universidade de aveiro

Sistemas Operativos

Professor Nuno Lau

Departamento de Eletrónica, Telecomunicações e Informática

Simulação de Ponte Aérea

Ana Raquel Paradinha	102491
Paulo Pinto	103234

A participação na realização do trabalho foi igualitária.

Índice

1 Introdução

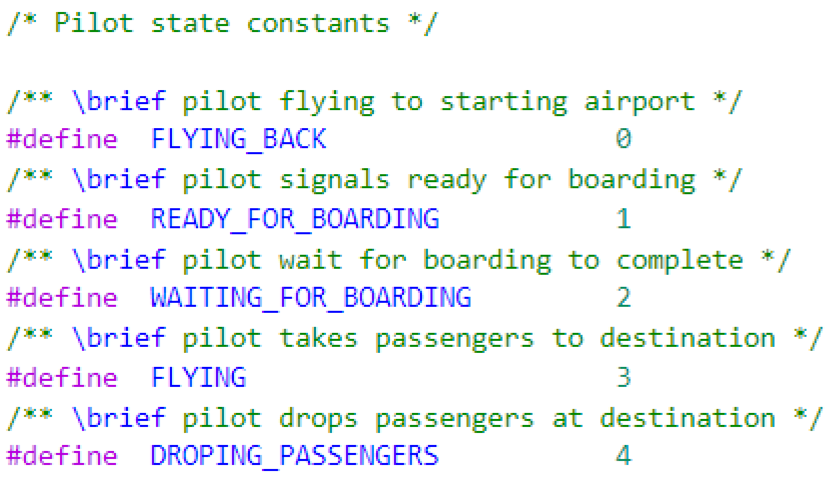
No âmbito da unidade curricular Sistemas Operativos foi-nos proposto desenvolver parte do código de vários scripts dados pelo professor, os quais usados em conjuntos permitem simular vários voos e passos deste. Nesse sentido, existem 3 operadores responsáveis pelo processo, a hospedeira, o piloto e os passageiros, e através do uso de semáforos foi possível definir quando e que ações cada um deles devia tomar.

Com efeito, com base nos conteúdos explorados nas aulas teóricas e práticas e alguma pesquisa em fontes externas conseguimos chegar ao resultado esperado. Nesse sentido, no presente relatório descrevemos as metodologias utilizadas para chegar a uma solução, assim como as dificuldades que encontramos e como foram resolvidas.

2 Implementação

2.1 Piloto

Para a nossa simulação da ponte aérea, começamos por implementar o piloto, uma vez que era o mais simples. Nesse sentido, este interveniente pode assumir cinco estados diferentes:



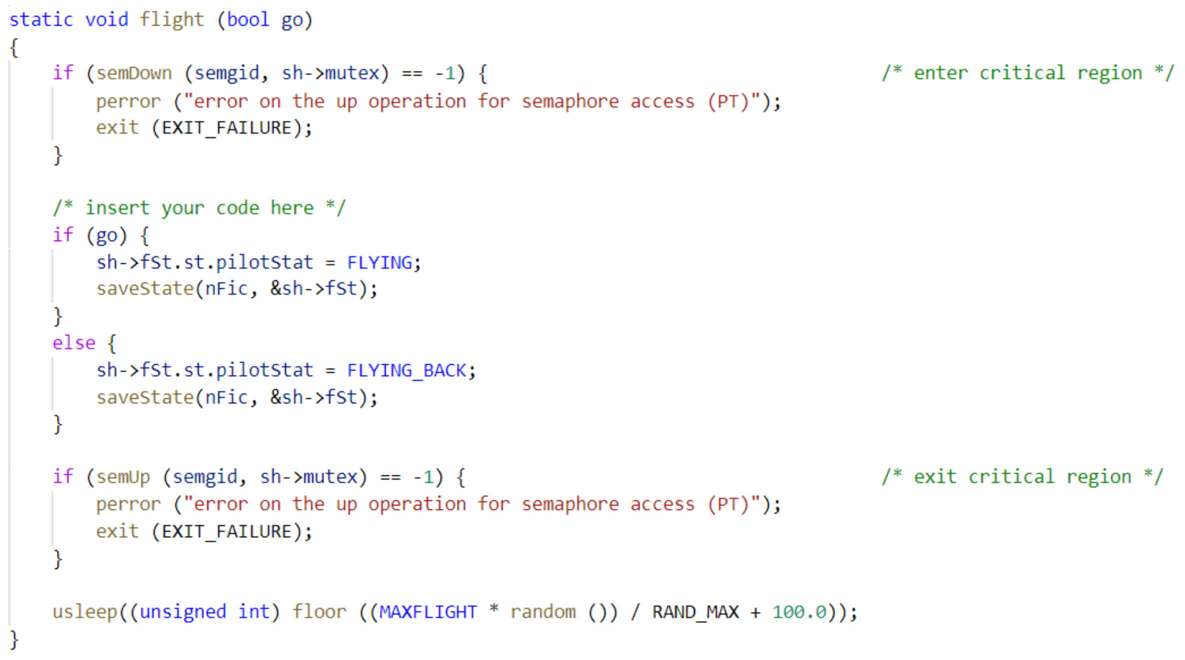
```
/* Pilot state constants */

/** \brief pilot flying to starting airport */
#define  FLYING_BACK                    0
/** \brief pilot signals ready for boarding */
#define  READY_FOR_BOARDING             1
/** \brief pilot wait for boarding to complete */
#define  WAITING_FOR_BOARDING           2
/** \brief pilot takes passengers to destination */
#define  FLYING                         3
/** \brief pilot drops passengers at destination */
#define  DROPING_PASSENGERS             4
```

Fig 1 - Estados do piloto

2.1.1 Função *flight*

Esta função recebe um argumento do tipo booleano, que indica se o piloto está a fazer um voo de *Origin* para *Target* ou vice-versa.

Com efeito, nesta função apenas fazemos alterações dentro da região crítica, nomeadamente, alterar e guardar o estado do piloto para ***FLYING*** ou ***FLYING_BACK***, consoante o argumento ***go*** seja, respetivamente, verdadeiro ou falso.



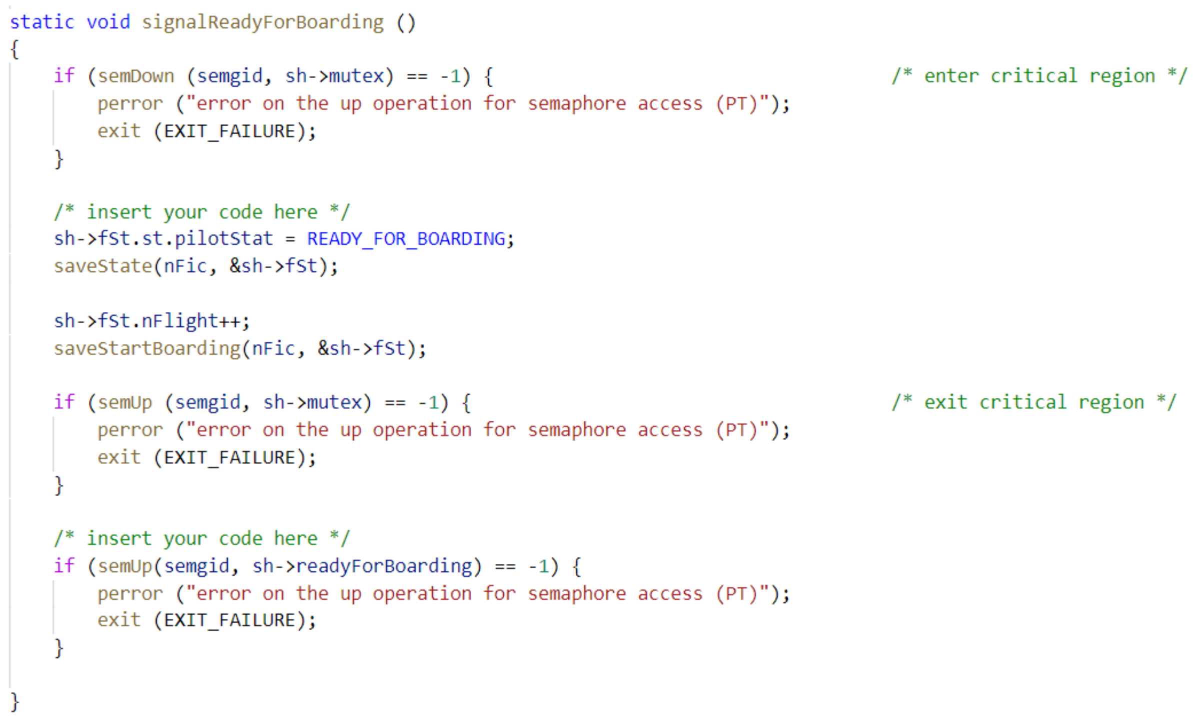
```
static void flight (bool go)
{
    if (semDown (semgid, sh->mutex) == -1) {                                    /* enter critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (go) {
        sh->fSt.st.pilotStat = FLYING;
        saveState(nFic, &sh->fSt);
    }
    else {
        sh->fSt.st.pilotStat = FLYING_BACK;
        saveState(nFic, &sh->fSt);
    }

    if (semUp (semgid, sh->mutex) == -1) {                                      /* exit critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    usleep((unsigned int) floor ((MAXFLIGHT * random ()) / RAND_MAX + 100.0));
}
```

Fig 2 - Função flight

2.1.2 Função *signalReadyForBoarding*

Neste excerto, o piloto sinaliza a hospedeira de que o avião está pronto para o embarque. Assim sendo, dentro da região crítica, alteramos o estado do piloto para ***READY_FOR_BOARDING*** e incrementamos o número do voo, guardando ambos com as funções fornecidas para o efeito.

De seguida, fazemos *up* ao semáforo *readyForBoarding,* o que vai permitir retomar a execução da hospedeira.

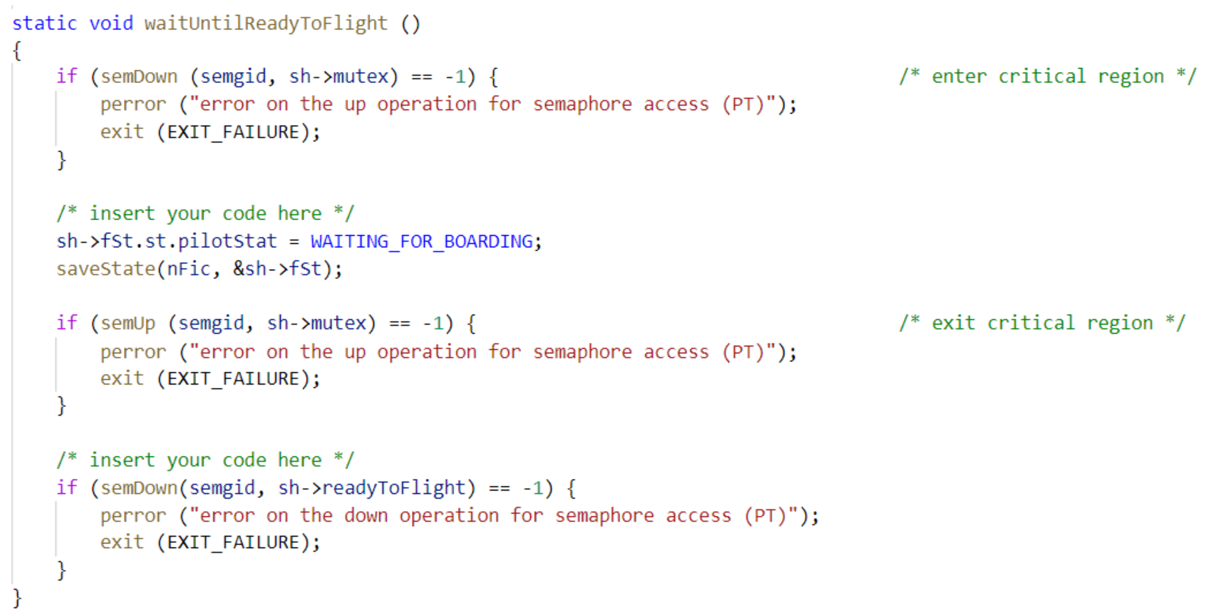


```c
static void signalReadyForBoarding ()
{
    if (semDown (semgid, sh->mutex) == -1) {                                  /* enter critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.st.pilotStat = READY_FOR_BOARDING;
    saveState(nFic, &sh->fSt);

    sh->fSt.nFlight++;
    saveStartBoarding(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                                    /* exit critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semUp(semgid, sh->readyForBoarding) == -1) {
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

}
```

Fig 3 - Função SignalReadyForBoarding

2.1.3 Função *waitUntilReadyToFlight*

Nesta função, o piloto aguarda que o avião esteja pronto para descolar. Por isso, dentro da região crítica, o estado do piloto passa para ***WAITING_FOR_BOARDING*** e fora desta o semáforo *readyToFlight* fica em down, o que representa a espera do interveniente.



```c
static void waitUntilReadyToFlight ()
{
    if (semDown (semgid, sh->mutex) == -1) {                          /* enter critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.st.pilotStat = WAITING_FOR_BOARDING;
    saveState(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                            /* exit critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semDown(semgid, sh->readyToFlight) == -1) {
        perror ("error on the down operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }
}
```

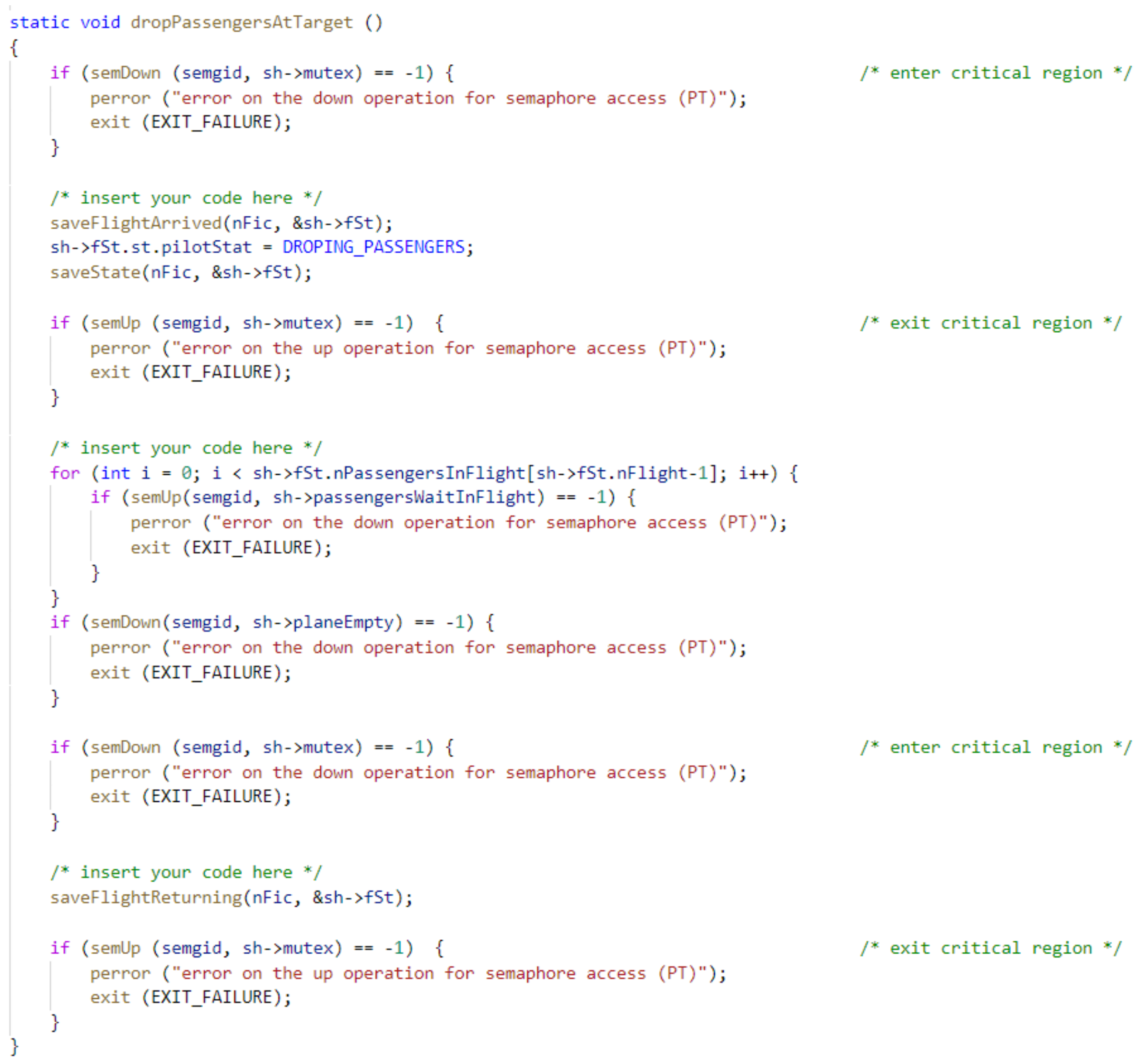
Fig 4 - Função WaitUntilReadyToFlight

2.1.3 Função *dropPassengersAtTarget*

Nesta parte do *script*, é descrito o processo desde que aterram até que todos os passageiros saem do avião. Portanto, na região crítica inicial, imprimimos no terminal a mensagem de que o avião chegou a *Target* e atualizamos o estado do piloto para ***DROPING_PASSENGERS***.

De seguida, num ciclo *for* que é iterado tantas vezes quantos os passageiros no voo atual fazemos *up* do semáforo *passengersWaitInFligth*, o qual os informa de que o voo terminou e podem sair do avião. Após o ciclo, o piloto espera que o último passageiro indique que o avião já está vazio, através do *down* feito ao semáforo *planeEmpty*.

Na segunda região crítica, apenas é chamada a função que imprime a mensagem informativa de que o avião está a regressar a *Origin*.

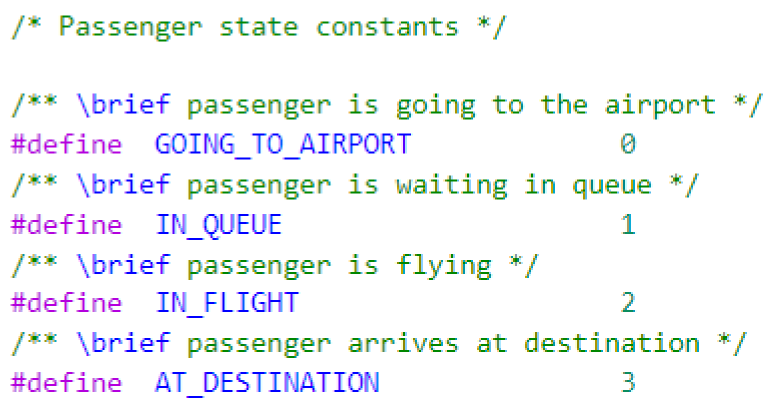


```c
static void dropPassengersAtTarget ()
{
    if (semDown (semgid, sh->mutex) == -1) {                                      /* enter critical region */
        perror ("error on the down operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    saveFlightArrived(nFic, &sh->fSt);
    sh->fSt.st.pilotStat = DROPING_PASSENGERS;
    saveState(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1)  {                                       /* exit critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    for (int i = 0; i < sh->fSt.nPassengersInFlight[sh->fSt.nFlight-1]; i++) {
        if (semUp(semgid, sh->passengersWaitInFlight) == -1) {
            perror ("error on the down operation for semaphore access (PT)");
            exit (EXIT_FAILURE);
        }
    }
    if (semDown(semgid, sh->planeEmpty) == -1) {
        perror ("error on the down operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    if (semDown (semgid, sh->mutex) == -1) {                                      /* enter critical region */
        perror ("error on the down operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    saveFlightReturning(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1)  {                                       /* exit critical region */
        perror ("error on the up operation for semaphore access (PT)");
        exit (EXIT_FAILURE);
    }
}
```

Fig 5 - Função dropPassengersAtTarget

2.2 Passageiro

Outro dos intervenientes da nossa simulação da ponte aérea, é o passageiro, cujos quatro estados possíveis são:



```
/* Passenger state constants */

/** \brief passenger is going to the airport */
#define  GOING_TO_AIRPORT             0
/** \brief passenger is waiting in queue */
#define  IN_QUEUE                     1
/** \brief passenger is flying */
#define  IN_FLIGHT                    2
/** \brief passenger arrives at destination */
#define  AT_DESTINATION               3
```

Fig 6 - Estados do Passageiro

2.2.1 Função *waitInQueue*

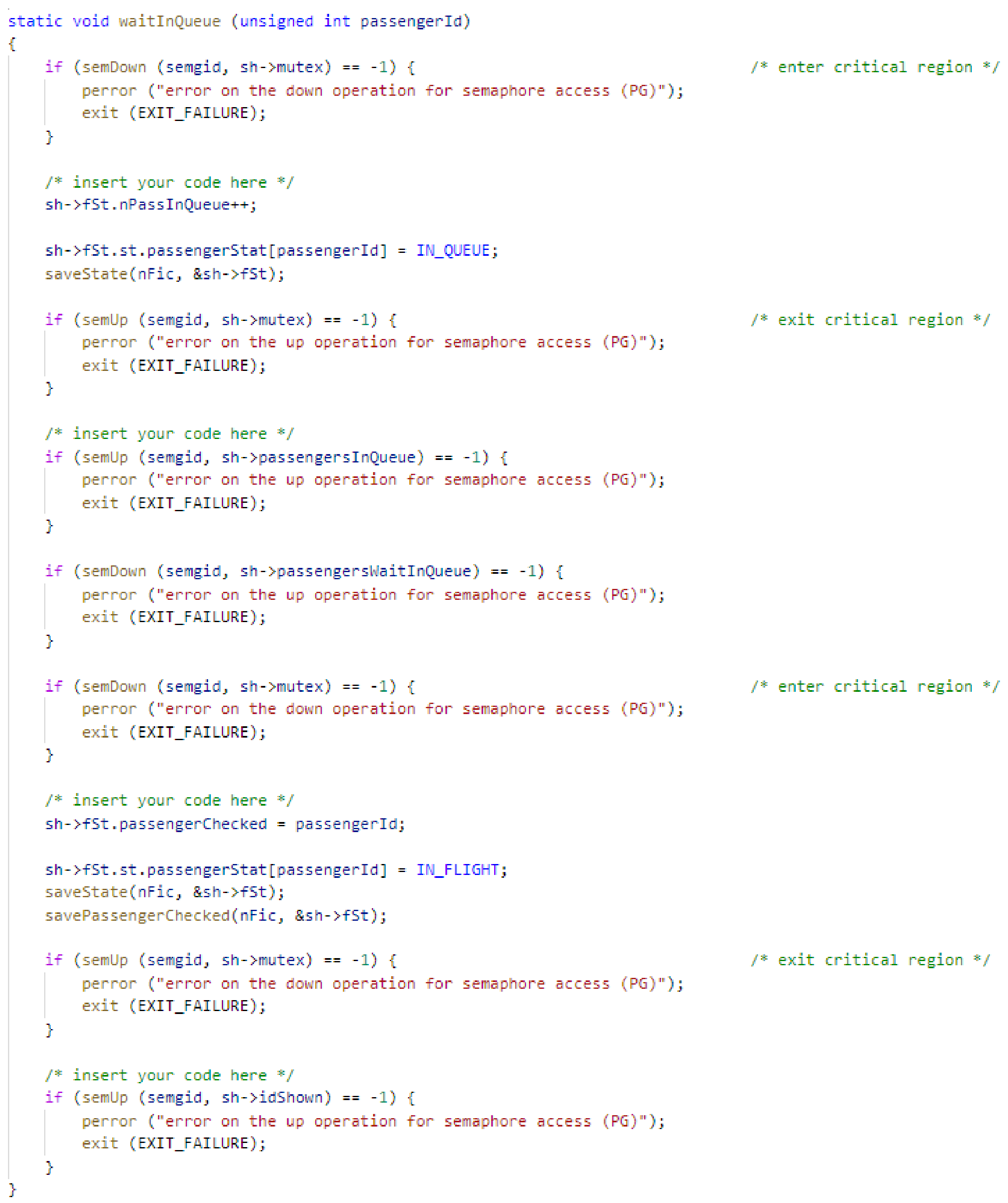
Nesta função, é descrito o processo desde que o passageiro chega à fila de espera até que entra no avião.

Assim, começamos por, dentro da primeira região crítica, incrementar o número de elementos na fila e alterar o estado do passageiro atual, cujo id é dado como argumento da função, para ***IN_QUEUE***.

Seguidamente, o semáforo *passengersInQueue* passa a *up*, permitindo continuar a execução da hospedeira, e damos *down* ao *passengersWaitInQueue*, que coloca o passageiro em espera.

Além disso, na última região crítica, alteramos a variável *passengerChecked* para o id do passageiro atual, já que esta identifica qual o último a ser autorizado a embarcar, e alteramos o estado deste para ***IN_FLIGHT***, imprimindo ambas as alterações.

Por fim, permitimos à hospedeira ver a identificação do passageiro dando up ao semáforo *idShown*.



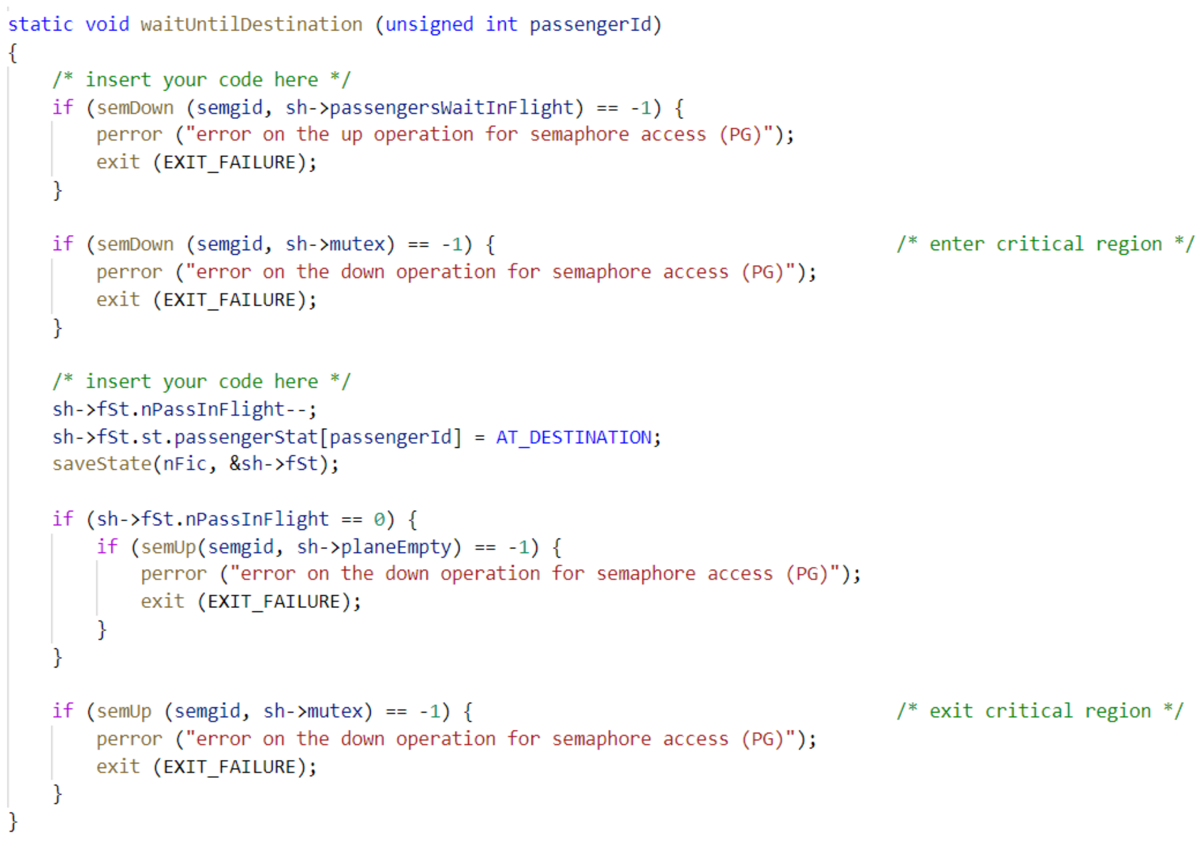
```c
static void waitInQueue (unsigned int passengerId)
{
    if (semDown (semgid, sh->mutex) == -1) {                                   /* enter critical region */
        perror ("error on the down operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.nPassInQueue++;

    sh->fSt.st.passengerStat[passengerId] = IN_QUEUE;
    saveState(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                                     /* exit critical region */
        perror ("error on the up operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semUp (semgid, sh->passengersInQueue) == -1) {
        perror ("error on the up operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    if (semDown (semgid, sh->passengersWaitInQueue) == -1) {
        perror ("error on the up operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    if (semDown (semgid, sh->mutex) == -1) {                                   /* enter critical region */
        perror ("error on the down operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.passengerChecked = passengerId;

    sh->fSt.st.passengerStat[passengerId] = IN_FLIGHT;
    saveState(nFic, &sh->fSt);
    savePassengerChecked(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                                     /* exit critical region */
        perror ("error on the down operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semUp (semgid, sh->idShown) == -1) {
        perror ("error on the up operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }
}
```

Fig 7 - Função WaitInQueue

2.2.2 Função *waitUntilDestination*

A função *waitUntilDestination* simula a espera dos passageiros no voo e o desembarque. Portanto, a primeira ação é pôr os passageiros em espera dando *down* ao semáforo *passengersWaitInFlight*.

Posteriormente, entramos na região crítica e decrementamos o número de pessoas no voo e alteramos o estado do passageiro atual para ***AT_DESTINATION***. Além disso, quando o número de passageiros passa a zero é dado *up* ao semáforo *planeEmpty*, permitindo ao piloto seguir para o voo de regresso.

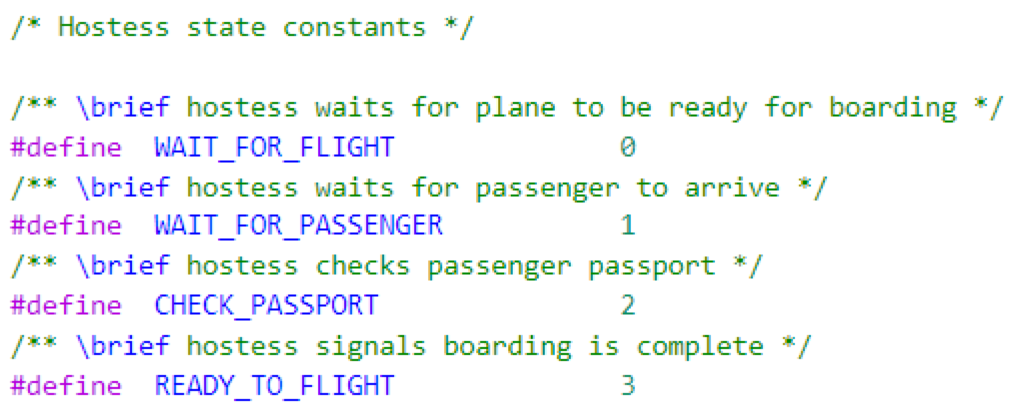


```
static void waitUntilDestination (unsigned int passengerId)
{
    /* insert your code here */
    if (semDown (semgid, sh->passengersWaitInFlight) == -1) {
        perror ("error on the up operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    if (semDown (semgid, sh->mutex) == -1) {                          /* enter critical region */
        perror ("error on the down operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.nPassInFlight--;
    sh->fSt.st.passengerStat[passengerId] = AT_DESTINATION;
    saveState(nFic, &sh->fSt);

    if (sh->fSt.nPassInFlight == 0) {
        if (semUp(semgid, sh->planeEmpty) == -1) {
            perror ("error on the down operation for semaphore access (PG)");
            exit (EXIT_FAILURE);
        }
    }

    if (semUp (semgid, sh->mutex) == -1) {                            /* exit critical region */
        perror ("error on the down operation for semaphore access (PG)");
        exit (EXIT_FAILURE);
    }
}
```

Fig 8 - Função WaitUntilDestination

2.3 Hospedeira

Por fim, temos como entidade principal da ponte aérea a hospedeira, responsável pela maior parte da sincronização da simulação. Pode assumir os seguintes estados:

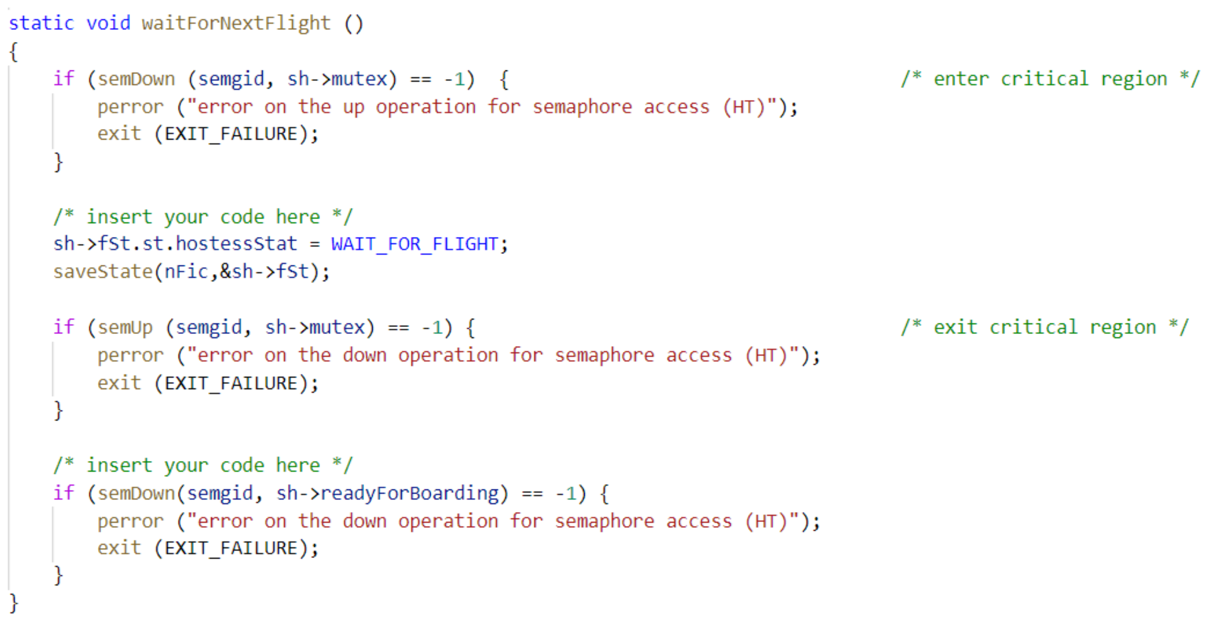


```
/* Hostess state constants */

/** \brief hostess waits for plane to be ready for boarding */
#define  WAIT_FOR_FLIGHT              0
/** \brief hostess waits for passenger to arrive */
#define  WAIT_FOR_PASSENGER           1
/** \brief hostess checks passenger passport */
#define  CHECK_PASSPORT               2
/** \brief hostess signals boarding is complete */
#define  READY_TO_FLIGHT              3
```

Fig 9 – Estados hospedeira

2.3.1 Função *waitForNextFlight*

A função *waitUntilNextFlight* simula o momento em que a hospedeira espera que o avião retorne do voo anterior. Logo, a primeira coisa a fazer é colocar o estado da hospedeira como ***WAIT_FOR_NEXT_FLIGHT*** e depois colocar *down* o semáforo *readyForBoarding*.



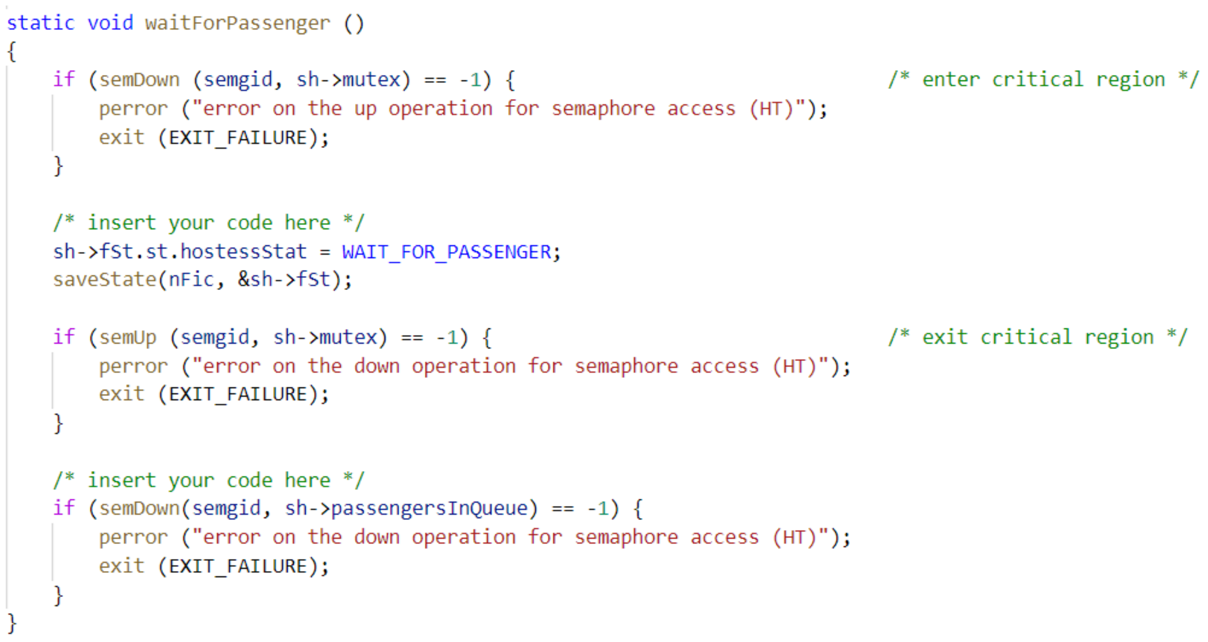
```
static void waitForNextFlight ()
{
    if (semDown (semgid, sh->mutex) == -1)  {                  /* enter critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.st.hostessStat = WAIT_FOR_FLIGHT;
    saveState(nFic,&sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                     /* exit critical region */
        perror ("error on the down operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semDown(semgid, sh->readyForBoarding) == -1) {
        perror ("error on the down operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }
}
```

Fig 10 - Função WaitForNextFlight

2.3.2 Função *waitForPassenger*

Nesta função, a hospedeira é colocada à espera dos passageiros na fila, por isso o seu estado é alterado para ***WAIT_FOR_PASSENGER***, dentro da região crítica.

Também é colocado o semáforo *passengersInQueue* em *down*, deixando a hospedeira em espera.



```c
static void waitForPassenger ()
{
    if (semDown (semgid, sh->mutex) == -1) {                /* enter critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.st.hostessStat = WAIT_FOR_PASSENGER;
    saveState(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                  /* exit critical region */
        perror ("error on the down operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semDown(semgid, sh->passengersInQueue) == -1) {
        perror ("error on the down operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }
}
```

Fig 11 - Função WaitForPassenger

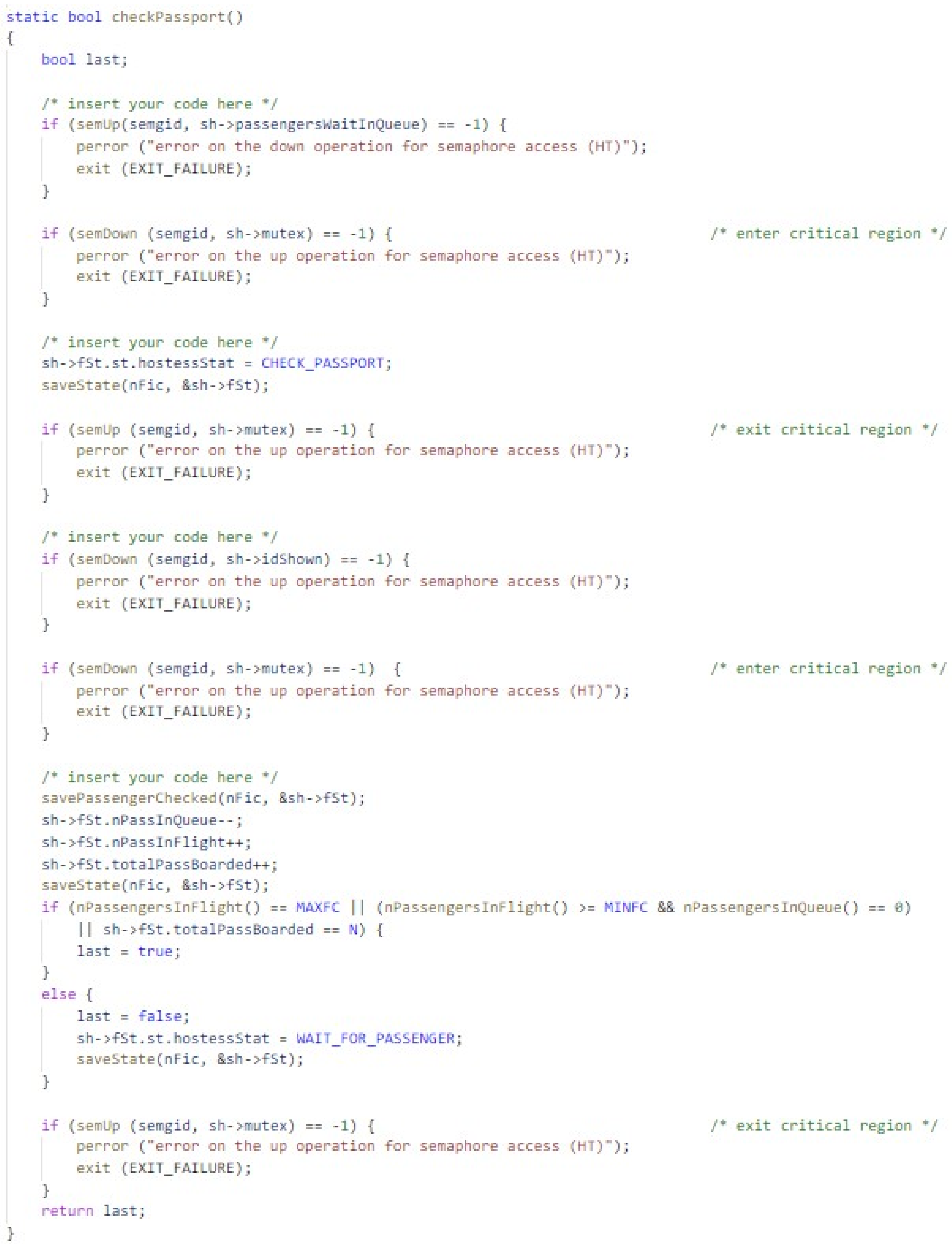
2.3.3 Função *checkPassport*

Nesta função começamos por dar *up* ao semáforo *passengersWaitInQueue*, que permite ao primeiro passageiro da fila avançar.

De seguida, dentro da região crítica alteramos o estado da hospedeira para ***CHECK_PASSPORT*** e fora desta passamos o semáforo *idShown* a *down*.

Na segunda região crítica, é imprimido no terminal o número do passageiro verificado, decrementado o número de passageiros na fila e incrementada a quantidade de pessoas no voo atual e no total de passageiros que já embarcaram em todos os voos.

Além disso, caso o limite máximo de passageiros por voo seja atingido ou tivermos pelo menos o mínimo necessário e não houver nenhum na fila ou ainda se já todos tiverem embarcado o valor de ***last*** passa a *true*, significando que o avião está pronto para partir. Se, por outro lado, nenhuma destas opções se verificar o valor de ***last*** é *false* e o estado da hospedeira é alterado para ***WAIT_FOR_PASSENGER***.



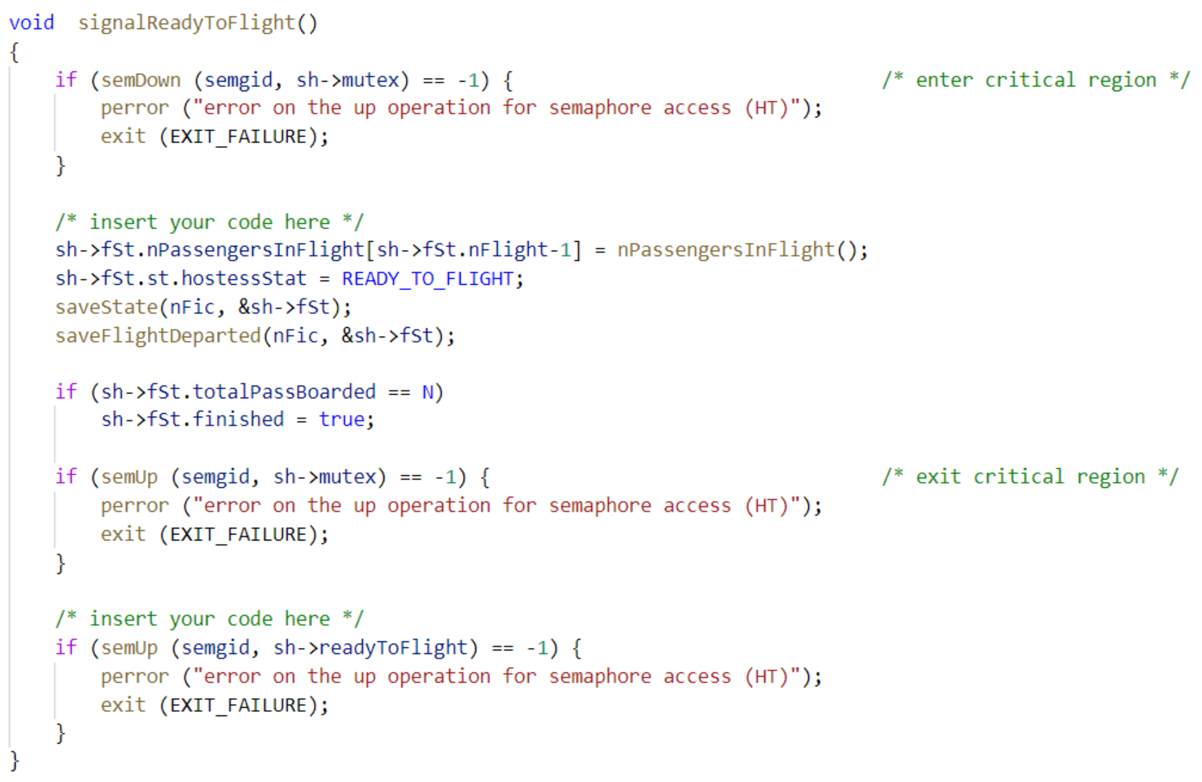
```c
static bool checkPassport()
{
    bool last;

    /* insert your code here */
    if (semUp(semgid, sh->passengersWaitInQueue) == -1) {
        perror ("error on the down operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.st.hostessStat = CHECK_PASSPORT;
    saveState(nFic, &sh->fSt);

    if (semUp (semgid, sh->mutex) == -1) {                                  /* exit critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semDown (semgid, sh->idShown) == -1) {
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    if (semDown (semgid, sh->mutex) == -1)  {                               /* enter critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    savePassengerChecked(nFic, &sh->fSt);
    sh->fSt.nPassInQueue--;
    sh->fSt.nPassInFlight++;
    sh->fSt.totalPassBoarded++;
    saveState(nFic, &sh->fSt);
    if (nPassengersInFlight() == MAXFC || (nPassengersInFlight() >= MINFC && nPassengersInQueue() == 0)
        || sh->fSt.totalPassBoarded == N) {
        last = true;
    }
    else {
        last = false;
        sh->fSt.st.hostessStat = WAIT_FOR_PASSENGER;
        saveState(nFic, &sh->fSt);
    }

    if (semUp (semgid, sh->mutex) == -1) {                                  /* exit critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }
    return last;
}
```

Fig 12 - Função CheckPassport

2.3.4 Função *signalReadyToFlight*

A função *signalReadyToFlight* prepara a hospedeira para avisar o piloto de que o avião está pronto para partir. Começamos por guardar o número de passageiros dentro do voo atual (*nPassangersInFlight*) e o estado da hospedeira para ***READY_TO_FLIGHT***. Quando o total de passageiros que já embarcaram em todos os voos iguala o número de pessoas que pretendemos deslocar a variável ***finished*** passa a ter o valor *true*, indicando que este é o último voo a ser feito.

Finalmente, damos *up* ao semáforo *readyToFlight* que indica ao piloto que pode descolar.



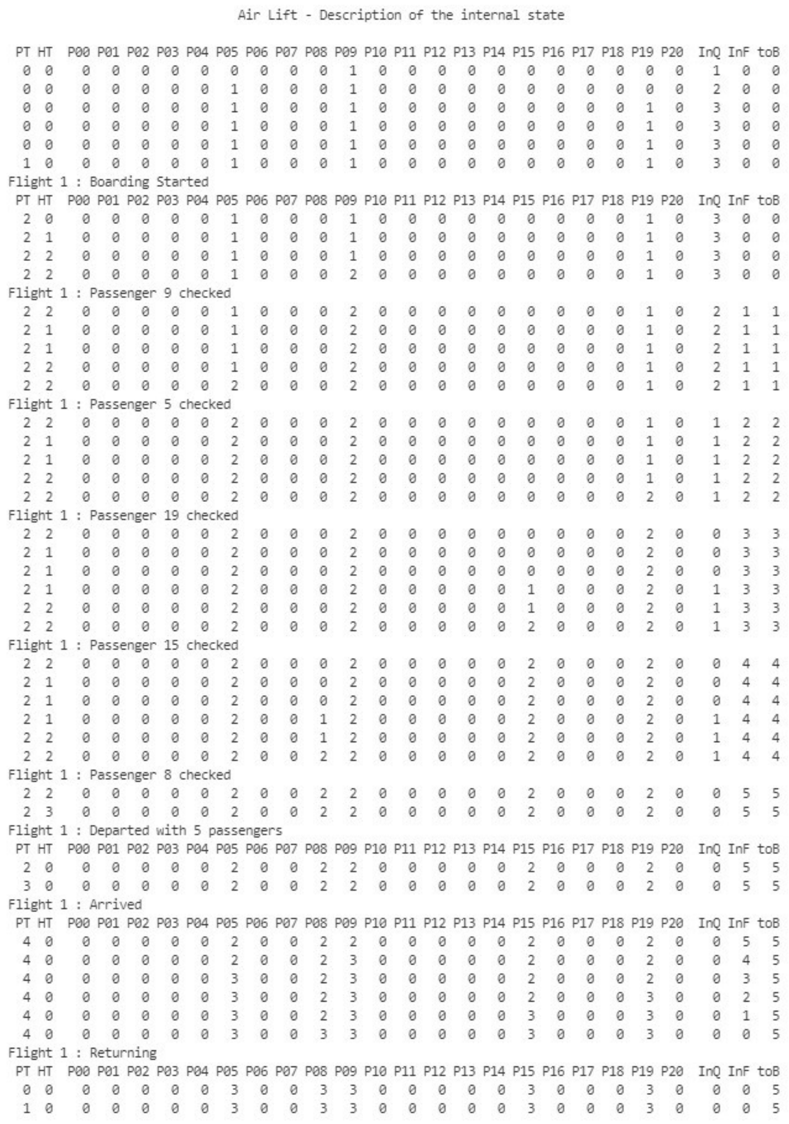
```
void  signalReadyToFlight()
{
    if (semDown (semgid, sh->mutex) == -1) {                       /* enter critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    sh->fSt.nPassengersInFlight[sh->fSt.nFlight-1] = nPassengersInFlight();
    sh->fSt.st.hostessStat = READY_TO_FLIGHT;
    saveState(nFic, &sh->fSt);
    saveFlightDeparted(nFic, &sh->fSt);

    if (sh->fSt.totalPassBoarded == N)
        sh->fSt.finished = true;

    if (semUp (semgid, sh->mutex) == -1) {                         /* exit critical region */
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

    /* insert your code here */
    if (semUp (semgid, sh->readyToFlight) == -1) {
        perror ("error on the up operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }
}
```

Fig 13 - Função signalReadyToFlight

3 Resultados

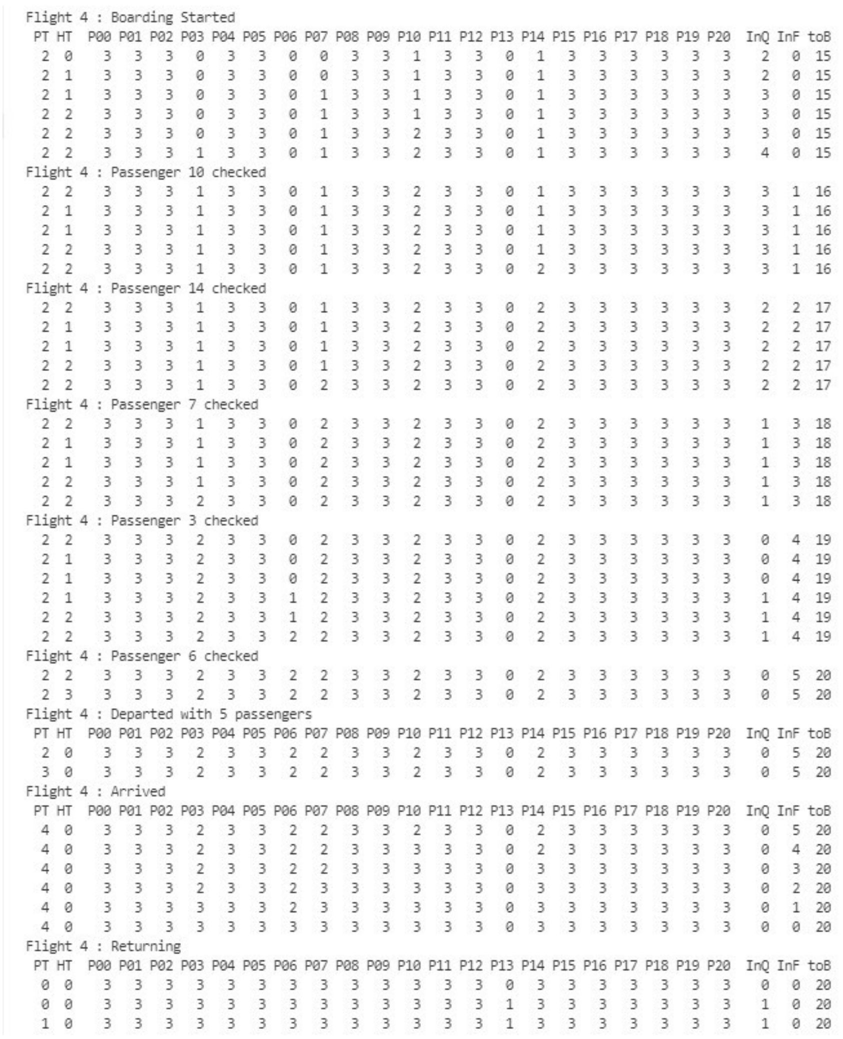
Ao longo da implementação, para verificar se o que estávamos a fazer, compilámos o código à vez com apenas um dos intervenientes feitos por nós, utilizando para os outros dois o código pré-compilado fornecido pelo professor. Para isso utilizámos os comandos *make pt*, *make pg* e *make ht*. Com isto foi nos possível identificar mais facilmente a ocorrência de deadlocks e as falhas da nossa implementação.

Após resolvermos todos os problemas que encontrámos executámos o comando *make all* e obtivemos o seguinte resultado.

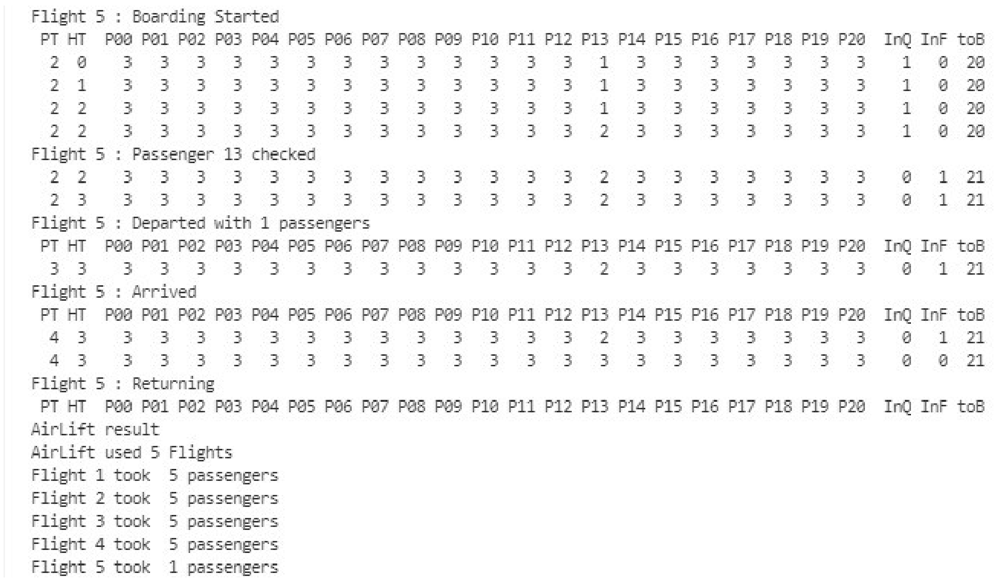


```
                         Air Lift - Description of the internal state

 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  0  0    0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0    1   0   0
  0  0    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0    2   0   0
  0  0    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
  0  0    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
  0  0    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
  1  0    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
Flight 1 : Boarding Started
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2  0    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
  2  1    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
  2  2    0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0    3   0   0
  2  2    0   0   0   0   0   1   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    3   0   0
Flight 1 : Passenger 9 checked
  2  2    0   0   0   0   0   1   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    2   1   1
  2  1    0   0   0   0   0   1   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    2   1   1
  2  1    0   0   0   0   0   1   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    2   1   1
  2  2    0   0   0   0   0   1   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    2   1   1
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    2   1   1
Flight 1 : Passenger 5 checked
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    1   2   2
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    1   2   2
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    1   2   2
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   1   0    1   2   2
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   2   0    1   2   2
Flight 1 : Passenger 19 checked
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   2   0    0   3   3
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   2   0    0   3   3
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   0   0   0   0   2   0    0   3   3
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   1   0   0   0   2   0    1   3   3
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   1   0   0   0   2   0    1   3   3
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   2   0   0   0   2   0    1   3   3
Flight 1 : Passenger 15 checked
  2  2    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   2   0   0   0   2   0    0   4   4
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   2   0   0   0   2   0    0   4   4
  2  1    0   0   0   0   0   2   0   0   0   2   0   0   0   0   0   2   0   0   0   2   0    0   4   4
  2  1    0   0   0   0   0   2   0   0   1   2   0   0   0   0   0   2   0   0   0   2   0    1   4   4
  2  2    0   0   0   0   0   2   0   0   1   2   0   0   0   0   0   2   0   0   0   2   0    1   4   4
  2  2    0   0   0   0   0   2   0   0   2   2   0   0   0   0   0   2   0   0   0   2   0    1   4   4
Flight 1 : Passenger 8 checked
  2  2    0   0   0   0   0   2   0   0   2   2   0   0   0   0   0   2   0   0   0   2   0    0   5   5
  2  3    0   0   0   0   0   2   0   0   2   2   0   0   0   0   0   2   0   0   0   2   0    0   5   5
Flight 1 : Departed with 5 passengers
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2  0    0   0   0   0   0   2   0   0   2   2   0   0   0   0   0   2   0   0   0   2   0    0   5   5
  3  0    0   0   0   0   0   2   0   0   2   2   0   0   0   0   0   2   0   0   0   2   0    0   5   5
Flight 1 : Arrived
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  4  0    0   0   0   0   0   2   0   0   2   2   0   0   0   0   0   2   0   0   0   2   0    0   5   5
  4  0    0   0   0   0   0   2   0   0   2   3   0   0   0   0   0   2   0   0   0   2   0    0   4   5
  4  0    0   0   0   0   0   3   0   0   2   3   0   0   0   0   0   2   0   0   0   2   0    0   3   5
  4  0    0   0   0   0   0   3   0   0   2   3   0   0   0   0   0   2   0   0   0   3   0    0   2   5
  4  0    0   0   0   0   0   3   0   0   2   3   0   0   0   0   0   3   0   0   0   3   0    0   1   5
  4  0    0   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   0   5
Flight 1 : Returning
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  0  0    0   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   0   5
  1  0    0   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   0   5
```

```
Flight 2 : Boarding Started
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2 0     0   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   0   5
  2 1     0   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   0   5
  2 1     1   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   0   5
  2 2     1   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   0   5
  2 2     2   0   0   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   0   5
  2 2     2   0   1   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    2   0   5
Flight 2 : Passenger 0 checked
  2 2     2   0   1   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   1   6
  2 1     2   0   1   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   1   6
  2 1     2   0   1   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   1   6
  2 2     2   0   1   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   1   6
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    1   1   6
Flight 2 : Passenger 2 checked
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   2   7
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   2   7
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   0   0   0   3   0   0   0   3   0    0   2   7
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   1   0   0   3   0   0   0   3   0    1   2   7
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   1   0   0   3   0   0   0   3   0    1   2   7
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   0    1   2   7
Flight 2 : Passenger 12 checked
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   0    0   3   8
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   0    0   3   8
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   0    0   3   8
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   1    1   3   8
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   1    1   3   8
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   0   0   3   2    1   3   8
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   1   0   3   2    2   3   8
Flight 2 : Passenger 20 checked
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   1   0   3   2    1   4   9
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   1   0   3   2    1   4   9
  2 1     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   1   0   3   2    1   4   9
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   1   0   3   2    1   4   9
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    1   4   9
Flight 2 : Passenger 17 checked
  2 2     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    0   5  10
  2 3     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    0   5  10
Flight 2 : Departed with 5 passengers
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2 0     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    0   5  10
  3 0     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    0   5  10
Flight 2 : Arrived
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  4 0     2   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    0   5  10
  4 0     3   0   2   0   0   3   0   0   3   3   0   0   2   0   0   3   0   2   0   3   2    0   4  10
  4 0     3   0   2   0   0   3   0   0   3   3   0   0   3   0   0   3   0   2   0   3   2    0   3  10
  4 0     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   2   0   3   2    0   2  10
  4 0     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   2   0   3   3    0   1  10
  4 0     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   3   0   3   3    0   0  10
Flight 2 : Returning
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  0 0     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   3   0   3   3    0   0  10
  1 0     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   3   0   3   3    0   0  10

 Flight 3 : Boarding Started
  PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
   2 0     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   3   0   3   3    0   0  10
   2 1     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   0   3   0   3   3    0   0  10
   2 1     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   1   3   0   3   3    1   0  10
   2 2     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   1   3   0   3   3    1   0  10
   2 2     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    1   0  10
 Flight 3 : Passenger 16 checked
   2 2     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    0   1  11
   2 1     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    0   1  11
   2 1     3   0   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    0   1  11
   2 1     3   1   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    1   1  11
   2 2     3   1   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    1   1  11
   2 2     3   2   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    1   1  11
 Flight 3 : Passenger 1 checked
   2 2     3   2   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    0   2  12
   2 1     3   2   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    0   2  12
   2 1     3   2   3   0   0   3   0   0   3   3   0   0   3   0   0   3   2   3   0   3   3    0   2  12
   2 1     3   2   3   0   0   3   0   0   3   3   0   1   3   0   0   3   2   3   0   3   3    1   2  12
   2 2     3   2   3   0   0   3   0   0   3   3   0   1   3   0   0   3   2   3   0   3   3    1   2  12
   2 2     3   2   3   0   0   3   0   0   3   3   0   2   3   0   0   3   2   3   0   3   3    1   2  12
 Flight 3 : Passenger 11 checked
   2 2     3   2   3   0   0   3   0   0   3   3   0   2   3   0   0   3   2   3   0   3   3    0   3  13
   2 1     3   2   3   0   0   3   0   0   3   3   0   2   3   0   0   3   2   3   0   3   3    0   3  13
   2 1     3   2   3   0   0   3   0   0   3   3   0   2   3   0   0   3   2   3   0   3   3    0   3  13
   2 1     3   2   3   0   1   3   0   0   3   3   0   2   3   0   0   3   2   3   0   3   3    1   3  13
   2 1     3   2   3   0   1   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    2   3  13
   2 2     3   2   3   0   1   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    2   3  13
   2 2     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    2   3  13
 Flight 3 : Passenger 4 checked
   2 2     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    1   4  14
   2 1     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    1   4  14
   2 1     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    1   4  14
   2 2     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   1   3   3    1   4  14
   2 2     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   2   3   3    1   4  14
 Flight 3 : Passenger 18 checked
   2 2     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   2   3   3    0   5  15
   2 3     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   2   3   3    0   5  15
 Flight 3 : Departed with 5 passengers
  PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
   2 0     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   2   3   3    0   5  15
   3 0     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   2   3   3    0   5  15
 Flight 3 : Arrived
  PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
   4 0     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   2   3   2   3   3    0   5  15
   4 0     3   2   3   0   2   3   0   0   3   3   0   2   3   0   0   3   3   3   2   3   3    0   4  15
   4 0     3   3   3   0   2   3   0   0   3   3   0   2   3   0   0   3   3   3   2   3   3    0   3  15
   4 0     3   3   3   0   2   3   0   0   3   3   0   3   3   0   0   3   3   3   2   3   3    0   2  15
   4 0     3   3   3   0   3   3   0   0   3   3   0   3   3   0   0   3   3   3   2   3   3    0   1  15
   4 0     3   3   3   0   3   3   0   0   3   3   0   3   3   0   0   3   3   3   3   3   3    0   0  15
 Flight 3 : Returning
  PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
   0 0     3   3   3   0   3   3   0   0   3   3   0   3   3   0   0   3   3   3   3   3   3    0   0  15
   0 0     3   3   3   0   3   3   0   0   3   3   1   3   3   0   0   3   3   3   3   3   3    1   0  15
   0 0     3   3   3   0   3   3   0   0   3   3   1   3   3   0   1   3   3   3   3   3   3    2   0  15
   1 0     3   3   3   0   3   3   0   0   3   3   1   3   3   0   1   3   3   3   3   3   3    2   0  15
```



```
Flight 4 : Boarding Started
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2  0    3   3   3   0   3   3   0   0   3   3   1   3   3   0   1   3   3   3   3   3   3    2   0  15
  2  1    3   3   3   0   3   3   0   0   3   3   1   3   3   0   1   3   3   3   3   3   3    2   0  15
  2  1    3   3   3   0   3   3   0   1   3   3   1   3   3   0   1   3   3   3   3   3   3    3   0  15
  2  2    3   3   3   0   3   3   0   1   3   3   1   3   3   0   1   3   3   3   3   3   3    3   0  15
  2  2    3   3   3   0   3   3   0   1   3   3   2   3   3   0   1   3   3   3   3   3   3    3   0  15
  2  2    3   3   3   1   3   3   0   1   3   3   2   3   3   0   1   3   3   3   3   3   3    4   0  15
Flight 4 : Passenger 10 checked
  2  2    3   3   3   1   3   3   0   1   3   3   2   3   3   0   1   3   3   3   3   3   3    3   1  16
  2  1    3   3   3   1   3   3   0   1   3   3   2   3   3   0   1   3   3   3   3   3   3    3   1  16
  2  1    3   3   3   1   3   3   0   1   3   3   2   3   3   0   1   3   3   3   3   3   3    3   1  16
  2  2    3   3   3   1   3   3   0   1   3   3   2   3   3   0   1   3   3   3   3   3   3    3   1  16
  2  2    3   3   3   1   3   3   0   1   3   3   2   3   3   0   2   3   3   3   3   3   3    3   1  16
Flight 4 : Passenger 14 checked
  2  2    3   3   3   1   3   3   0   1   3   3   2   3   3   0   2   3   3   3   3   3   3    2   2  17
  2  1    3   3   3   1   3   3   0   1   3   3   2   3   3   0   2   3   3   3   3   3   3    2   2  17
  2  1    3   3   3   1   3   3   0   1   3   3   2   3   3   0   2   3   3   3   3   3   3    2   2  17
  2  2    3   3   3   1   3   3   0   1   3   3   2   3   3   0   2   3   3   3   3   3   3    2   2  17
  2  2    3   3   3   1   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    2   2  17
Flight 4 : Passenger 7 checked
  2  2    3   3   3   1   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   3  18
  2  1    3   3   3   1   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   3  18
  2  1    3   3   3   1   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   3  18
  2  2    3   3   3   1   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   3  18
  2  2    3   3   3   2   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   3  18
Flight 4 : Passenger 3 checked
  2  2    3   3   3   2   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   4  19
  2  1    3   3   3   2   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   4  19
  2  1    3   3   3   2   3   3   0   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   4  19
  2  1    3   3   3   2   3   3   1   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   4  19
  2  2    3   3   3   2   3   3   1   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   4  19
  2  2    3   3   3   2   3   3   2   2   3   3   2   3   3   0   2   3   3   3   3   3   3    1   4  19
Flight 4 : Passenger 6 checked
  2  2    3   3   3   2   3   3   2   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   5  20
  2  3    3   3   3   2   3   3   2   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   5  20
Flight 4 : Departed with 5 passengers
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2  0    3   3   3   2   3   3   2   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   5  20
  3  0    3   3   3   2   3   3   2   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   5  20
Flight 4 : Arrived
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  4  0    3   3   3   2   3   3   2   2   3   3   2   3   3   0   2   3   3   3   3   3   3    0   5  20
  4  0    3   3   3   2   3   3   2   2   3   3   3   3   3   0   2   3   3   3   3   3   3    0   4  20
  4  0    3   3   3   2   3   3   2   2   3   3   3   3   3   0   3   3   3   3   3   3   3    0   3  20
  4  0    3   3   3   2   3   3   2   3   3   3   3   3   3   0   3   3   3   3   3   3   3    0   2  20
  4  0    3   3   3   3   3   3   2   3   3   3   3   3   3   0   3   3   3   3   3   3   3    0   1  20
  4  0    3   3   3   3   3   3   3   3   3   3   3   3   3   0   3   3   3   3   3   3   3    0   0  20
Flight 4 : Returning
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  0  0    3   3   3   3   3   3   3   3   3   3   3   3   3   0   3   3   3   3   3   3   3    0   0  20
  0  0    3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3   3   3   3   3   3    1   0  20
  1  0    3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3   3   3   3   3   3    1   0  20
```



```
Flight 5 : Boarding Started
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  2  0    3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3   3   3   3   3   3    1   0  20
  2  1    3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3   3   3   3   3   3    1   0  20
  2  2    3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3   3   3   3   3   3    1   0  20
  2  2    3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3   3   3   3   3   3    1   0  20
Flight 5 : Passenger 13 checked
  2  2    3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3   3   3   3   3   3    0   1  21
  2  3    3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3   3   3   3   3   3    0   1  21
Flight 5 : Departed with 1 passengers
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  3  3    3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3   3   3   3   3   3    0   1  21
Flight 5 : Arrived
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
  4  3    3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3   3   3   3   3   3    0   1  21
  4  3    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3    0   0  21
Flight 5 : Returning
 PT HT  P00 P01 P02 P03 P04 P05 P06 P07 P08 P09 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  InQ InF toB
AirLift result
AirLift used 5 Flights
Flight 1 took  5 passengers
Flight 2 took  5 passengers
Flight 3 took  5 passengers
Flight 4 took  5 passengers
Flight 5 took  1 passengers
```

Fig 14 – Resultado da execução do código

Com a implementação descrita obtivemos o seguinte quadro no qual identificamos para cada semáforo utilizado qual a entidade responsável pelos ups e downs, assim como as respetivas funções e quantidades.

Semáforo	*Entidade Down*	*Função Down*	*#Down*	*Entidade Up*	*Função Up*	*#Up*
passengersInQueue	hostess	waitForPassenger	21	passengers	waitInQueue	21
passengersWaitInQueue	passengers	waitInQueue	21	hostess	checkPassport	21
passengersWaitInFlight	passengers	waitUntilDestination	21	pilot	dropPassengersAtTarget	21
readyForBoarding	hostess	waitForNextFlight	1 por voo	Pilot	signalReadyForBoarding	1 por voo
readyToFlight	pilot	waitUntilReadyToFlight	1 por voo	hostess	signalReadyToFlight	1 por voo
idShown	hostess	checkPassport	21	passenger	waitInQueue	21

planeEmpty	pilot	dropPassengersAtTarget	1 por voo	passenger	waitUntilDestination	1 por voo

Fig 15 – Tabela dos semáforos

4 Conclusão

Os scripts desenvolvidos vão de encontro ao proposto, permitindo simular com sucesso vários voos sem que se verifiquem deadlocks durante as ações de cada interveniente.

Com efeito, a realização deste trabalho contribuiu positivamente para o aprofundamento dos nossos conhecimentos no uso de semáforos, através da pesquisa que exigiu, desde com se deve usar corretamente os semáforos e como realizar ações na região crítica. Além disso, temos agora uma melhor noção do que é a shared memory e da sua importância nos scripts.

Durante a execução do trabalho tivemos de lidar com algumas dificuldades, nomeadamente perceber quando ativar e desativar os semáforos, isto é, acertar a sincronização das ações de cada interveniente, mas com recurso aos guiões práticos e alguns sites e de algumas tentativas-erros, conseguimos chegar ao resultado pretendido.

5 Bibliografia

Para a realização deste trabalho consultamos os slides teóricos e os guiões práticos disponibilizados pelo docente, assim como os seguintes sites (consultados entre os dias 21/01/22 e 25/01/22):

- https://stackoverflow.com/
- https://www.geeksforgeeks.org/