

AEDS 2 - TP

Introdução

Para a implementação deste trabalho, foi proposto um problema que enfrentamos todos os dias. As filas em uma cantina, especificamente a cantina do prédio ICEX. Era necessário codificar um programa em que fosse possível controlar as filas do ICEX, controlando quantas pessoas chegam na fila em um determinado espaço de tempo, quantas pessoas saem de cada fila, etc.

O intuito deste programa era ter como resultado a média de tempo gasto pelas pessoas nas filas em um espaço de 4 horas. 4 horas correspondem a 240 iterações, já que cada iteração (intervalo de tempo) equivale a 1 minuto.

Foi necessário criar estruturas de dados como filas e pilhas, fazer com que a cada iteração fossem adicionadas pessoas na fila de compra de ficha. Além disso foi necessário o controle entre uma fila e outra, a cada iteração retirar pessoas de uma fila e colocar em outra, sempre adicionando o tempo e guardando o valor do tempo gasto por cada pessoa.

A partir do momento que podemos calcular a média gasta por cada pessoa, conseguimos alterar alguns dados, como número de pessoas atendidas por minuto na fila de fichas, número de pratos na pilha, etc., com o intuito de melhorar a eficiência das filas.

Desenvolvimento

Para a realização do programa, foram utilizadas duas filas e uma pilha. Fila para compra de fichas, fila para pegar bandeja e fila para servir comida. Pilha de pratos.

O número de pessoas que chegam em cada fila foi definido como variável, para que pudessem ser testados vários valores, a fim de encontrar a melhor média de tempo.

Inicialmente são definidas as variáveis que serão usadas ao longo do programa, e também são inicializadas as filas e a pilha de pratos.

As primeiras pessoas são inseridas na fila de compra de ficha, com tempo igual a zero, de acordo com a quantidade definida de pessoas que chegam na fila por minuto.

Logo após, começa-se o loop com as iterações que simbolizam os minutos. Ele vai de zero a duzentos e quarenta, que equivale a quatro horas.

Dentro do loop, as operações são realizadas em ordem inversa, para não acontecer de duas operações serem realizadas por uma pessoa na mesma iteração, acrescentando assim somente um minuto em seu tempo gasto. Por exemplo, se dentro do loop eu acrescento um minuto no tempo de uma pessoa que está na fila, e logo depois ela é atendida e muda de fila, ela estaria realizando várias coisas no mesmo minuto. Realizando as operações ao contrário, primeiro a pessoa sai da fila, sendo assim quando a fila é percorrida acrescentando um minuto em cada pessoa que ainda está esperando, a primeira da fila já foi retirada e inserida na próxima fila.

Depois de todas as operações serem feitas e o tempo contado, a cada pessoa que termina seu fluxo e sai da última fila com todos os alimentos, uma variável é salva com seu tempo total gasto, que somado com todos os outros e dividido pelo número de pessoas que concluíram o fluxo gera a média de tempo gasto.

Implementação

```
num_clientes = 0;
// número de iterações igual a quantidade de segundos
for(tempo_total=0;tempo_total<240;tempo_total++){

    if (aux_comida == 0){
        if(vazia(FilaFicha)==0){
            tempo_aux = retira(FilaComida);
            tempo_media = tempo_media + tempo_aux;
            num_clientes++;
            aux_comida = 4;
        }
    }

    if(insere_pratos == 0){
        for(i=0;i<10;i++){
            if(num_pratos<=max_bandejas){
                push(PilhaPratos);
                num_pratos++;
            }
        }
        insere_pratos = intervalo;
    }

    // a cada iteração, x pessoas pegam bandeja:
    //for para quantidade de pessoas que pegam bandeja (saem da filaBandeja e entram na filaComida por segundo)
    percorreFila(FilaComida);
    for(i=0;i<entrada_filaComida;i++){
        if(vazia(FilaBandeja)==0){
            if(vaziaPilha(PilhaPratos) == 0){
                tempo_aux = retira(FilaFicha);
                pop(PilhaPratos);
                num_pratos--;
                insere(FilaComida, tempo_aux+1);
            }
        }
    }
}
```

Toda a lógica do meu programa está no main, e as funções de manipulação no outro arquivo.c. Como dito no tópico acima, as operações foram feitas ao contrário para garantir uma execução mais precisa.

Existe uma variável chamada `aux_comida` que, como o próprio nome diz, é uma variável auxiliar. Essa variável tem o valor de 4 inicialmente e vai decrescendo a medida que alguém vai servir a comida. Assim, a pessoa só sai da fila depois de gastar 4 intervalos de tempo para servir a comida.

Caso exista alguém na fila para a comida que já tenha gastado 4 tempos para servir a comida, essa pessoa é retirada da fila e seu tempo é armazenado na variável `tempo_media`.

A função `percorreFila()` é a responsável por adicionar um minuto em todas as pessoas que estão na fila esperando. Ela é chamada para todas as filas, e se a fila estiver vazia não faz nada.

Existe também uma lógica para a pilha de pratos, em que os pratos tem uma quantidade máxima definida de acordo com os padrões do programa, e um intervalo de reposição, que é contado com as iterações.

```

void percorreFila(Node *FILA){
    if(vazia(FILA)){
        //fila vazia, não acrescenta nada
    }
    else{
        Node *tmp = FILA->prox;
        while(tmp != NULL){
            tmp->tempo++;
            tmp = tmp->prox;
        }
    }
}

```

Função percorreFila, que é utilizada para percorrer todas as filas acrescentando 1 no tempo de quem ainda está na fila esperando.

Análise de complexidade

- Main : $O(n^2)$;
- Inicia: $O(1)$;
- iniciaPilha: $O(1)$;
- Vazia e vaziaPilha: $O(1)$;
- Aloca e alocaPilha: $O(1)$;
- Todas as outras são $O(n)$;

Resultados

Os resultados podem variar muito de acordo com a configuração proposta. Quanto mais pessoas chegando na fila, maior o acúmulo e maior o tempo de espera.

Alguns exemplos:

```
alinetorres@aline-pc:~/Área de Trabalho/aeds/cantina$ ./cantina
 Bem vindo ao programa de controle da cantina!
 Escolha algumas configurações de controle das filhas e pilha para que a media de
 tempo gasto por cliente seja calculado.
 Digite o número de pessoas que chegam a fila de fichas por minuto:1

 Digite o número de pessoas que são atendidas por minuto:1

 Digite o número de pessoas que pegam bandejas por minuto:1

 Digite o número máximo de bandejas:1

 Digite o intervalo de tempo para repor bandejas:1

tempo medio: 91.000000
```

```
 Bem vindo ao programa de controle da cantina!
 Escolha algumas configurações de controle das filhas e pilha para que a media de
 tempo gasto por cliente seja calculado.
 Digite o número de pessoas que chegam a fila de fichas por minuto:2

 Digite o número de pessoas que são atendidas por minuto:1

 Digite o número de pessoas que pegam bandejas por minuto:1

 Digite o número máximo de bandejas:30

 Digite o intervalo de tempo para repor bandejas:12

tempo medio: 92.000000
```

Conclusão

Com esse trabalho foi possível realizar o controle das filas de uma cantina. Foi possível perceber que quanto mais pessoas para atender e quanto menos pessoas chegam na fila, maior é a chance de gastar um tempo menor.

A fila de servir a comida faz com que as pessoas se acumulem, por causa da demora de 4 tempos para servir comida.

Quanto mais recursos, para que mais de uma pessoa faça algo ao mesmo tempo, menor a média de tempo gasto.