universidade de aveiro

LICENCIATURA EM ENGENHARIA INFORMÁTICA
SISTEMAS OPERATIVOS
2021/2022

Simulação de Ponte Aérea

Turma P3

Catarina Alves Marques | 81382

Victor Barros Melo | 101099

Índice

Introdução

O presente relatório refere-se ao segundo trabalho prático proposto na disciplina de Sistemas Operativos que tinha como objectivo o desenvolvimento de uma aplicação em C para a simulação de uma ponte aérea de um avião de forma a transportar *n* passageiros entre as cidades Origin e Target. Para poder descolar, há certas condições a ter em consideração: o avião tem o número mínimo de passageiros e não há mais ninguém na fila, o avião está cheio ou os *n* passageiros já embarcaram.

A resolução deste problema pressupõe a compreensão dos mecanismos relacionados à execução e sincronização de processos e threads, nomeadamente, pela necessidade do uso de semáforos e de memória partilhada.

Estrutura do Programa

De modo a facilitar a compreensão do funcionamento da aplicação começou-se por desenhar um diagrama (figura 1) para determinar o acesso à memória através de semáforos, a partir do qual foi criada a tabela 1.

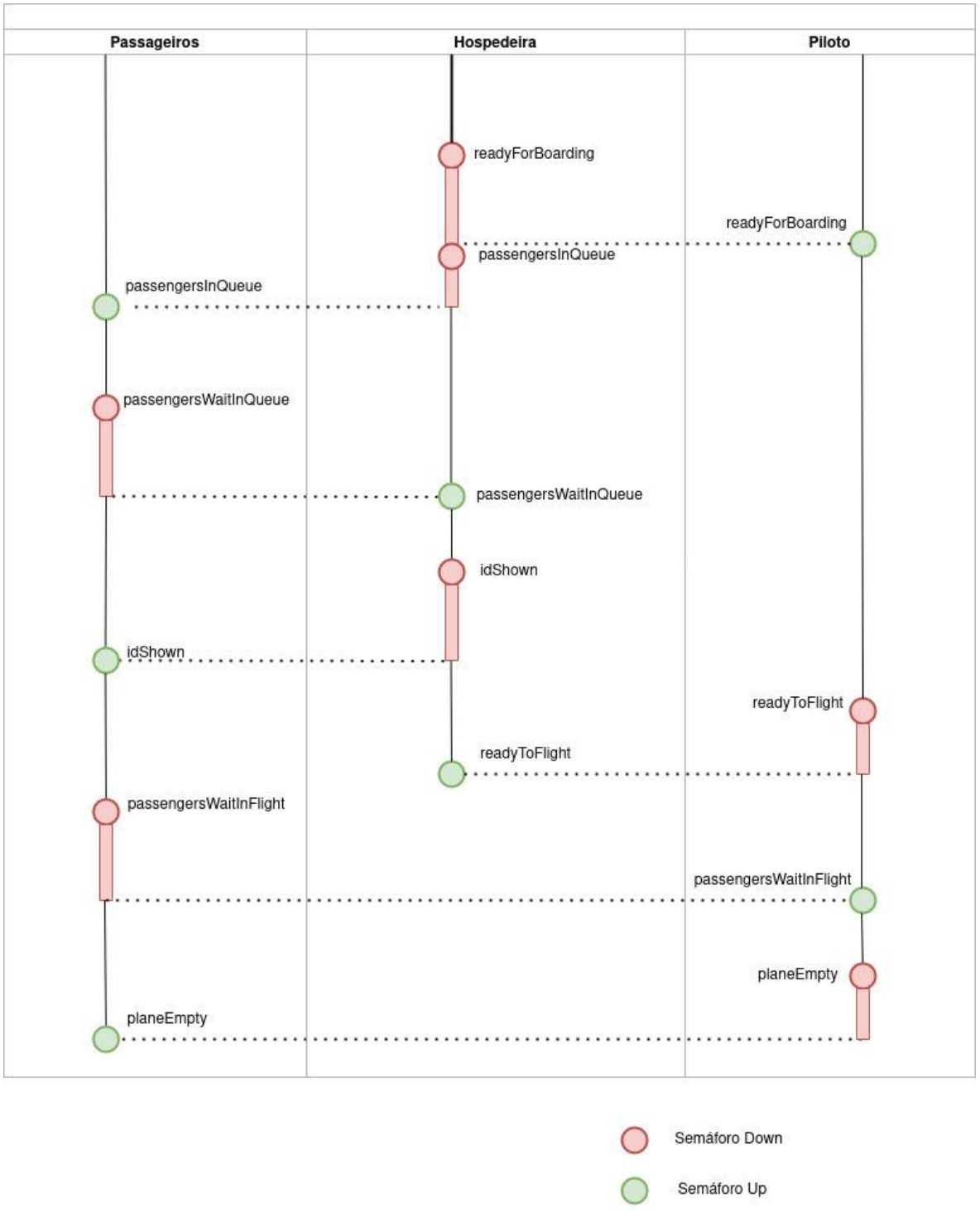

Figura 1 - Diagrama de fluxo da aplicação

- Entidade - DOWN	- Entidade - DOWN	**Semáforo**	- Entidade - UP	- Entidade - UP
HOSPEDEIRA	waitForNextFlight()	**ready For Boarding**	PILOTO	signalReadyForBoarding()
HOSPEDEIRA	waitForPassenger()	**passengers In Queue**	PASSAGEIRO	waitInQueue()
PASSAGEIRO	waitInQueue()	**passengers Wait In Queue**	HOSPEDEIRA	checkPassport()
HOSPEDEIRA	checkPassport()	**id Shown**	PASSAGEIRO	waitInQueue()
PILOTO	waitUntilreadyToFlight()	**ready To Flight**	HOSPEDEIRA	signalReadyToFlight()
PASSAGEIRO	waitUntilDestination()	**passengers Wait in Flight**	PILOTO	dropPassengersAtTarget()
PILOTO	dropPassengersAtTarget()	**planeEmpty**	PASSAGEIRO	waitUntilDestination()

Tabela 1.

Assumindo que no início da simulação o avião se encontra em trânsito para Origin, determinou-se que o ciclo se inicia com a hospedeira a aguardar por um próximo voo.

Nesse sentido o ficheiro *semSharedMemHostess.c* foi o ponto de partida, especificamente a função *waitForNextFlight()*, onde o estado da hospedeira é atualizado para *WAIT_FOR_FLIGHT* e o semáforo *readyForBoarding* passa a Down.

```
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);
   }
```

De seguida, em *semSharedMemPilot.c*, o estado do piloto é atualizado para *READY_FOR_BOARDING*, o número do voo é incrementado, o estado StartBoarding é guardado, e o semáforo readyForBoarding passa a up.

```
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);
   }

   sh->fSt.st.pilotStat = READY_FOR_BOARDING;
   sh->fSt.nFlight++;
   saveState(nFic, &sh->fSt);
   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);
   }
   if (semUp(semgid, sh->readyForBoarding) == -1)
   { // Up ready for boarding
       perror("error on the down operation for semaphore access");
       exit(EXIT_FAILURE);
   }
}
```

Na fase seguinte, a hospedeira aguarda pela chegada dos passageiros (o seu estado é atualizado para *WAIT_FOR_PASSENGER*), e o semáforo *passengersInQueue* passa a down.

```
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);
   }
```

```
        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);
    }


    if (semDown (semgid, sh->passengersInQueue) == -1) {
/* Wait for passengers to get in queue */
        perror ("error on the down operation for semaphore access (HT)");
        exit (EXIT_FAILURE);
    }

}
```

O passageiro, que neste momento já está no aeroporto, coloca-se na fila. As alterações relativas ao passageiro são realizadas no ficheiro *semSharedMemPassenger.c*.

Na função waitInQueue(), o número de passageiros na fila é incrementado, e o estado do passageiro passa a ser *IN_QUEUE*. O semáforo *passengersInQueue* anteriormente down pela hospedeira, passa a up, já o *passengersWaitInQueue* passa a down e o estado do passageiro passa a *IN_FLIGHT.*

```
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);
    }
    sh->fSt.nPassInQueue += 1;
    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);
    }

    if (semUp(semgid, sh->passengersInQueue) == -1)
    { // UP passenger in queue
```

```
        perror("error on the down operation for semaphore access (PG)");
        exit(EXIT_FAILURE);
    }

    if (semDown(semgid, sh->passengersWaitInQueue) == -1)
    { /* enter critical region */
        perror("error on the down 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);
    }

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

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

    sh->fSt.passengerChecked = passengerId;

    if (semUp(semgid, sh->idShown) == -1)
    { // UP id shown
        perror("error on the down operation for semaphore access");
        exit(EXIT_FAILURE);
    }
}
```

Neste momento, a hospedeira faz up do *passengersWaitInQueue* e começa a verificar a identidade dos passageiros (função *checkPassport()*), por isso o seu estado passa a *CHECK_PASSPORT e* o semáforo *idShown* passa a down. Aqui, o número de passageiros na fila decrementa, e consequentemente, o número de passageiros no voo incrementa, bem como o total de passageiros a bordo. No caso de faltar apenas um passageiro para o avião ter o número mínimo de passageiros e ainda um passageiro na fila; faltar apenas um passageiro para o avião estar cheio ou *n-1* passageiros já terem embarcado, o passageiro em causa é o último passageiro.

```
static bool checkPassport()
{
   bool last = false;

   /* insert your code here */
   if (semUp (semgid, sh->passengersWaitInQueue) == -1)  {
/* enter critical region */
       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);

   }

       sh->fSt.st.hostessStat = CHECK_PASSPORT;

   if( nPassengersInFlight() == MAXFC-1 || ( nPassengersInQueue() == 1 &&
nPassengersInFlight() >= MINFC-1 ) || sh->fSt.totalPassBoarded == N-1){
           last = true;
   }
   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);
   }


   if (semDown (semgid, sh->idShown) == -1)  {
/* Wait for Passengers to show id */
       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);

   }
```

```
        sh->fSt.nPassInQueue--;
        sh->fSt.nPassInFlight++;
        sh->fSt.totalPassBoarded++;
        savePassengerChecked(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;
}
```

O piloto aguarda que o embarque acabe e por isso o seu estado é atualizado para *WAITING_FOR_BOARDING*. Depois, dá down a *readyToFlight.*

```
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);
    }

    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);
    }

    if (semDown(semgid, sh->readyToFlight) == -1)
    {
        perror("error on the down operation for semaphore readyToFLight
(PT)");
        exit(EXIT_FAILURE);
    }
}
```

Com o avião pronto a descolar, a hospedeira dá up em *readyToFlight* depois de atualizar o seu estado para *READY_TO_FLIGHT* e de registar o número de passageiros que

embarcaram. Confirma também se os *n* passageiros estão a bordo e guarda o estado do voo como *finished*, caso se confirme.

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

   sh->fSt.st.hostessStat = READY_TO_FLIGHT;
   sh->fSt.nPassengersInFlight[sh->fSt.nFlight-1] = nPassengersInFlight();

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

   saveState(nFic, &sh->fSt);
   saveFlightDeparted(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);
   }


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

}
```

Os passageiros aguardam pela chegada ao destino, *passengersWaitInFlight* está down. De seguida, o seu estado muda para *AT_DESTINATION*, e à medida que desembarcam o número de passageiros no voo decrementa.

```
static void waitUntilDestination(unsigned int passengerId)
{

   if (semDown(semgid, sh->passengersWaitInFlight) == -1)
   { /* enter critical region */
       perror("error on the down 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);
    }

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

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

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

Enquanto deixa os passageiros no destino, o estado do piloto é *DROPING_PASSENGERS*. Depois o semáforo *passengersWaitInFlight* passa a up e o estado do voo *FlightArrived* é guardado. O semáforo planeEmpty passa a down, e se o número de passageiros no voo for zero, o passageiro dá up no planeEmpty.

```
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);
    }
    for(int i=0; i< sh->fSt.nPassengersInFlight[sh->fSt.nFlight-1];i++){
        if (semUp (semgid, sh->passengersWaitInFlight) == -1)  {
            perror ("error on the up operation for semaphore access (PT)");
            exit (EXIT_FAILURE);
        }
```

```
    }

    sh->fSt.st.pilotStat = DROPING_PASSENGERS;
    saveFlightArrived(nFic, &sh->fSt);
    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);
    }

    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);
    }

    if(sh->fSt.totalPassBoarded == N){
        sh->fSt.finished = 1;
    }

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

Testes

Teste Constantes Padrão

```
// FICHEIRO: probConst.h

/* Generic parameters */

#define N 21                          /** \brief number of passengers */

#define MINFC 5                       /** \brief min flight capacity */

#define MAXFC 10                      /** \brief max flight capacity */

#define MAXNF 10                      /** \brief max flight capacity */

#define MAXTRAVEL 30000.0             /** \brief max flight capacity */

#define MAXFLIGHT 2000.0              /** \brief max flight capacity */
```

```
// OUTPUT FICHEIRO: ./probSemSharedAirLift

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   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
  0  0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0
  0  0   1   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0
  0  0   1   1   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0
  0  0   1   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   0
  0  0   1   1   1   1   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   6   0   0
  0  0   1   1   1   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   7   0   0
  0  0   1   1   1   1   1   1   1   0   0   0   0   0   0   1   0   0   0   0   0   0   8   0   0
  0  0   1   1   1   1   1   1   1   0   0   0   0   0   1   1   0   0   0   0   0   0   9   0   0
  0  0   1   1   1   1   1   1   1   1   0   0   0   0   1   1   0   0   0   0   0   0  10   0   0
  0  0   1   1   1   1   1   1   1   1   0   0   0   0   1   1   0   0   0   0   0   1  11   0   0
  0  0   1   1   1   1   1   1   1   1   0   0   0   0   1   1   1   0   0   0   0   1  12   0   0
  0  0   1   1   1   1   1   1   1   1   0   0   1   0   1   1   1   0   0   0   0   1  13   0   0
  0  0   1   1   1   1   1   1   1   1   0   0   1   0   1   1   1   1   0   0   0   1  14   0   0
  0  0   1   1   1   1   1   1   1   1   1   0   1   0   1   1   1   1   0   0   0   1  15   0   0
  0  0   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   0   1  16   0   0
  0  0   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1  17   0   0
  0  0   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1  17   0   0
Flight 0 : 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   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   17   0   0
  1 0   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   17   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   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   17   0   0
  2 1   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   17   0   0
  2 2   1   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   17   0   0
  2 2   2   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   17   0   0
Flight 1 : Passenger 0 checked
  2 1   2   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   16   1   1
  2 2   2   1   1   1   1   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   16   1   1
  2 2   2   1   1   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   16   1   1
Flight 1 : Passenger 4 checked
  2 1   2   1   1   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   15   2   2
  2 2   2   1   1   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   15   2   2
  2 2   2   2   1   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   15   2   2
Flight 1 : Passenger 1 checked
  2 1   2   2   1   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   14   3   3
  2 2   2   2   1   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   14   3   3
  2 2   2   2   2   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   14   3   3
Flight 1 : Passenger 2 checked
  2 1   2   2   2   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   13   4   4
  2 2   2   2   2   1   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   13   4   4
  2 2   2   2   2   2   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   13   4   4
Flight 1 : Passenger 3 checked
  2 1   2   2   2   2   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   12   5   5
  2 2   2   2   2   2   2   1   1   1   1   0   1   1   1   1   1   1   0   0   1   0   1   12   5   5
  2 2   2   2   2   2   2   1   2   1   1   0   1   1   1   1   1   1   0   0   1   0   1   12   5   5
Flight 1 : Passenger 6 checked
  2 1   2   2   2   2   2   1   2   1   1   0   1   1   1   1   1   1   0   0   1   0   1   11   6   6
  2 1   2   2   2   2   2   2   2   1   1   0   1   1   1   1   1   1   0   0   1   0   1   11   6   6
  2 2   2   2   2   2   2   2   2   1   1   0   1   1   1   1   1   1   0   0   1   0   1   11   6   6
Flight 1 : Passenger 5 checked
  2 1   2   2   2   2   2   2   2   1   1   0   1   1   1   1   1   1   0   0   1   0   1   10   7   7
  2 2   2   2   2   2   2   2   2   1   1   0   1   1   1   1   1   1   0   0   1   0   1   10   7   7
  2 2   2   2   2   2   2   2   2   1   1   0   1   1   1   2   1   1   0   0   1   0   1   10   7   7
Flight 1 : Passenger 13 checked
  2 1   2   2   2   2   2   2   2   1   1   0   1   1   1   2   1   1   0   0   1   0   1    9   8   8
  2 2   2   2   2   2   2   2   2   1   1   0   1   1   1   2   1   1   0   0   1   0   1    9   8   8
  2 2   2   2   2   2   2   2   2   1   1   0   1   1   2   2   1   1   0   0   1   0   1    9   8   8
Flight 1 : Passenger 12 checked
  2 1   2   2   2   2   2   2   2   1   1   0   1   1   2   2   1   1   0   0   1   0   1    8   9   9
  2 2   2   2   2   2   2   2   2   1   1   0   1   1   2   2   1   1   0   0   1   0   1    8   9   9
  2 2   2   2   2   2   2   2   2   2   1   0   1   1   2   2   1   1   0   0   1   0   1    8   9   9
Flight 1 : Passenger 7 checked
```

```
 2 3  2 2 2 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7 10 10
Flight 1 : Departed with 10 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 2 2 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7 10 10
Flight 1 : Departed with 10 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 0  2 2 2 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7 10 10
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  2 2 2 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7 10 10
 4 0  3 2 2 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  9 10
 4 0  3 3 2 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  8 10
 4 0  3 3 3 2 2 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  7 10
 4 0  3 3 3 2 3 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  6 10
 4 0  3 3 3 3 3 2 2 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  5 10
 4 0  3 3 3 3 3 2 3 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  4 10
 4 0  3 3 3 3 3 3 3 2 1 0 1 1 2 2 1 1 0 0 1 0 1  7  3 10
 4 0  3 3 3 3 3 3 3 2 1 0 1 1 2 3 1 1 0 0 1 0 1  7  2 10
 4 0  3 3 3 3 3 3 3 2 1 0 1 1 3 3 1 1 0 0 1 0 1  7  1 10
 4 0  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 0 0 1 0 1  7  0 10
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  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 0 0 1 0 1  7  0 10
 1 0  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 0 0 1 0 1  7  0 10
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
 1 1  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 0 0 1 0 1  7  0 10
 1 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 0 0 1 0 1  7  0 10
 2 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 0 0 1 0 1  7  0 10
 2 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 1 0 1 0 1  8  0 10
 2 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 1 0 1 0 2  8  0 10
Flight 2 : Passenger 20 checked
 2 1  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 1 0 1 0 2  7  1 11
 2 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 1 1 1 0 1 0 2  7  1 11
 2 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 2 1 1 0 1 0 2  7  1 11
Flight 2 : Passenger 14 checked
 2 1  3 3 3 3 3 3 3 3 1 0 1 1 3 3 2 1 1 0 1 0 2  6  2 12
 2 2  3 3 3 3 3 3 3 3 1 0 1 1 3 3 2 1 1 0 1 0 2  6  2 12
 2 2  3 3 3 3 3 3 3 3 1 0 2 1 3 3 2 1 1 0 1 0 2  6  2 12
Flight 2 : Passenger 10 checked
 2 1  3 3 3 3 3 3 3 3 1 0 2 1 3 3 2 1 1 0 1 0 2  5  3 13
 2 2  3 3 3 3 3 3 3 3 1 0 2 1 3 3 2 1 1 0 1 0 2  5  3 13
 2 2  3 3 3 3 3 3 3 3 1 0 2 1 3 3 2 2 1 0 1 0 2  5  3 13
Flight 2 : Passenger 15 checked
 2 1  3 3 3 3 3 3 3 3 1 0 2 1 3 3 2 2 1 0 1 0 2  4  4 14
 2 2  3 3 3 3 3 3 3 3 1 0 2 1 3 3 2 2 1 0 1 0 2  4  4 14
```

```
 2 2  3 3 3 3 3 3 3 3 2 0 2 1 3 3 2 2 1 0 1 0 2  4 4 14
Flight 2 : Passenger 8 checked
 2 1  3 3 3 3 3 3 3 3 2 0 2 1 3 3 2 2 1 0 1 0 2  3 5 15
 2 2  3 3 3 3 3 3 3 3 2 0 2 1 3 3 2 2 1 0 1 0 2  3 5 15
 2 2  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 1 0 1 0 2  3 5 15
Flight 2 : Passenger 11 checked
 2 1  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 1 0 1 0 2  2 6 16
 2 2  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 1 0 1 0 2  2 6 16
 2 2  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 1 0 2 0 2  2 6 16
Flight 2 : Passenger 18 checked
 2 1  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 1 0 2 0 2  1 7 17
 2 2  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 1 0 2 0 2  1 7 17
 2 2  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 2 0 2 0 2  1 7 17
Flight 2 : Passenger 16 checked
 2 3  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 2 0 2 0 2  0 8 18
Flight 2 : Departed with 8 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 3 3 3 3 3 2 0 2 2 3 3 2 2 2 0 2 0 2  0 8 18
Flight 2 : Departed with 8 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 0  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 2 0 2 0 2  0 8 18
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  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 2 0 2 0 2  0 8 18
 4 0  3 3 3 3 3 3 3 3 2 0 2 2 3 3 2 2 2 0 2 0 3  0 7 18
 4 0  3 3 3 3 3 3 3 3 2 0 2 2 3 3 3 2 2 0 2 0 3  0 6 18
 4 0  3 3 3 3 3 3 3 3 2 0 3 2 3 3 3 2 2 0 2 0 3  0 5 18
 4 0  3 3 3 3 3 3 3 3 2 0 3 2 3 3 3 3 2 0 2 0 3  0 4 18
 4 0  3 3 3 3 3 3 3 3 3 0 3 2 3 3 3 3 2 0 2 0 3  0 3 18
 4 0  3 3 3 3 3 3 3 3 3 0 3 3 3 3 3 3 2 0 2 0 3  0 2 18
 4 0  3 3 3 3 3 3 3 3 3 0 3 3 3 3 3 3 2 0 3 0 3  0 1 18
 4 0  3 3 3 3 3 3 3 3 3 0 3 3 3 3 3 3 3 0 3 0 3  0 0 18
 4 0  3 3 3 3 3 3 3 3 3 0 3 3 3 3 3 3 3 0 3 1 3  1 0 18
 4 0  3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 0 3 1 3  2 0 18
 4 0  3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3  3 0 18
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 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3  3 0 18
 1 0  3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3  3 0 18
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 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3  3 0 18
 2 1  3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3  3 0 18
 2 2  3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3  3 0 18
 2 2  3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 2 3  3 0 18
Flight 3 : Passenger 19 checked
```

```
  2  1   3  3  3  3  3  3  3  3  3  1  3  3  3  3  3  3  3  1  3  2  3   2  1  19
  2  2   3  3  3  3  3  3  3  3  3  1  3  3  3  3  3  3  3  1  3  2  3   2  1  19
  2  2   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  1  3  2  3   2  1  19
Flight 3 : Passenger 9 checked
  2  1   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  1  3  2  3   1  2  20
  2  2   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  1  3  2  3   1  2  20
  2  2   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  2  3  2  3   1  2  20
Flight 3 : Passenger 17 checked
  2  3   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  2  3  2  3   0  3  21
Flight 3 : Departed with 3 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
Flight 3 : Departed with 3 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  2  3  3  3  3  3  3  3  2  3  2  3   0  3  21
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  3   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  2  3  2  3   0  3  21
  4  3   3  3  3  3  3  3  3  3  3  2  3  3  3  3  3  3  3  3  3  2  3   0  2  21
  4  3   3  3  3  3  3  3  3  3  3  2  3  3  3  3  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
AirLift result
AirLift used 3 Flights
Flight 1 took 10 passengers
Flight 2 took  8 passengers
Flight 3 took  3 passengers
teste 1
```

Testes com Diversas Constantes

```
#define N 50                          /** \brief number of passengers */

#define MINFC 20                       /** \brief min flight capacity */

#define MAXFC 30                      /** \brief max flight capacity */

#define MAXNF 20                      /** \brief max flight capacity */

#define MAXTRAVEL 20000.0             /** \brief max flight capacity */

#define MAXFLIGHT 3000.0              /** \brief max flight capacity */
```

```
AirLift used 2 Flights
Flight 1 took 30 passengers
Flight 2 took 20 passengers
teste 2
```

```
#define N 10                          /** \brief number of passengers */

#define MINFC 2                       /** \brief min flight capacity */

#define MAXFC 8                      /** \brief max flight capacity */

#define MAXNF 20                      /** \brief max flight capacity */

#define MAXTRAVEL 70000.0             /** \brief max flight capacity */

#define MAXFLIGHT 5000.0              /** \brief max flight capacity */
```

```
AirLift used 5 Flights
Flight 1 took  2 passengers
Flight 2 took  2 passengers
Flight 3 took  2 passengers
Flight 4 took  2 passengers
Flight 5 took  2 passengers
teste 3
```

```
#define N 60                            /** \brief number of passengers */

#define MINFC 20                        /** \brief min flight capacity */

#define MAXFC 30                        /** \brief max flight capacity */

#define MAXNF 30                        /** \brief max flight capacity */

#define MAXTRAVEL 230000.0              /** \brief max flight capacity */

#define MAXFLIGHT 501200.0              /** \brief max flight capacity */
```

```
AirLift result
AirLift used 3 Flights
Flight 1 took 29 passengers
Flight 2 took 30 passengers
Flight 3 took  1 passengers
```

teste 4

```
#define N 18                           /** \brief number of passengers */

#define MINFC 5                        /** \brief min flight capacity */

#define MAXFC 10                       /** \brief max flight capacity */

#define MAXNF 20                       /** \brief max flight capacity */

#define MAXTRAVEL 45000.0              /** \brief max flight capacity */

#define MAXFLIGHT 12983.0               /** \brief max flight capacity */
```

```
AirLift result
AirLift used 3 Flights
Flight 1 took  5 passengers
Flight 2 took 10 passengers
Flight 3 took  3 passengers
```

teste 5

Conclusão

O desenvolvimento da aplicação pressupôs uma boa compreensão do problema proposto, sendo que fazer um diagrama do fluxo de execução e, posteriormente, uma tabela com os semáforos utilizados em cada função, tenha sido um dos facilitadores do trabalho.
Os testes executados permitiram confirmar o bom funcionamento da aplicação, uma vez que ao simular com diferentes valores para o número de passageiros, números mínimos e máximos de passageiros por voo, foram obtidos resultados que cumpriam sempre esses requisitos.