Documentação - Áulus Pinho

Introdução:

O código proposto representa um cenário de um restaurante em funcionamento durante um período de 4 horas. Neste período, a cada 1 minuto duas pessoas chegam na fila que antecede ao caixa(Fila 1), uma pessoa finaliza seu atendimento no caixa, 1 pessoa finaliza o processo de obtenção de uma bandeja e prato e 1 pessoa serve-se de uma única guarnição. Totalizando 4 guarnições, o tempo total para que cada pessoa sirva-se em todas elas é de 4 minutos. A Pilha de pratos oferecida pelo restaurante contém um número fixo de 30 pratos, repostos em dez unidades a cada período de doze minutos.

O objetivo do código é propor um tempo médio que cada pessoa leva para completar seu atendimento (entrar na fila até o terminar de servir-se dos alimentos).

Desenvolvimento:

Para realização da tarefa, foram utilizados Tipos Abstratos de Dados (TADs): Filas, Pilhas e respectivas Células e Itens. Ao total, no caso base (Descrito na Introdução acima) foram criadas 3 Filas - 1 de entrada, 1 que precede à pilha de bandejas e 1 representando o processo de serviência dos alimentos - e 1 Pilha(de bandejas).
Estruturando o processo de funcionamento do restaurante, foi considerado que no tempo inicial (igual a 0) 1 pessoa chegaria diretamente ao caixa e 1 ocuparia a primeira posição da fila inicial de espera. No tempo 1 a pessoa que estava no caixa dirigia-se à segunda fila para, em seguida, pegar uma bandeja e servir-se com a primeira guarnição; a pessoa que estava na fila inicial entrava no caixa e mais duas pessoas entravam na primeira fila. A partir daí o processo se repete enquanto o tempo for menor que duzentos e quarenta minutos. O processo total de sair da segunda fila, pegar uma bandeja e em seguida servir-se da primeira guarnição só acontece se a pilha de bandejas contiver ao menos uma. É importante ressaltar que o processo específico de sair da segunda fila para pegar uma bandeja parava a partir do tempo duzentos e trinta e cinco(inclusive), já que nos minutos seguintes até duzentos e quarenta(exceto) ninguém que começasse a servir-se dos alimentos conseguiria terminar no tempo hábil.

O cálculo da média dá-se somando os tempos próprios de cada cliente e dividindo pelo total de clientes que terminaram todo o processo. Os tempos de cada cliente são devidamente armazenados nas estruturas relativas a cada um deles, e contados subtraindo do tempo em que terminaram de servirem-se da última guarnição, o tempo em que entraram na primeira fila.

Implementação:

As funções do codigo a serem destacadas sao:

- Função de Verificação:

```
void VerificaTudo(int tempo,TPilha *Pr){
   int k,tam = 30 - Pr->Tamanho;
         if((tempo % 12)==0){
                  if(tam > 0 && tam < 10){
                           for(k=0;k<tam;k++){
                                    Empilha(Pr);
                           }
                  }
                  else if(tam > 10 && tam <31){
                           for(k=0;k<10;k++){
                                    Empilha(Pr);
                           }
                  }
         }
}
```

Tal função é responsável por repor a pilha de bandejas em no máximo dez unidades a cada doze minutos. O número máximo de bandejas (trinta) é respeitado. Recebe por valor o tempo atual - para efeito de comparação - e por referência a pilha de bandejas - de modo a permitir o Empilhamento de novos elementos(bandejas).

- Função de chegada do cliente na fila inicial:

```
void ChegaCliente(TFila *Fi, TItem *cont,int tempo){
   cont->Chave +=1;
   cont->Time = tempo;
   //printf("chave: %d",cont->Chave);
         Enfileira(*cont,Fi);
   cont->Chave +=1;
   cont->Time = tempo;
   //printf("chave: %d",cont->Chave);
         Enfileira(*cont,Fi);
}
```

Tal função é responsável por Enfileirar na primeira fila (fila inicial precedente ao caixa) mais 2 clientes, dado o incremento do tempo em 1 minuto. Recebe por referência uma fila (Fila 1, no caso) e um item; este último conterá as informações do cliente, a saber seu respectivo número e tempo de entrada na fila.Recebe por valor o tempo atual, que comporá as informações do item relativo ao cliente.

- Função representante do Caixa:

```
void Caixa(TFila *Ent,TFila *Sai){
         TItem Ficha;
         Desenfileira(Ent,&Ficha);
```

```
        Enfileira(Ficha,Sai);
}
```

Tal função ao receber por referência 2 Filas, sendo a primeira a fila de origem (Fila 1) e a segunda a fila de destino (Fila 2). É responsável por Desenfileirar o cliente da fila inicial e Enfileira-lo na fila que antecede a pilha das bandejas. Um TAD Item é criado para permitir a cópia das informações do cliente.

- Condicional:

```
if(t<236){

        if(!PVazia(Pilha1)){

                Desenfileira(&Fila2,&Temp);
                Desempilha(&Pilha1);
                t++;
                Temp.Time = (t - Temp.Time) + 4;
                Enfileira(Temp,&Almoco);
        }
        else
                t++;
}
else
        t++;
```

Nesta condicional é feita o Desenfileiramento de um cliente, o Desempilhamento de uma bandeja e o subsequente Enfileiramento do cliente na Fila de Almoço, culminando no incremento de 1 unidade na variável responsável pelo tempo do programa. Por estes processos é possível representar o ato de saída do cliente da fila que antecede a pilha de bandejas, o ato de pegar uma bandeja e se dirigir até a primeira das 4 guarnições. Todo esse processo gasta 1 minuto e, portanto, o tempo é incrementado em uma unidade. Dentro da lógica do programa, a Fila Almoco será usada como base de contagem do número de clientes que terminaram o processo de atendimento. Desta maneira, considerando o tempo de 4 minutos para o cliente se servir das 4 guarnições, o tempo total gasto pelo mesmo é resultante da subtração do tempo de entrada na Fila 1 pelo tempo atual e, logo após, o resultado é somado a mais 4 unidade de tempo.

- Loop:

```
while(!FVazia(Almoco)){
        Desenfileira(&Almoco,&Temp);
        TempoTotal += Temp.Time;
        PessoasTotal +=1;
}
```

Neste Loop é feito a contagem de pessoa que finalizaram o atendimento baseado na lógica do programa outrora explicada. Desenfileira-se a Fila Almoco, somam-se os tempos de atendimento de cada cliente e incrementa-se o número total de clientes 1 unidade por vez.

Análise das Complexidades:

Dado que

- o while principal executa tudo 240 vezes,
- cada função é chamada uma única vez,
- cada função tem em si ao menos uma função de (Des)Empilhamento/(Des)Enfileiramento,

A complexidade total do programa é O(n) para todos os casos.

Resultados:

Com o teste do caso inicial obtiveram-se 218 clientes totais que completaram seu atendimento em um tempo médio de 64 minutos.