que a pessoa executa, pois sem isso, a pessoa executará 3 ações simultaneamente em um único espaço de tempo.

Linha 36: A pessoa desenfileirada, após passar pela interação do caixa, que consome espaço de tempo, é então enfileirada na fila 2.

Linhas 38-40: A cada 12 espaços de tempo, a função AcrescentaBandeja empilhará mais bandejas, acrescendo o topo da pilha até 10, desde que a pilha não ultrapasse o limite de 30 bandejas.

Linhas 42-44: Caso a pilha esteja vazia (ou seja, caso Pilha->Topo ==0 seja TRUE) o usuário deverá esperar pela reposição. Sendo assim, o apontador BandejaDisponivel apontará para NULL. Isso é importante, pois o arroz só poderá ser servido caso o BandejaDisponivel não aponte para NULL.

Linhas 46 e 47: Caso a pilha não esteja vazia, uma bandeja será desempilhada da mesma (decrescendo-se 1 do topo da pilha), e a pessoa responsável pela ação será desenfileirada da fila 2.

Linha 48: Fazemos aqui o mesmo procedimento adotado na linha 36, o Count++ serve para evitar a realização de ações simultâneas.

Linha 49: A variável BandejaDisponivel passa a apontar para a pessoa desenfileirada.

Linhas 53-55: Condição para que a pessoa sirva-se de salada, em que ela já precisa ter se servido de guarnição e não de salada ainda. Isso ajuda a manter a ordem das pessoas que saem da fila 2, além de evitar a repetição dos pratos. Dessa forma, a variável PessoaSalada assume a pessoa que estava em PessoaGuarnicao, e essa pessoa é marcada com JaSalada = 1, para indicar que esta já se serviu de salada. Importante ressaltar que o primeiro alimento que a pessoa irá se servir será o arroz, porém os ifs estão em ordem invertida pois, caso contrário, a pessoa irá se servir dos 4 alimentos de uma vez só, pelo fato de a condição do if ser sempre verdadeira.

Linhas 57-64: Funcionamento idêntico ao procedimento da salada, porém com o feijão e a guarnição.

Linha 65-69: No caso do arroz, o funcionamento é parecido, com a PessoaArroz assumindo a pessoa apontada por BandejaDisponivel e marcando-a com JaArroz = 1. Porém, nesse caso, temos a utilização de uma flag, a qual serve apenas para marcar a entrada da pessoa nos procedimentos das comidas. Além disso, mais uma vez fazemos Count++, para evitarmos interações simultâneas no código.

Linhas 72-74: Finalmente, temos que, assim que a pessoa for marcada com JaSalada = 1, esta completará seu atendimento e terá sua contagem definida por (Count += (tempo total - Ident)), que calcula o tempo atual, em espaços de tempo, menos o tempo em que ela entrou na fila, mais (+=) o tempo que foi somado nas interações individuais ao longo do código (Count++). O somatório que definirá a média também é definido aqui.

Linhas 78 e 79: Após o fim do loop, será calculado o valor da média do tempo de atendimento e printado.

ANÁLISE

Como, em todos as configurações, temos um número máximo pré-definido de elementos que entram no programa, e funções que são executadas da ordem O(1), então todos os casos terão uma complexidade de O(1).

RESULTADOS

Configuração: caso base.

Obtive, em minha implementação, uma média de 73 minutos para o atendimento, em que 218 pessoas foram atendidas.

Configuração: 2 caixas, todo o resto mantido como no caso base.

Não faria diferença implementar duas ou uma fila para o caso de um único caixa, portanto, serão considerados aqui dois caixas em uma fila única. Nesta configuração, duas pessoas são atendidas a cada espaço de tempo, enquanto que chegam duas pessoas na fila a cada espaço de tempo. Portanto, percebe-se que a fila 1 (fila do caixa) torna-se obsoleta neste cenário. Apesar de excluir a necessidade de um das filas, o resultado deste cenário não observa alteração em relação ao caso base, pois a maior agilidade na fila 1 resultará em menor agilidade na fila 2, já que o déficit de bandejas será maior. Portanto, novamente, média de 73 minutos para 218 pessoas atendidas.

Configuração: 40 bandejas na pilha, todo o resto mantido como no caso base.

Obviamente, o maior número de bandejas na pilha ajuda a diminuir o déficit total de bandejas, o que aumenta ligeiramente a eficiência do sistema. 228 pessoas atendidas e 68 minutos de média.

Configuração: 40 bandejas, 2 caixas, todo o resto mantido como na base.

Nesse cenário, a pilha de bandejas chega em seu limite, mas não fica sem bandejas em momento algum, o que aumenta novamente a eficiência. O resultado, porém, ainda não é o esperado, pois o procedimento das comidas ainda toma muito tempo. Importante ressaltar que, devido ao fato de que a pilha nunca fica vazia, alterações no intervalo de reposição não surtirão efeitos. 237 pessoas atendidas e 65 minutos de média.

Configuração: intervalo de reposição = 11min, 2 caixas, todo o resto mantido como na base.

Semelhante ao caso acima, a pilha chegará em seu limite, mas não faltará nenhuma bandeja, mesmo com o topo travado em 30. Sendo assim, mesmo diminuindo-se o intervalo de reposição, ou aumentando-se o máximo de bandejas, o resultado será o mesmo, já que o procedimento das comidas não foi alterado. 237 pessoas atendidas e 65 minutos de média.

Configuração: número de bandejas repostas = 11, 2 caixas, todo o resto mantido como na base.

Idêntico ao que ocorre com a configuração anterior.

Configuração: 2 caixas, 2 pilhas de 40 bandejas, e 2 mesas para cada alimento, todo o resto mantido como a base.

Ao utilizarem-se 2 caixas e 2 pilhas com 40 bandejas cada, exclui-se a necessidade de quaisquer filas no sistema, já que chegam 2 pessoas por espaço de tempo, e ambas vão ser atendidas no caixa e pegarão bandejas simultaneamente, as filas não irão encher. Após pegarem as bandejas, cada pessoa poderá pegar o seu alimento separadamente, o que não gastará o tempo daqueles que estão atrás. Importante salientar que os resultados seriam os mesmos caso, ao invés de duas pilhas de 40 bandejas, tivéssemos duas pilhs de 30 com taxa de reposição de 11 a cada 12 minutos, ou taxa de reposição de 10 a cada 11 minutos. Desta forma, obtemos o tempo mínimo de 6 minutos de média, com 474 pessoas sendo atendidas.

CONCLUSÃO

Percebe-se, portanto, que a configuração que utiliza 2 caixas, 2 pilhas com 40 bandejas cada, e 2 mesas para cada alimento ser servido mostrou-se a mais eficiente, dado que, neste cenário, as filas são extintas e o impacto da espera das bandejas torna-se nulo. O aumento de mesas para servirem-se os alimentos é essencial, pois dessa forma, todas as pessoas que estão em seu atendimento poderão se servir independentemente, sem precisarem esperar o atendimento da pessoa posterior. Assim, maximiza-se o número de pessoas atendidas, e essas pessoas gozam de um tempo médio de 6 minutos, o mínimo necessário para completar-se o atendimento.