Sistemas Operativos

Airlift Simulation

Licenciatura em Engenharia Informática

Leonardo dos Santos Flórido - 103360 - 50%
Gabriel Hall Abreu - 102851 - 50%

Índice

Introdução

O segundo trabalho prático de Sistemas Operativos consiste numa aplicação em C que, através da utilização de semáforos, simula o processo necessário de uma Ponte Aérea.

Este problema tem como objetivos compreender a utilização de semáforos, assim como os mecanismos associados à execução e sincronização de processos e threads.

Será necessário alterar o código base fornecido pelo o docente em 3 ficheiros correspondentes às entidades: hostess, passenger e pilot.

Estruturas de dados

Neste problema existem três estruturas de dados bastante importantes: FULL_STAT, STAT e SHARED_DATA. A estrutura STAT está incorporada na estrutura FULL_STAT e contém as informações dos estados de cada entidade do problema. A estrutura FULL_STAT além da estrutura STAT tem também as variáveis partilhadas usadas neste problema. Por fim, a estrutura SHARED_DATA contém a estrutura FULL_STAT e também os semáforos usados na resolução deste problema.

```
24  typedef struct
25  { /** \brief pilot state */
26      unsigned int pilotStat;
27      /** \brief hostess state */
28      unsigned int hostessStat;
29      /** \brief passengers state array */
30      unsigned int passengerStat[N];
31
32  } STAT;
33
34
35  /**
36   *  \brief Definition of <em>full state of the problem</em> data type.
37   */
38  typedef struct
39  { /** \brief state of all intervening entities */
40      STAT st;
41      /** \brief number of passengers at each flight */
42      unsigned int nPassengersInFlight[MAXNF];
43      /** \brief flight number */
44      unsigned int nFlight;
45
46      /** \brief number of passengers waiting */
47      unsigned int nPassInQueue;
48      /** \brief number of passengers flying */
49      unsigned int nPassInFlight;
50      /** \brief total number of passengers already boarded in every flight */
51      unsigned int totalPassBoarded;
52      /** \brief air lift finished */
53      bool finished;
54      /** \brief passenger id of last passenger to check passport */
55      int passengerChecked;
56
57  } FULL_STAT;
```

```
26  typedef struct
27          { /** \brief full state of the problem */
28            FULL_STAT fSt;
29
30            /* semaphores ids */
31            /** \brief identification of critical region protection semaphore – val = 1 */
32            unsigned int mutex;
33            /** \brief identification of semaphore used by hostess to wait for passengers - val = 0 */
34            unsigned int passengersInQueue;
35            /** \brief identification of semaphore used by passengers to wait for hostess – val = 0 */
36            unsigned int passengersWaitInQueue;
37            /** \brief identification of semaphore used by passengers to wait for flight to end – val = 0 */
38            unsigned int passengersWaitInFlight;
39            /** \brief identification of semaphore used by hostess to wait for starting boarding – val = 0  */
40            unsigned int readyForBoarding;
41            /** \brief identification of semaphore used by pilot to wait for boarding to complete - val = 0 */
42            unsigned int readyToFlight;
43            /** \brief identification of semaphore used by hostess to wait for passenger identification - val = 0 */
44            unsigned int idShown;
45            /** \brief identification of semaphore used by pilot to wait for last passenger to leave plane - val = 0 */
46            unsigned int planeEmpty;
47
48          } SHARED_DATA;
```

Foi feita uma tabela com informação sobre os semáforos, em que entidades fazem Up e Down e em que funções acontecem.

	Entidades		Funções	
Semáforo	Up	Down	Up	Down
mutex	Todas	Todas	Todas	Todas
passengersInQueue	Passengers	Hostess	waitInQueue	waitForPassenger
passengersWaitInQueue	Hostess	Passengers	checkPassport	waitInQueue
passengersWaitInFlight	Pilot	Passengers	dropPassengersAtTarget	waitUntilDestination
readyForBoarding	Pilot	Hostess	signalReadyForBoarding	waitForNextFlight
readyToFlight	Hostess	Pilot	signalReadyToFlight	waitUntilReadyToFlight
idShown	Passengers	Hostess	waitInQueue	checkPassport
planeEmpty	Passengers	Pilot	waitUntilDestination	dropPassengersAtTarget

Entidade Hostess

A hostess começa inicialmente à espera que o pilot assinale quando os passengers podem embarcar. Quando o pilot lhe dá sinal de confirmação, fica à espera dos passengers e quando começam a chegar, começa a verificar o passport e o id de cada passenger deixando-os entrar no avião. Quando o avião já estiver apto para se deslocar a hostess avisa o pilot.
O ciclo de vida da hostess consiste em esperar pela confirmação do pilot para iniciar o boarding, esperar pelos passengers, verificar o passport e o id de cada passenger e dizer ao pilot que o avião está pronto para partir.

```
114     /* simulation of the life cycle of the hostess */
115
116     int nPassengers = 0;
117     bool lastPassengerInFlight;
118
119     while (nPassengers < N)
120     {
121         waitForNextFlight();
122         do
123         {
124             waitForPassenger();
125             lastPassengerInFlight = checkPassport();
126             nPassengers++;
127         } while (!lastPassengerInFlight);
128         signalReadyToFlight();
129     }
```

Esta entidade tem 4 funções: waitForNextFlight, waitForPassenger, checkPassport e signalReadyToFlight.

waitForNextFlight()

Nesta função a hostess entra na zona crítica (Down no mutex) para alterar o seu estado para WAIT_FOR_FLIGHT (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex) e da semDown no semáforo READYFORBOARDING.

```
143  /**
144   *  \brief wait for Next Flight.
145   *
146   *  Hostess updates its state and waits for plane to be ready for boarding
147   *  The internal state should be saved.
148   *
149   */
150
151  static void waitForNextFlight()
152  {
153      if (semDown(semgid, sh->mutex) == -1)
154      { /* enter critical region */
155          perror("error on the down operation for semaphore access (HT)");
156          exit(EXIT_FAILURE);
157      }
158
159      /* insert your code here */
160      sh->fSt.st.hostessStat = WAIT_FOR_FLIGHT;
161      saveState(nFic, &sh->fSt);
162
163      if (semUp(semgid, sh->mutex) == -1)
164      { /* exit critical region */
165          perror("error on the up operation for semaphore access (HT)");
166          exit(EXIT_FAILURE);
167      }
168
169      /* insert your code here */
170      semDown(semgid, READYFORBOARDING);
171  }
```

waitForPassenger()

Nesta função a hostess entra na zona crítica (Down no mutex) para alterar o seu estado para WAIT_FOR_PASSENGER (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex) e dá semDown no semáforo PASSENGERSINQUEUE.

```
173  /**
174   *  \brief hostess waits for passenger
175   *
176   *  hostess waits for passengers to arrive at airport.
177   *  The internal state should be saved.
178   */
179
180  static void waitForPassenger()
181  {
182      if (semDown(semgid, sh->mutex) == -1)
183      { /* enter critical region */
184          perror("error on the down operation for semaphore access (HT)");
185          exit(EXIT_FAILURE);
186      }
187
188      /* insert your code here */
189      sh->fSt.st.hostessStat = WAIT_FOR_PASSENGER;
190      saveState(nFic, &sh->fSt);
191
192      if (semUp(semgid, sh->mutex) == -1)
193      { /* exit critical region */
194          perror("error on the up operation for semaphore access (HT)");
195          exit(EXIT_FAILURE);
196      }
197
198      /* insert your code here */
199      semDown(semgid, PASSENGERSINQUEUE);
200  }
```

checkPassport()

Nesta função a hostess dá semUp no semáforo PASSENGERSWAITINQUEUE para avisar os passengers que vai começar a verificar os passports e entra na zona crítica (Down no mutex) muda também o seu estado para CHECK_PASSPORT (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex). A hostess verifica o passport do passenger e fica à espera do ID. Logo dá semDown no semáforo IDSHOWN. De seguida, volta a entrar na zona crítica (Down no mutex) para atualizar as variáveis partilhadas. Como um dos passenger que estava na fila entra no avião, decrementa a variável nPassInQueue (Número de passengers na fila), incrementa a variável nPassInFlight (Número de passengers no avião) e incrementa a variável totalPassBoarded (Número de passengers que já foram embarcados). O passenger embarcado é guardado através da função savePassengerChecked(), voltamos a guardar o estado através da função saveState(). Ainda dentro da zona crítica, queremos saber se o passenger embarcado é o último, last = true se for o último ou last = false se não o for. Se o número de passengers embarcados naquele voo for igual à capacidade máxima ou se o avião já tiver a quantidade mínima de tripulantes e já não existirem passengers na fila ou se já foram todos os passenger embarcados, last = true, caso contrário last = false. Depois sai da zona crítica (Down no mutex), no final a função retorna a variável last.

```
202  /**
203   *  \brief passport check
204   *
205   *  The hostess checks passenger passport and waits for passenger to show id
206   *  The internal state should be saved twice.
207   *
208   *  \return should be true if this is the last passenger for this flight
209   *    that is:
210   *      - flight is at its maximum capacity
211   *      - flight is at or higher than minimum capacity and no passenger waiting
212   *      - no more passengers
213   */
214
215  static bool checkPassport()
216  {
217      bool last;
218
219      /* insert your code here */
220      semUp(semgid, PASSENGERSWAITINQUEUE);
221
222      if (semDown(semgid, sh->mutex) == -1)
223      { /* enter critical region */
224          perror("error on the down operation for semaphore access (HT)");
225          exit(EXIT_FAILURE);
226      }
227
228      /* insert your code here */
229      sh->fSt.st.hostessStat = CHECK_PASSPORT;
230      saveState(nFic, &sh->fSt);
231
232      if (semUp(semgid, sh->mutex) == -1)
233      { /* exit critical region */
234          perror("error on the up operation for semaphore access (HT)");
235          exit(EXIT_FAILURE);
236      }
237
238      /* insert your code here */
239      semDown(semgid, IDSHOWN);
240
241      if (semDown(semgid, sh->mutex) == -1)
242      { /* enter critical region */
243          perror("error on the up operation for semaphore access (HT)");
244          exit(EXIT_FAILURE);
245  }
246
247  /* insert your code here */
248  sh->fSt.nPassInQueue--;
249  sh->fSt.nPassInFlight++;
250  sh->fSt.totalPassBoarded++;
251  savePassengerChecked(nFic, &sh->fSt);
252  saveState(nFic, &sh->fSt);
253  if ((nPassengersInFlight() == MAXFC) || ((MINFC <= nPassengersInFlight()) && (nPassengersInQueue() == 0)) || (sh->fSt.totalPassBoarded == N))
254      last = true;
255  else
256      last = false;
257
258  if (semUp(semgid, sh->mutex) == -1)
259  { /* exit critical region */
260      perror("error on the up operation for semaphore access (HT)");
261      exit(EXIT_FAILURE);
262  }
263
264  /* insert your code here */
265  last = last & true;
266
267  return last;
268  }
```

signalReadyToFlight()

Nesta função a hostess entra na zona crítica (Down no mutex) para alterar o seu estado para READY_TO_FLIGHT (o estado é guardado através da função saveState()), além disso regista o número de passengers que estão dentro do avião daquele voo no array nPassengersInFlight (Cada elemento do array corresponde ao número de passengers que foram embarcados no voo correspondente ao seu (index + 1)), o voo é salvo através da função saveFlightDeparted(). A hostess verifica se todos os passengers com destino a *Target* já foram embarcados, se for true então altera a variável partilhada finished para true, para o pilot saber que este é o seu último voo com passengers. Depois sai da zona crítica (Up no mutex) e dá semUp no semáforo READYTOFLIGHT para avisar o pilot que pode começar o voo.

```
280  /**
281   *  \brief signal ready to flight
282   *
283   *  The flight is ready to go.
284   *  The hostess updates her state, registers the number of passengers in this flight
285   *  and checks if the airlift is finished (all passengers have boarded).
286   *  Hostess informs pilot that plane is ready to flight.
287   *  The internal state should be saved.
288   */
289  void signalReadyToFlight()
290  {
291      if (semDown(semgid, sh->mutex) == -1)
292      { /* enter critical region */
293          perror("error on the down operation for semaphore access (HT)");
294          exit(EXIT_FAILURE);
295      }
296
297      /* insert your code here */
298      sh->fSt.st.hostessStat = READY_TO_FLIGHT;
299      saveState(nFic, &sh->fSt);
300      // registers the number of passengers in this flight
301      sh->fSt.nPassengersInFlight[sh->fSt.nFlight - 1] = nPassengersInFlight();
302      saveFlightDeparted(nFic, &sh->fSt);
303      // check if the airlift is finished (all passengers have boarded)
304      if (sh->fSt.totalPassBoarded == N)
305          sh->fSt.finished = true;
306
307      if (semUp(semgid, sh->mutex) == -1)
308      { /* exit critical region */
309          perror("error on the up operation for semaphore access (HT)");
310          exit(EXIT_FAILURE);
311      }
312
313      /* insert your code here */
314      semUp(semgid, READYTOFLIGHT);
315  }
```

Entidade Passenger

Os passengers chegam ao aeroporto num tempo random, quando chegam ao aeroporto ficam à espera na fila para entrar no avião, são atendidos pela hostess para entregarem o passport e o id, de seguida a hostess deixa-os entrar no avião, quando estiverem dentro avião e já estiverem a voar são avisados pelo o pilot que podem começar a sair do avião. O último passenger diz ao pilot que o avião já não tem mais ninguém. O ciclo de vida do passenger consiste em ir para o aeroporto, esperar na fila, entregar o passport e o id, esperar que o voo acabe e sair do avião.

```
107     /* simulation of the life cycle of the passenger */
108
109     travelToAirport();
110     waitInQueue(n);
111     waitUntilDestination(n);
```

Esta entidade tem 4 funções: travelToAirport, leavePlane, waitInQueue e waitUntilDestination.

travelToAirport()

Os passengers chegam ao aeroporto num tempo random.

```
125 /**
126  *  \brief passenger goes to airport
127  *
128  *  The passenger takes a random time to reach the airport
129  */
130
131 static bool travelToAirport()
132 {
133     usleep((unsigned int)floor((MAXTRAVEL * random()) / RAND_MAX + 1000));
134
135     return true;
136 }
```

leavePlane(unsigned int passengerId)

O propósito desta função é decrementar a variável partilhada nPassInFlight e mudar o estado do passenger para AT_DESTINATION. Esta função será utilizada na zona crítica da função waitUntilDestination().

```
138  /**
139   *  \brief passenger leaves the plane.
140   *
141   *  Update the number of passengers in flight and arrive at destination.
142   *
143   *  \param passengerId passenger id
144   */
145
146  static void leavePlane(unsigned int passengerId)
147  {
148      sh->fSt.nPassInFlight--;
149      sh->fSt.st.passengerStat[passengerId] = AT_DESTINATION;
150  }
```

waitInQueue(unsigned int passengerId)

Nesta função o passenger entra na zona crítica (Down no mutex) para atualizar a variável partilhada e mudar o seu estado. Como o passenger está na fila incrementamos a variável nPassInQueue e alteramos o seu estado para IN_QUEUE (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex). O passenger dá semUp ao semáforo PASSENGERSINQUEUE visto encontrar-se já na fila. De seguida dá semDown ao semáforo PASSENGERSWAITINQUEUE porque está à espera que a hostess verifique o seu passport. O passenger volta a entrar na zona crítica (Down no mutex) para alterar a variável partilhada passengerChecked, a variável passengerChecked vai ser igual ao id do último passenger que a hostess verificou o passport, o passenger vai alterar também o seu estado para IN_FLIGHT (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex). A hostess já verificou o passport do passenger e pediu-lhe o id, o passenger dá semUp no semáforo IDSHOWN para dar o id e permissão para ler.

```
/**
 *  \brief wait for its turn to be checked by hostess
 *
 *  Passenger should update number of passenger in queue, and inform hostess that he is ready for boarding
 *  after being acknowledged by hostess passenger should provide its id to hostess and giver her permission to read the id
 *  The internal state should be saved twice.
 *
 *  \param passengerId passenger id
 */

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 */
    semUp(semgid, PASSENGERSINQUEUE);
    semDown(semgid, PASSENGERSWAITINQUEUE);

    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);

    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 */
    semUp(semgid, IDSHOWN);
}
```

waitUntilDestination(unsigned int passengerId)

Nesta função o passenger dá semDown ao semáforo PASSENGERWAITINFLIGHT, pois encontra-se dentro do avião a espera que o voo acabe. O passenger entra na zona crítica (Down no mutex) e recorrendo a função leavePlane() o passenger vai sair do avião decrementado a variável partilhada nPassInFlight e atualizando o seu estado para AT_DESTINATION (o estado é guardado através da função saveState()). O último passenger a sair do avião informa o pilot que o avião

está vazio, esta ação ocorre quando o nPassInFlight for igual a zero, quando isso acontece o último passenger da semUp no semáforo PLANEEMPTY, informando assim ao pilot que o avião está vazio. Depois sai da zona crítica (Up no mutex).

```
205  /**
206   *  \brief passenger waits for flight to terminate and arrives at destination.
207   *
208   *  passenger should wait for flight end, update the number of passengers in flight and
209   *  arrive at destination.
210   *  last passenger must inform pilot that plane is empty.
211   *  The internal state should be saved.
212   *
213   *  \param passengerId passenger id
214   */
215
216  static void waitUntilDestination(unsigned int passengerId)
217  {
218
219      /* insert your code here */
220      semDown(semgid, PASSENGERSWAITINFLIGHT);
221
222      if (semDown(semgid, sh->mutex) == -1)
223      { /* enter critical region */
224          perror("error on the down operation for semaphore access (PG)");
225          exit(EXIT_FAILURE);
226      }
227
228      /* insert your code here */
229      leavePlane(passengerId);
230      saveState(nFic, &sh->fSt);
231      // last passenger inform pilot that plane is empty
232      if (sh->fSt.nPassInFlight == 0)
233          semUp(semgid, PLANEEMPTY);
234
235      if (semUp(semgid, sh->mutex) == -1)
236      { /* exit critical region */
237          perror("error on the up operation for semaphore access (PG)");
238          exit(EXIT_FAILURE);
239      }
240  }
241
```

Entidade Pilot

O pilot inicialmente está a voar de *Target* para *Origin*, quando chega à *Origin* informa a hostess que o avião está pronto para ser embarcado. O pilot fica à espera que a hostess lhe diga quando o avião está pronto. Depois de receber a informação que o avião está pronto, o pilot começa o voo de *Origin* para *Target*. Quando chega a *Target* deixa os passengers no aeroporto e volta para *Origin*. Quando não houver mais passengers com destino a *Target* o pilot acaba o seu ciclo de vida. O ciclo de vida do pilot consiste em voar de *Target* para *Origin* e de *Origin* para *Target*, informar a hostess que o avião está pronto para ser embarcado, esperar pelo o sinal da hostess que o avião está pronto para voar e deixar os passengers em Target.

```
104     /* simulation of the life cycle of the pilot */
105
106     while (!isFinished())
107     {
108         flight(false); // from target to origin
109         signalReadyForBoarding();
110         waitUntilReadyToFlight();
111         flight(true); // from origin to target
112         dropPassengersAtTarget();
113     }
```

Esta entidade tem 4 funções: flight, signalReadyForBoarding, waitUntilReadyToFlight e dropPassengersAtTarget.

flight(bool go)

O pilot entra na zona crítica (Down no mutex) se a variável go for true vai alterar o seu estado para FLYING, se for false vai alterar o seu estado para FLYING_BACK (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex).

```
135  /**
136   *  \brief flight.
137   *
138   *  The pilot takes passenger to destination (go) or
139   *  plane back to starting airport (return)
140   *  state should be saved.
141   *
142   *  \param go true if going to destination
143   */
144
145  static void flight(bool go)
146  {
147      if (semDown(semgid, sh->mutex) == -1)
148      { /* enter critical region */
149          perror("error on the up operation for semaphore access (PT)");
150          exit(EXIT_FAILURE);
151      }
152
153      /* insert your code here */
154      if (go == true)
155      {
156          sh->fSt.st.pilotStat = FLYING;
157      }
158      else
159      {
160          sh->fSt.st.pilotStat = FLYING_BACK;
161      }
162      saveState(nFic, &sh->fSt);
163
164      if (semUp(semgid, sh->mutex) == -1)
165      { /* exit critical region */
166          perror("error on the up operation for semaphore access (PT)");
167          exit(EXIT_FAILURE);
168      }
169
170      usleep((unsigned int)floor((MAXFLIGHT * random()) / RAND_MAX + 100.0));
171  }
```

signalReadyForBoarding()

O pilot entra na zona crítica (Down no mutex) altera o seu estado para READY_FOR_BOARDING e incrementa a variável partilhada nFlight (Número de voo que vai ocorrer), o estado é guardado através da função saveState(). Depois sai da zona crítica (Up no mutex). O pilot avisa a hostess que o avião está pronto para ser embarcado e por isso dá semUp no semáforo READYFORBOARDING.

```
173  /**
174   *  \brief pilot informs hostess that plane is ready for boarding
175   *
176   *  The pilot updates its state and signals the hostess that boarding may start
177   *  The flight number should be updated.
178   *  The internal state should be saved.
179   */
180
181  static void signalReadyForBoarding()
182  {
183      if (semDown(semgid, sh->mutex) == -1)
184      { /* enter critical region */
185          perror("error on the down operation for semaphore access (PT)");
186          exit(EXIT_FAILURE);
187      }
188
189      /* insert your code here */
190      sh->fSt.st.pilotStat = READY_FOR_BOARDING;
191      sh->fSt.nFlight++;
192      saveState(nFic, &sh->fSt);
193      saveStartBoarding(nFic, &sh->fSt);
194
195      if (semUp(semgid, sh->mutex) == -1)
196      { /* exit critical region */
197          perror("error on the up operation for semaphore access (PT)");
198          exit(EXIT_FAILURE);
199      }
200
201      /* insert your code here */
202      semUp(semgid, READYFORBOARDING);
203  }
```

waitUntilReadyToFlight()

O pilot entra na zona crítica (Down no mutex) para alterar o seu estado para WAITING_FOR_BOARDING (o estado é guardado através da função saveState()). Depois sai da zona crítica (Up no mutex) e dá semDown no semáforo READYTOFLIGHT.

```
205  /**
206   *  \brief pilot waits for plane to get filled with passengers.
207   *
208   *  The pilot updates its state and wait for Boarding to finish
209   *  The internal state should be saved.
210   */
211
212  static void waitUntilReadyToFlight()
213  {
214      if (semDown(semgid, sh->mutex) == -1)
215      { /* enter critical region */
216          perror("error on the up operation for semaphore access (PT)");
217          exit(EXIT_FAILURE);
218      }
219
220      /* insert your code here */
221      sh->fSt.st.pilotStat = WAITING_FOR_BOARDING;
222      saveState(nFic, &sh->fSt);
223
224      if (semUp(semgid, sh->mutex) == -1)
225      { /* exit critical region */
226          perror("error on the up operation for semaphore access (PT)");
227          exit(EXIT_FAILURE);
228      }
229
230      /* insert your code here */
231      semDown(semgid, READYTOFLIGHT);
232  }
```

dropPassengersAtTarget()

O pilot entra na zona crítica (Down no mutex), recorrendo à função saveFlightArrived() salva o estado do voo, de seguida o pilot altera o seu estado para DROPING_PASSENGERS (o estado é guardado através da função saveState()). Por cada passenger que se encontra no avião o pilot dá semUp no semáforo PASSENGERSWAITINFLIGHT para indicar a cada passenger que já pode sair do avião. Depois sai da zona crítica (Up no mutex). O pilot dá semDown no semáforo PLANEEMPTY porque está à espera que todos passengers saiam do avião. O pilot volta a entrar na zona crítica (Down no mutex) para salvar o estado do voo recorrendo à função saveFlightReturning visto que o pilot, quando o avião estiver vazio, retorna para *Origin*. Depois sai da zona crítica (Up no mutex).

```
234 /**
235  *  \brief pilot drops passengers at destination.
236  *
237  *  Pilot update its state and allows passengers to leave plane
238  *  Pilot must wait for all passengers to leave plane before starting to return.
239  *  The internal state should not be saved twice (after allowing passengeres to leave and after the plane is empty).
240  */
241
242 static void dropPassengersAtTarget()
243 {
244     if (semDown(semgid, sh->mutex) == -1)
245     { /* enter critical region */
246         perror("error on the down operation for semaphore access (PT)");
247         exit(EXIT_FAILURE);
248     }
249
250     /* insert your code here */
251     saveFlightArrived(nFic, &sh->fSt);
252     sh->fSt.st.pilotStat = DROPING_PASSENGERS;
253     saveState(nFic, &sh->fSt);
254     for (int i = sh->fSt.nPassInFlight; i > 0; i--)
255     {
256         semUp(semgid, PASSENGERSWAITINFLIGHT);
257     }
258
259     if (semUp(semgid, sh->mutex) == -1)
260     { /* exit critical region */
261         perror("error on the up operation for semaphore access (PT)");
262         exit(EXIT_FAILURE);
263     }
264
265     /* insert your code here */
266     semDown(semgid, PLANEEMPTY);
267
268     if (semDown(semgid, sh->mutex) == -1)
269     { /* enter critical region */
270         perror("error on the down operation for semaphore access (PT)");
271         exit(EXIT_FAILURE);
272     }
273
274     /* insert your code here */
275     saveFlightReturning(nFic, &sh->fSt);
276
277     if (semUp(semgid, sh->mutex) == -1)
278     { /* exit critical region */
279         perror("error on the up operation for semaphore access (PT)");
280         exit(EXIT_FAILURE);
281     }
282 }
283
```

Resultados obtidos

```
                                  _______
                        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   0   0   0   0   0   0   0   0   0   0   0   0    0   0   0
  0  0    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0   0   0
  1  0    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0   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   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0   0   0
  2  1    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0   0   0
  2  1    0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0    1   0   0
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0    1   0   0
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0    1   0   0
Flight 1 : Passenger 10 checked
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0    0   1   1
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0    0   1   1
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   1   0   0   0    1   1   1
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   1   0   0   0    1   1   1
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   2   0   0   0    1   1   1
Flight 1 : Passenger 17 checked
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   2   0   0   0    0   2   2
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   2   0   0   0    0   2   2
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   1   0   0   0   2   0   0   0    1   2   2
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   1   0   0   0   2   0   0   0    1   2   2
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   0   0    1   2   2
Flight 1 : Passenger 13 checked
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   0   0    0   3   3
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   0   0    0   3   3
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   1   0    1   3   3
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   1   0    1   3   3
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   2   0    1   3   3
Flight 1 : Passenger 19 checked
  2  2    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   2   0    0   4   4
  2  1    0   0   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   2   0   2   0    0   4   4
  2  1    0   0   0   0   0   0   0   0   1   0   2   0   0   2   0   0   0   2   0   2   0    1   4   4
  2  2    0   0   0   0   0   0   0   0   1   0   2   0   0   2   0   0   0   2   0   2   0    1   4   4
  2  2    0   0   0   0   0   0   0   0   2   0   2   0   0   2   0   0   0   2   0   2   0    1   4   4
Flight 1 : Passenger 8 checked
  2  2    0   0   0   0   0   0   0   0   2   0   2   0   0   2   0   0   0   2   0   2   0    0   5   5
  2  3    0   0   0   0   0   0   0   0   2   0   2   0   0   2   0   0   0   2   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   0   0   0   2   0   2   0   0   2   0   0   0   2   0   2   0    0   5   5
  3  0    0   0   0   0   0   0   0   0   2   0   2   0   0   2   0   0   0   2   0   2   0    0   5   5
  3  0    0   0   0   0   0   0   0   0   2   0   2   0   0   2   0   1   0   2   0   2   0    1   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   0   0   0   2   0   2   0   0   2   0   1   0   2   0   2   0    1   5   5
  4  0    0   0   0   0   0   0   0   0   2   0   3   0   0   2   0   1   0   2   0   2   0    1   4   5
  4  0    0   0   0   0   0   0   0   0   2   0   3   0   0   2   0   1   0   2   0   3   0    1   3   5
  4  0    0   0   0   0   0   0   0   0   3   0   3   0   0   2   0   1   0   2   0   3   0    1   2   5
  4  0    0   0   0   0   0   0   0   0   3   0   3   0   0   3   0   1   0   2   0   3   0    1   1   5
  4  0    0   0   0   0   0   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    1   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   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    1   0   5
  0  0    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    2   0   5
  1  0    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    2   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   1   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    2   0   5
  2  1    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    2   0   5
  2  2    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   1   0   3   0   3   0    2   0   5
  2  2    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    2   0   5
Flight 2 : Passenger 15 checked
  2  2    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    1   1   6
  2  1    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    1   1   6
  2  2    0   0   0   0   1   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    1   1   6
  2  2    0   0   0   0   2   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    1   1   6
Flight 2 : Passenger 4 checked
  2  2    0   0   0   0   2   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    0   2   7
  2  1    0   0   0   0   2   0   0   0   3   0   3   0   0   3   0   2   0   3   0   3   0    0   2   7
  2  1    0   0   0   0   2   0   0   1   3   0   3   0   0   3   0   2   0   3   0   3   0    1   2   7
  2  2    0   0   0   0   2   0   0   1   3   0   3   0   0   3   0   2   0   3   0   3   0    1   2   7
  2  2    0   0   0   0   2   0   0   2   3   0   3   0   0   3   0   2   0   3   0   3   0    1   2   7
Flight 2 : Passenger 7 checked
  2  2    0   0   0   0   2   0   0   2   3   0   3   0   0   3   0   2   0   3   0   3   0    0   3   8
  2  1    0   0   0   0   2   0   0   2   3   0   3   0   0   3   0   2   0   3   0   3   0    0   3   8
  2  1    1   0   0   0   2   0   0   2   3   0   3   0   0   3   0   2   0   3   0   3   0    1   3   8
  2  2    1   0   0   0   2   0   0   2   3   0   3   0   0   3   0   2   0   3   0   3   0    1   3   8
  2  2    2   0   0   0   2   0   0   2   3   0   3   0   0   3   0   2   0   3   0   3   0    1   3   8
  2  2    2   0   0   0   2   0   0   2   3   0   3   0   0   3   1   2   0   3   0   3   0    2   3   8
Flight 2 : Passenger 0 checked
  2  2    2   0   0   0   2   0   0   2   3   0   3   0   0   3   1   2   0   3   0   3   0    1   4   9
  2  1    2   0   0   0   2   0   0   2   3   0   3   0   0   3   1   2   0   3   0   3   0    1   4   9
  2  2    2   0   0   0   2   0   0   2   3   0   3   0   0   3   1   2   0   3   0   3   0    1   4   9
  2  2    2   0   0   0   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    1   4   9
Flight 2 : Passenger 14 checked
  2  2    2   0   0   0   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    0   5  10
  2  3    2   0   0   0   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    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   0   0   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    0   5  10
  3  0    2   0   0   0   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    0   5  10
  3  0    2   0   0   1   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    1   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   0   1   2   0   0   2   3   0   3   0   0   3   2   2   0   3   0   3   0    1   5  10
  4  0    2   0   0   1   2   0   0   2   3   0   3   0   0   3   2   3   0   3   0   3   0    1   4  10
  4  0    2   0   0   1   2   0   0   3   3   0   3   0   0   3   2   3   0   3   0   3   0    1   3  10
  4  0    3   0   0   1   2   0   0   3   3   0   3   0   0   3   2   3   0   3   0   3   0    1   2  10
  4  0    3   0   0   1   3   0   0   3   3   0   3   0   0   3   2   3   0   3   0   3   0    1   1  10
  4  0    3   0   0   1   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   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   0   1   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   0  10
  1  0    3   0   0   1   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   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   0   1   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   0  10
  2  1    3   0   0   1   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   0  10
  2  2    3   0   0   1   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   0  10
  2  2    3   0   0   2   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   0  10
Flight 3 : Passenger 3 checked
  2  2    3   0   0   2   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    0   1  11
  2  1    3   0   0   2   3   0   0   3   3   0   3   0   0   3   3   3   0   3   0   3   0    0   1  11
  2  1    3   0   0   2   3   0   1   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   1  11
  2  2    3   0   0   2   3   0   1   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   1  11
  2  2    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   0    1   1  11
Flight 3 : Passenger 6 checked
  2  2    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   0    0   2  12
  2  1    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   0    0   2  12
  2  1    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   1    1   2  12
  2  2    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   1    1   2  12
  2  2    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    1   2  12
Flight 3 : Passenger 20 checked
  2  2    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    0   3  13
  2  1    3   0   0   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    0   3  13
  2  1    3   0   1   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    1   3  13
  2  2    3   0   1   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    1   3  13
  2  2    3   0   2   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    1   3  13
Flight 3 : Passenger 2 checked
  2  2    3   0   2   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    0   4  14
  2  1    3   0   2   2   3   0   2   3   3   0   3   0   0   3   3   3   0   3   0   3   2    0   4  14
  2  1    3   0   2   2   3   0   2   3   3   1   3   0   0   3   3   3   0   3   0   3   2    1   4  14
  2  2    3   0   2   2   3   0   2   3   3   1   3   0   0   3   3   3   0   3   0   3   2    1   4  14
  2  2    3   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   0   3   0   3   2    1   4  14
Flight 3 : Passenger 9 checked
  2  2    3   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   0   3   0   3   2    0   5  15
  2  3    3   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   0   3   0   3   2    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   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   0   3   0   3   2    0   5  15
  3  0    3   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   0   3   0   3   2    0   5  15
  3  0    3   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   1   3   0   3   2    1   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   0   2   2   3   0   2   3   3   2   3   0   0   3   3   3   1   3   0   3   2    1   5  15
  4  0    3   0   2   2   3   0   3   3   3   2   3   0   0   3   3   3   1   3   0   3   2    1   4  15
  4  0    3   0   2   2   3   0   3   3   3   2   3   0   0   3   3   3   1   3   0   3   3    1   3  15
  4  0    3   0   3   2   3   0   3   3   3   2   3   0   0   3   3   3   1   3   0   3   3    1   2  15
  4  0    3   0   3   3   3   0   3   3   3   2   3   0   0   3   3   3   1   3   0   3   3    1   1  15
  4  0    3   0   3   3   3   0   3   3   3   3   3   0   0   3   3   3   1   3   0   3   3    1   0  15
  4  0    3   0   3   3   3   1   3   3   3   3   3   0   0   3   3   3   1   3   0   3   3    2   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   0   3   3   3   1   3   3   3   3   3   0   0   3   3   3   1   3   0   3   3    2   0  15
  0  0    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   1   3   0   3   3    3   0  15
  1  0    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   1   3   0   3   3    3   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   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   1   3   0   3   3    3   0  15
  2  1    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   1   3   0   3   3    3   0  15
  2  2    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   1   3   0   3   3    3   0  15
  2  2    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    3   0  15
```

```
Flight 4 : Passenger 16 checked
  2  2    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    2   1  16
  2  1    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    2   1  16
  2  2    3   0   3   3   3   1   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    2   1  16
  2  2    3   0   3   3   3   2   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    2   1  16
Flight 4 : Passenger 5 checked
  2  2    3   0   3   3   3   2   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    1   2  17
  2  1    3   0   3   3   3   2   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    1   2  17
  2  2    3   0   3   3   3   2   3   3   3   3   3   0   1   3   3   3   2   3   0   3   3    1   2  17
  2  2    3   0   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    1   2  17
Flight 4 : Passenger 12 checked
  2  2    3   0   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    0   3  18
  2  1    3   0   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    0   3  18
  2  1    3   1   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    1   3  18
  2  2    3   1   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    1   3  18
  2  2    3   2   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    1   3  18
Flight 4 : Passenger 1 checked
  2  2    3   2   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    0   4  19
  2  1    3   2   3   3   3   2   3   3   3   3   3   0   2   3   3   3   2   3   0   3   3    0   4  19
  2  1    3   2   3   3   3   2   3   3   3   3   3   1   2   3   3   3   2   3   0   3   3    1   4  19
  2  2    3   2   3   3   3   2   3   3   3   3   3   1   2   3   3   3   2   3   0   3   3    1   4  19
  2  2    3   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   0   3   3    1   4  19
Flight 4 : Passenger 11 checked
  2  2    3   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   0   3   3    0   5  20
  2  3    3   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   0   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   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   0   3   3    0   5  20
  3  0    3   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   0   3   3    0   5  20
  3  0    3   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   1   3   3    1   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   2   3   3   3   2   3   3   3   3   3   2   2   3   3   3   2   3   1   3   3    1   5  20
  4  0    3   2   3   3   3   3   3   3   3   3   3   2   2   3   3   3   2   3   1   3   3    1   4  20
  4  0    3   2   3   3   3   3   3   3   3   3   3   3   2   3   3   3   2   3   1   3   3    1   3  20
  4  0    3   2   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   1   3   3    1   2  20
  4  0    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   1   3   3    1   1  20
  4  0    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3    1   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   3   3   3   3   3   1   3   3    1   0  20
  1  0    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   1   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   3   3   3   3   3   1   3   3    1   0  20
  2  1    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3    1   0  20
  2  2    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   1   3   3    1   0  20
  2  2    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3    1   0  20
Flight 5 : Passenger 18 checked
  2  2    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   3    0   1  21
  2  3    3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   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   3   3   3   3   2   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   3   3   3   3   2   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    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
```

Conclusão

Com a realização deste trabalho conseguimos perceber o funcionamento e a importância dos mecanismos associados à execução e sincronização de processos e threads.

Bibliografia

- Slides da disciplina
- Documentação fornecida pelo DoxyFile