

Instituto Federal de Minas Gerais - IFMG Campus Formiga
Danilo Araújo Domingues Silva

Documentação do Trabalho Simulador de Aeroporto

Estrutura de Dados
Profª Denise Rezende

Formiga
2022

O código que construí para o Trabalho de Estrutura de Dados de Simulação de Aeroporto consiste em 3 arquivos que se interligam por meio do uso da biblioteca “.h” criada.

O arquivo “Funcoes.h” contém apenas o cabeçalho de todas as funções construídas e utilizadas no código, funções estas que estão presentes por sua vez no arquivo “Funcoes.c”, que por sua vez são “chamadas” na função main, que está presente no arquivo principal “ED_AeroportoV10” (que é a décima primeira versão do código).

Inicialmente são criadas as variáveis globais, que são: ‘ID’, a identificação global de cada avião gerado; ‘iteracao’, que nos permite fazer o controle de quantas vezes o código será executado, já que a cada finalização de cada execução a variável iteração é então iterada; e por fim ‘quedas’, que contabiliza o número de aviões que não conseguiram pousar, e então caíram.

Para uso destas, todas as pistas, prateleiras e filas de decolagem são iniciadas. Então, já dentro do loop while (que no caso enquanto a variável ‘iteracao’ for diferente de 1000, o código continua sendo executado), a função menor_prat é chamada, sendo que esta verifica todas as prateleiras, e pega a menor delas para que o avião seja gerado e inserido.

A função menor_FD, que possui estrutura muito semelhante à função menor_prat é chamada logo em seguida para que seja realizada a verificação da menor fila de decolagem também para que os aviões sejam gerados e inseridos.

```
while(iteracao!=1000){
    printf("\n*****\n");

    //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ VERIFICACAO DA MENOR ESTRUTURA $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

    //verificando qual a menor prateleira para insercao dos avioes
    menor_prat(&prat1, &prat2, &prat3, &prat4, &ID);

    //verificando qual a menor fila de decolagem para insercao dos avioes
    menor_FD(&FD1, &FD2, &FD3, &ID);

    //sleep(3);

    //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ FIM DA VERIFICACAO DA MENOR ESTRUTURA $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
}
```

Todas as filas, prateleiras e pistas são impressas no terminal e em seguida as seguintes funções são chamadas:

```
emergencia(&prat1, &prat2, &prat3, &prat4, &p3);
```

```
chosen_p1(&prat1, &prat2, &p1);
```

```
chosen_p2(&prat3, &prat4, &p2);
```

```
chosen_FD(&FD1, &FD2, &FD3, &p3);
```

```
pistavazia_p1_p2(&FD1, &FD2, &FD3, &p1, &p2);
```

```
fuel_dec(&prat1, &prat2, &prat3, &prat4, &quedas);
```

O funcionamento de todas estas funções são bem simples, mas talvez não tão fáceis de se entender o controle dos aviões e as condições para tal controle. A função emergência verifica se alguma das 4 prateleiras de pouso possui algum avião na segunda posição de cada uma delas verificando se o combustível destes possui valor igual a 1. Caso tenha e a pista 3 esteja vazia (lembrando que a pista 3 seria basicamente somente para decolagem dos aviões das filas de decolagem), esta função manda o primeiro avião desta fila para pista 3, já que não se pode retirar do meio, e então, como as funções são chamadas na ordem anteriormente apresentada, o avião com 1 de combustível é enviado para uma das pistas (p1 ou p2) pela função ou chosen_p1 (prateleira escolhida para pouso da pista 1) ou chosen_p2 (prateleira escolhida para pouso da pista 2), pois dentro destas funções (chosen_p1 e chosen_p2), antes da verificação da quantidade de aviões em cada prateleira, é verificada a quantidade de combustível de cada primeiro avião das prateleiras, ou seja, ele priorizará o pouso do avião com combustível 1.

A função chosen_FD apenas define qual a maior fila de decolagem para pousar o avião da maior prateleira. Dentro desta função é chamada outra função para a efetiva decolagem. A estrutura é a mesma para chosen_p1 e chosen_p2, a diferença é que para chosen_p1 são passadas as prateleiras 1 e 2, e para chosen_p2 as prateleiras 3 e 4. Dentro de chosen_p1 e chosen_p2 também são chamadas outras funções para o efetivo pouso.

A função 'pistavazia_p1_p2' faz a verificação de que se a pista 1 e/ou 2 estiverem vazias, estas são então usadas para a decolagem dos aviões para as prateleiras, otimizando o código e evitando de deixar as pistas sem utilização.

A função 'fuel_dec' apenas decrementa o combustível de todos os aviões presentes nas prateleiras (já que os aviões estão voando (a função não é chamada para as filas de decolagem já que aviões que estão para decolar, mas ainda estão em terra não gastam combustível)) a cada iteração/execução do código.

Logo todas as pistas, prateleiras e filas de decolagem são impressas novamente no terminal, mas desta vez mostrando para onde cada avião foi movimentado. Antes de o código ser executado novamente, as pistas são limpas com a função retira_aviao, sendo chamada 3 vezes, uma para cada pista.

A seguir, apenas um exemplo de código, da função 'emergencia':

```
//$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ POUSO NA PISTA 3 EM SITUAÇÃO  
DE EMERGENCIA $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
//VERIFICADA  
//funcao que pouso um aviao em emergencia  
void insere_emergencia(tpfila *fila, tpfila *p3){  
    //se a prateleira estiver vazia, nao tem de onde tirar o aviao, por  
    isso a verificacao  
    if (!estavazia(*fila)){//se a funcao estavazia retornar 1, quer  
    dizer que esta vazia, caso contrario, nao esta' vazia  
        if (estavazia(*p3)){//se a pista 3 estiver vazia, insere
```

```

        p3->ultimo->proximo = fila->primeiro->proximo; //apontando a
primeira casa utilizavel da pista 3 para o primeiro aviao da maior
pista analisada anteriormente
        p3->ultimo = p3->ultimo->proximo; //andando com a ultima
casa, ja' que acebei de "inserir"
        p3->ultimo->Tf = time(NULL);
                                printf("\nTempo de Pouso: %.0fs\n",
diffftime(p3->ultimo->Tf, p3->ultimo->Ti));

        if(fila->primeiro->proximo->proximo == NULL){ //se
fila->primeiro->proximo->proximo for igual a NULL, entao so' ha' uma
casa na prateleira
                fila->primeiro->proximo = NULL; //e se ha' so' uma casa
na prateleira, eu "elimino" esta casa
                fila->ultimo = fila->primeiro; //se eu retirei todas as
casas desta lista, agora ela esta' nula, ou seja, primeiro = ultimo
        }else{
                                fila->primeiro->proximo =
fila->primeiro->proximo->proximo; //se a fila nao tiver apenas uma casa,
eu simplesmente ignoro a primeira casa utilizavel e aponto para a
segunda,

//excluindo a primeira, ou seja, retirando o primeiro aviao
        }

        p3->ultimo->proximo = NULL; //tirando a ligacao entre o
proximo aviao para nao levar todos os avioes juntos para a pista
        fila->qtd--; //decremento a quantidade de itens da minha
prateleira que acabei de retirar
    }
}

//VERIFICADA
//funcao que antes de aterrissar qualquer aviao, ele verifica se ha
algum aviao na segunda posicao com combustivel igual a 1
void emergencia(tpfila *prat1, tpfila *prat2, tpfila *prat3, tpfila
*prat4, tpfila *p3){

    if(estavazia(*p3)){
        tpfila *emergencia = prat1; //de inicio o ponteiro ja recebe o
endereco da prateleira 1, para poder comparar

```

```

        //abaixo eu primeiro verifico se uma das prateleiras esta
vazia, pois se uma delas estiver vazia,
        //nao tem como eu comparar para ver se o combustivel esta em
caso de urgencia

        //neste codigo estou considerando como urgencia ter menos de 3
de combustivel

        if(!estavazia(*prat1)){//se a prateleira 1 nao estiver vazia
            if(prat1->primeiro->proximo->proximo != NULL){//se existir
um segundo aviao nesta lista
                if(prat1->primeiro->proximo->proximo->fuel ==
1){//primeiro e' verificado a urgencia de acordo com o combustivel
                    emergencia = prat1;//se o combustivel do primeiro
aviao for menor ou igual a 3, logo, a prateleira escolhida para
retirada e' a 1
                    insere_emergencia(emergencia, p3);//chamando funcao
para insercao na pista 1 e retirando o aviao da prateleira escolhida
                    return;//como a prateleira ja foi escolhida para
retirada, ele sai da funcao
                }
            }
        }

        if(!estavazia(*prat2)){//se a prateleira 1 nao estiver vazia
            if(prat2->primeiro->proximo->proximo != NULL){//se existir
um segundo aviao nesta lista
                if(prat2->primeiro->proximo->proximo->fuel ==
1){//primeiro e' verificado a urgencia de acordo com o combustivel
                    emergencia = prat2;//se o combustivel do primeiro
aviao for menor ou igual a 3, logo, a prateleira escolhida para
retirada e' a 1
                    insere_emergencia(emergencia, p3);//chamando funcao
para insercao na pista 1 e retirando o aviao da prateleira escolhida
                    return;//como a prateleira ja foi escolhida para
retirada, ele sai da funcao
                }
            }
        }

        if(!estavazia(*prat3)){//se a prateleira 1 nao estiver vazia
            if(prat3->primeiro->proximo->proximo != NULL){//se existir
um segundo aviao nesta lista

```

```

        if(prat3->primeiro->proximo->proximo->fuel ==
1){//primeiro e' verificado a urgencia de acordo com o combustivel
            emergencia = prat3;//se o combustivel do primeiro
aviao for menor ou igual a 3, logo, a prateleira escolhida para
retirada e' a 1
            insere_emergencia(emergencia, p3);//chamando funcao
para insercao na pista 1 e retirando o aviao da prateleira escolhida
            return;//como a prateleira ja foi escolhida para
retirada, ele sai da funcao
        }
    }
}

if(!estavazia(*prat4)){//se a prateleira 1 nao estiver vazia
    if(prat4->primeiro->proximo->proximo != NULL){//se existir
um segundo aviao nesta lista
        if(prat4->primeiro->proximo->proximo->fuel ==
1){//primeiro e' verificado a urgencia de acordo com o combustivel
            emergencia = prat4;//se o combustivel do primeiro
aviao for menor ou igual a 3, logo, a prateleira escolhida para
retirada e' a 1
            insere_emergencia(emergencia, p3);//chamando funcao
para insercao na pista 1 e retirando o aviao da prateleira escolhida
            return;//como a prateleira ja foi escolhida para
retirada, ele sai da funcao
        }
    }
}

}

}

//$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ FIM POUSO NA PISTA 3 EM
SITUACAO DE EMERGENCIA $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

Por último, a imagem abaixo mostra de forma ilustrativa o conceito do trabalho:

Imagem ilustrativa feita por Danilo Araújo Domingues Silva - IFMG Formiga

