

Relatório do 2º projeto de Sistemas Operativos (SO)

Projeto: Simulação de Ponte Aérea

Grupo: Guilherme Dias (nº 103128)
Tomás Almeida (nº 103300)

Data de entrega: 1 de Fevereiro 2022

Docentes: Nuno Lau (TP) e Guilherme Campos (P)

Índice

Introdução.....	3
Funções.....	4
Tabela Entidade/Função.....	5
Explicação e prints do código.....	6
Resultados/Testes.....	17
Conclusão.....	22
Bibliografia.....	22

* -> Anexo inclui o código `multi_ordered_tree`, bem como o código matlab para a criação de todos os gráficos, e alguns testes de código.

Introdução

Este trabalho tem como objetivo encontrar uma solução em C, que através da utilização de semáforos, simula uma ponte aérea e tudo o que a envolve, através da alteração de estados do Piloto, Hospedeira de bordo e dos passageiros em si.

Para tal foi necessário completar o código base fornecido, complementando as funções dos ficheiros correspondentes ao Piloto(semSharedMemPilot.c), Hospedeira(semSharedMemHostess.c), e Passageiros(semSharedMemPassenger.c)

Funções

Foram completadas 10 funções para cumprir os objetivos do trabalho.

Relacionadas com a Hospedeira:

- `waitForNextFlight()`
- `waitForPassenger()`
- `checkPassport()`
- `signalReadyToFlight()`

Relacionadas com o Piloto:

- `flight()`
- `signalReadyForBoarding()`
- `waitUntilReadyToFlight()`
- `dropPassengersAtTarget()`

Relacionadas com os Passageiros:

- `waitInQueue()`
- `waitUntilDestination()`

Tabela Entidade/Função

- Tabela utilizada durante a realização do projeto para melhor entendimento sobre quais entidades fazem up ou down, e em que funções é que isso será verificado.

Semáforo	Entidade Down	Função Down	Entidade Up	Função Up
passengersInQueue	Hostess	waitForPassengers()	Passengers	waitInQueue()
passengersWaitInQueue	Passengers	waitInQueue()	Hostess	checkPassport()
passengersWaitInFlight	Passengers	waitUntilDestination()	Pilot	dropPassengersAtTarget()
readyForBoarding	Hostess	waitForNextFlight()	Pilot	signalReadyForBoarding()
readyToFlight	Pilot	waitUntilReadyToFlight()	Hostess	signalReadyToFlight()
idShown	Hostess	checkPassport()	Passengers	waitInQueue()
planeEmpty	Pilot	dropPassengersAtTarget()	Passengers	waitUntilDestination()

Explicação e prints do código

Entidade Hostess:

- **waitForNextFlight()**

Nesta função a hospedeira espera que o avião esteja pronto para o embarque. É dada entrada na zona crítica(Down no mutex) para alterar o estado da hospedeira para `WAIT_FOR_FLIGHT`, o estado é guardado através do `saveState()`. De seguida sai da zona crítica(Up no mutex) e dá up no semáforo `readyForBoarding`.

```
142 static void waitForNextFlight ()
143 {
144     if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
145         perror ("error on the up operation for semaphore access (HT)");
146         exit (EXIT_FAILURE);
147     }
148
149     /* insert your code here */
150     sh->fSt.st.hostessStat = WAIT_FOR_FLIGHT;
151     saveState(nFic,&sh->fSt);
152
153     if (semUp (semgid, sh->mutex) == -1)                                    /* exit critical region */
154     { perror ("error on the down operation for semaphore access (HT)");
155       exit (EXIT_FAILURE);
156     }
157
158     /* insert your code here */
159
160     if (semDown(semgid, sh->readyForBoarding) == -1){
161         perror ("error on the down operation for semaphore access (HT)");
162         exit (EXIT_FAILURE);
163     }
164 }
```

- **waitForPassenger()**

Nesta função a hospedeira espera pelos passageiros chegarem ao aeroporto. É dada entrada na zona crítica(Down no mutex) para alterar o estado da hospedeira para `WAIT_FOR_PASSENGER`, o estado é guardado através do `saveState()`. De seguida sai da zona crítica(Up no mutex) e dá up no semáforo `passengersInQueue`.

```

173 static void waitForPassenger ()
174 {
175     if (semDown (semgid, sh->mutex) == -1)                                /* enter critical region */
176     { perror ("error on the up operation for semaphore access (HT)");
177         exit (EXIT_FAILURE);
178     }
179
180     /* insert your code here */
181     sh->fSt.st.hostessStat = WAIT_FOR_PASSENGER;
182     saveState(nFic, &sh->fSt);
183
184     if (semUp (semgid, sh->mutex) == -1) {                                /* exit critical region */
185         perror ("error on the down operation for semaphore access (HT)");
186         exit (EXIT_FAILURE);
187     }
188
189     /* insert your code here */
190     if (semDown(semgid, sh->passengersInQueue) == -1){
191         perror ("error on the down operation for semaphore access (HT)");
192         exit (EXIT_FAILURE);
193     }
194
195 }

```

- **checkPassport ()**

Nesta função a hospedeira verifica o passaporte do passageiro e espera que este mostre o bilhete de identidade. A função começa dando up ao semáforo passengersWaitInQueue, é depois dada entrada na zona crítica(Down no mutex) para alterar o estado da hospedeira para CHECK_PASSPORT, o estado é guardado através do saveState(). De seguida sai da zona crítica(Up no mutex), é dado um down no semáforo idShown, voltamos a entrar numa zona crítica(Down no mutex), damos save do passageiro verificado pela hospedeira, fazemos então o update do número de passageiro na fila, número de passageiros no voo, número total de embarcados, o estado é guardado através do saveState(), é verificado se o avião está cheio, ou se não há mais passageiros para entrar no avião, e por fim dá-se up no mutex e saímos da zona crítica, retornando se o last é verdadeiro ou falso.

```

210 static bool checkPassport()
211 {
212     bool last;
213
214     /* insert your code here */
215     if (semUp(semgid, sh->passengersWaitInQueue) == -1){
216         perror ("error on the down operation for semaphore access (HT)");
217         exit (EXIT_FAILURE);
218     }
219
220     if (semDown (semgid, sh->mutex) == -1) { /* enter critical region */
221         perror ("error on the up operation for semaphore access (HT)");
222         exit (EXIT_FAILURE);
223     }
224
225     /* insert your code here */
226
227     sh->fSt.st.hostessStat = CHECK_PASSPORT;
228     saveState(nFic, &sh->fSt);
229
230     if (semUp (semgid, sh->mutex) == -1) { /* exit critical region */
231         perror ("error on the up operation for semaphore access (HT)");
232         exit (EXIT_FAILURE);
233     }
234
235     /* insert your code here */
236
237     if (semDown (semgid, sh->idShown) == -1){
238         perror ("error on the up operation for semaphore access (HT)");
239         exit (EXIT_FAILURE);
240     }
241
242     if (semDown (semgid, sh->mutex) == -1) { /* enter critical region */
243         perror ("error on the up operation for semaphore access (HT)");
244         exit (EXIT_FAILURE);
245     }

```



```

247  /* insert your code here */
248  savePassengerChecked(nFic, &sh->fSt);
249
250  sh->fSt.nPassInFlight++; // numero de passageiros no voo aumenta 1
251  sh->fSt.nPassInQueue--; // numero de passageiros na fila diminui 1
252  sh->fSt.totalPassBoarded++; //numero total de passageiros que embarcaram aumenta 1
253  saveState(nFic, &sh->fSt);
254
255  if ((nPassengersInFlight() >= MINFC && nPassengersInQueue() == 0) || nPassengersInFlight() == MAXFC || sh->fSt.totalPassBoarded == N){
256      last = true; // condições para checkar se este é o último passageiro a embarcar.
257  }
258  else{
259      last = false;
260      sh->fSt.st.hostessStat = WAIT_FOR_PASSENGER;
261      saveState(nFic, &sh->fSt);
262  }
263
264  if (semUp (semgid, sh->mutex) == -1) {                                     /* exit critical region */
265      perror ("error on the up operation for semaphore access (HT)");
266      exit (EXIT_FAILURE);
267  }
268
269  /* insert your code here */
270  //No need for that.
271  return last;
272  }

```

- **signalReadyToFlight()**

Nesta função a hospedeira informa o piloto que está tudo pronto para a descolagem. É dada entrada na zona crítica(Down no mutex) para registrar o número de passageiros no avião e para alterar o estado da hospedeira para `READY_TO_FLIGHT`, o estado é guardado através do `saveState()` e é também salvo o estado do voo(Departed). De seguida é verificado se o número de passageiros embarcados em todos os voos é igual ao número de passageiros existentes(21), e se isso se verificar então é dado o airlift como finalizado. Sai da zona crítica(Up no mutex) e dá up no semáforo `readyToFlight`.

```

293 void signalReadyToFlight()
294 {
295     if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
296         perror ("error on the up operation for semaphore access (HT)");
297         exit (EXIT_FAILURE);
298     }
299
300     /* insert your code here */
301
302     sh->fSt.nPassengersInFlight[sh->fSt.nFlight-1] = nPassengersInFlight(); // atualiza nº de passageiros neste voo
303     sh->fSt.st.hostessStat = READY_TO_FLIGHT;
304     saveState(nFic, &sh->fSt);
305     saveFlightDeparted(nFic, &sh->fSt);
306
307     if (sh->fSt.totalPassBoarded == N)
308         sh->fSt.finished = true;
309
310     if (semUp (semgid, sh->mutex) == -1) {                                    /* exit critical region */
311         perror ("error on the up operation for semaphore access (HT)");
312         exit (EXIT_FAILURE);
313     }
314
315     /* insert your code here */
316
317     if (semUp (semgid, sh->readyToFlight) == -1){
318         perror ("error on the up operation for semaphore access (HT)");
319         exit (EXIT_FAILURE);
320     }
321 }

```

Entidade Pilot:

- **flight()**

Nesta função o piloto leva os passageiros para o destino ou então leva o avião de volta para o aeroporto de origem. É dada entrada na zona crítica(Down no mutex) para alterar o estado do piloto para `FLYING` (se `go==true`, o que significa que o piloto está a voar para um destino) ou para `FLYING_BACK` (se `go==false`, o que significa que o piloto está a voar para a origem) , o estado é guardado através do `saveState()`. De seguida sai da zona crítica(Up no mutex).

```

137 static void flight (bool go)
138 {
139     if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
140         perror ("error on the up operation for semaphore access (PT)");
141         exit (EXIT_FAILURE);
142     }
143
144     /* insert your code here */
145     if (go == true){ //se go == true, avião está a voar para o destino, else está a voltar.
146         sh->fSt.st.pilotStat = FLYING;
147         saveState(nFic, &sh->fSt);
148     }
149
150     else{
151         sh->fSt.st.pilotStat = FLYING_BACK;
152         saveState(nFic, &sh->fSt);
153     }
154
155     if (semUp (semgid, sh->mutex) == -1) {                                    /* exit critical region */
156         perror ("error on the up operation for semaphore access (PT)");
157         exit (EXIT_FAILURE);
158     }
159
160     usleep((unsigned int) floor ((MAXFLIGHT * random ()) / RAND_MAX + 100.0));
161 }

```

- **signalReadyForBoarding()**

Nesta função o piloto informa a hospedeira que o avião está pronto para o embarque. É dada entrada na zona crítica(Down no mutex) para alterar o estado do piloto para `READY_FOR_BOARDING`, o estado é guardado através do `saveState()`. É adicionado um ao número do voo e é salvo o estado(`saveStartBoarding`), de seguida sai da zona crítica(Up no mutex) e dá up no semáforo `readyForBoarding`.

```

171 static void signalReadyForBoarding ()
172 {
173     if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
174         perror ("error on the up operation for semaphore access (PT)");
175         exit (EXIT_FAILURE);
176     }
177
178     /* insert your code here */
179     sh->fSt.st.pilotStat = READY_FOR_BOARDING;
180     saveState(nFic, &sh->fSt);
181     sh->fSt.nFlight++; // numero do voo sobe 1
182     saveStartBoarding(nFic, &sh->fSt);
183
184     if (semUp (semgid, sh->mutex) == -1) {                                    /* exit critical region */
185         perror ("error on the up operation for semaphore access (PT)");
186         exit (EXIT_FAILURE);
187     }
188
189     /* insert your code here */
190     if (semUp(semgid, sh->readyForBoarding) == -1){
191         perror ("error on the up operation for semaphore access (PT)");
192         exit (EXIT_FAILURE);
193     }
194
195 }

```

- **waitUntilReadyToFlight()**

Nesta função o piloto espera que os passageiros entrem no avião . É dada entrada na zona crítica(Down no mutex) para alterar o estado do piloto para WAITING_FOR_BOARDING, o estado é guardado através do saveState(), e de seguida sai da zona crítica(Up no mutex) e é dado down no semáforo readyToFlight.

```

204 static void waitUntilReadyToFlight ()
205 {
206     if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
207         perror ("error on the up operation for semaphore access (PT)");
208         exit (EXIT_FAILURE);
209     }
210
211     /* insert your code here */
212     sh->fSt.st.pilotStat = WAITING_FOR_BOARDING;
213     saveState(nFic, &sh->fSt);
214
215     if (semUp (semgid, sh->mutex) == -1) {                                    /* exit 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     if (semDown(semgid, sh->readyToFlight) == -1){
222         perror ("error on the down operation for semaphore access (PT)");
223         exit (EXIT_FAILURE);
224     }
225 }

```

- **dropPassengersAtTarget ()**

Nesta função o piloto deixa sair os passageiros do avião . É dada entrada na zona crítica(Down no mutex), mudança do estado do voo para Arrived(saveFlightArrived), é alterado o estado do piloto para DROPPING_PASSENGERS, o estado é guardado através do saveState(), de seguida sai da zona crítica(Up no mutex) e é dado up no semáforo passengersWaitInFlight por cada passageiro no voo. É feito um down no semáforo planeEmpty, e voltamos a entrar numa zona crítica(Down no mutex), é salvo o estado do voo, returning(saveFlightReturning), e de seguida sai da zona crítica(Up no mutex) .

```

235 static void dropPassengersAtTarget ()
236 {
237     if (semDown (semgid, sh->mutex) == -1) {                                /* enter critical region */
238         perror ("error on the down operation for semaphore access (PT)");
239         exit (EXIT_FAILURE);
240     }
241
242     /* insert your code here */
243     saveFlightArrived(nFic, &sh->fSt); //voo chegou ao destino
244
245     sh->fSt.st.pilotStat = DROPPING_PASSENGERS;
246     saveState(nFic, &sh->fSt);
247
248     if (semUp (semgid, sh->mutex) == -1) {                                    /* exit critical region */
249         perror ("error on the up operation for semaphore access (PT)");
250         exit (EXIT_FAILURE);
251     }
252
253     /* insert your code here */
254

```

```

253     /* insert your code here */
254
255
256     int x = 0;
257     while(x < sh->fSt.nPassengersInFlight[sh->fSt.nFlight-1]){
258         x++;
259         if (semUp(semgid, sh->passengersWaitInFlight) == -1) { //up no semáforo passengersWaitInFlight por cada passageiro no voo
260             perror ("error on the down operation for semaphore access (PT)");
261             exit (EXIT_FAILURE);
262         }
263
264     }
265
266     if (semDown(semgid, sh->planeEmpty) == -1){
267         perror ("error on the down operation for semaphore access (PT)");
268         exit (EXIT_FAILURE);
269     }
270
271     if (semDown (semgid, sh->mutex) == -1){                                     /* enter critical region */
272         perror ("error on the down operation for semaphore access (PT)");
273         exit (EXIT_FAILURE);
274     }
275
276     /* insert your code here */
277     saveFlightReturning(nFic, &sh->fSt);
278
279     if (semUp (semgid, sh->mutex) == -1) {                                     /* exit critical region */
280         perror ("error on the up operation for semaphore access (PT)");
281         exit (EXIT_FAILURE);
282     }
283 }

```

Entidade Passenger:

- **waitInQueue()**

Nesta função o passageiro deixa esperar pela sua vez de ser verificado pela hospedeira de bordo. É dada entrada na zona crítica(Down no mutex), adiciona-se 1 ao número de passageiros em queue, o estado do passageiro em questão(relacionado com o seu ID) é alterado para `IN_QUEUE`, depois o estado é guardado através do `saveState()`. É feito um up no mutex para sair da zona crítica, um up no semáforo `passengersInQueue`, e um down no semáforo `passengersWaitInQueue`. É realizado um down no mutex para voltar a entrar numa zona crítica, é feito o update do ID do último passageiro a ser checkado pela hospedeira, e é alterado o estado do passageiro para `IN_FLIGHT`, o estado é guardado através do `saveState()`, e de seguida sai da zona crítica(Up no mutex) e é dado up no semáforo `idShown`.

```

141 static void waitInQueue (unsigned int passengerId)
142 {
143     if (semDown (semgid, sh->mutex) == -1) {                                     /* enter critical region */
144         perror ("error on the down operation for semaphore access (PG)");
145         exit (EXIT_FAILURE);
146     }
147
148     /* insert your code here */
149
150     sh->fSt.nPassInQueue++; //adiciona 1 passageiro à queue
151     sh->fSt.st.passengerStat[passengerId] = IN_QUEUE; //estado do passageiro em questão(relacionado com o seu ID) é alterado
152     saveState(nFic, &sh->fSt);
153
154     if (semUp (semgid, sh->mutex) == -1)                                         /* exit critical region */
155     { perror ("error on the up operation for semaphore access (PG)");
156       exit (EXIT_FAILURE);
157     }
158
159     /* insert your code here */
160     if (semUp (semgid, sh->passengersInQueue) == -1)
161     {   perror ("error on the up operation for semaphore access (PG)");
162         exit (EXIT_FAILURE);
163     }
164
165
166     if (semDown (semgid, sh->passengersWaitInQueue) == -1)
167     {   perror ("error on the up operation for semaphore access (PG)");
168         exit (EXIT_FAILURE);
169     }
170
171     if (semDown (semgid, sh->mutex) == -1) {                                     /* enter critical region */
172         perror ("error on the down operation for semaphore access (PG)");
173         exit (EXIT_FAILURE);
174     }
175
176     /* insert your code here */
177     sh->fSt.passengerChecked = passengerId; //update do ID do último passageiro a ser checkado pela hospedeira
178     sh->fSt.st.passengerStat[passengerId] = IN_FLIGHT; //estado do passageiro em questão(relacionado com o seu ID) é alterado
179     saveState(nFic, &sh->fSt);
180
181
182
183
184     if (semUp (semgid, sh->mutex) == -1) {                                     /* exit critical region */
185         perror ("error on the down operation for semaphore access (PG)");
186         exit (EXIT_FAILURE);
187     }
188
189     /* insert your code here */
190     if (semUp (semgid, sh->idShown) == -1){
191         perror ("error on the up operation for semaphore access (PG)");
192         exit (EXIT_FAILURE);
193     }
194 }

```


- waitUntilDestination()

Nesta função o passageiro espera que o voo termine e chegue ao destino. É realizado um down no semáforo passengersWaitInFlight e de seguida é dada entrada na zona crítica(Down no mutex), retira-se 1 ao número de passageiros no voo, o estado do passageiro em questão(relacionado com o seu ID) é alterado para AT_DESTINATION, depois o estado é guardado através do saveState(). Se o número de passageiros no voo for igual a 0, é feito um up do semáforo planeEmpty, e é feito então um up no mutex para sair da zona crítica.

```
207 static void waitUntilDestination (unsigned int passengerId)
208 {
209     /* insert your code here */
210     if (semDown (semgid, sh->passengersWaitInFlight) == -1){
211         perror ("error on the up operation for semaphore access (PG)");
212         exit (EXIT_FAILURE);
213     }
214     if (semDown (semgid, sh->mutex) == -1) {                                     /* enter critical region */
215         perror ("error on the down operation for semaphore access (PG)");
216         exit (EXIT_FAILURE);
217     }
218     /* insert your code here */
219     sh->fSt.nPassInFlight--; //-1 passageiro no voo
220     sh->fSt.st.passengerStat[passengerId] = AT_DESTINATION; //passageiro chegou ao destino
221     saveState(nFic, &sh->fSt);
222     if (sh->fSt.nPassInFlight == 0) {
223         if (semUp(semgid, sh->planeEmpty) == -1) { //se o número de passageiros no voo for 0, avião está vazio
224             perror ("error on the down operation for semaphore access (PG)");
225             exit (EXIT_FAILURE);
226         }
227     }
228     if (semUp (semgid, sh->mutex) == -1) {                                     /* exit critical region */
229         perror ("error on the down operation for semaphore access (PG)");
230         exit (EXIT_FAILURE);
231     }
232 }
233 }
```

Resultados/Testes

Print de uma run do programa:

```
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$ ./probSemSharedMemAirLift
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	
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	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
2	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	
2	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	
2	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	
2	2	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	
2	2	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	
Flight 1 : Passenger 4 checked																										
2	2	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	
2	1	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	
2	1	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	
2	2	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	
2	2	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	1	1	
Flight 1 : Passenger 11 checked																										
2	2	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	2	
2	1	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	2	
2	1	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	2	
2	1	0	0	0	0	2	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	1	2	2	
2	2	0	0	0	0	2	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	1	2	2	
2	2	0	0	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	2	2	
Flight 1 : Passenger 8 checked																										
2	2	0	0	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	3	3	
2	1	0	0	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	3	3	
2	1	0	0	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	3	3	
2	1	0	1	0	0	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	3	3	
2	1	0	1	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	2	3	3	
2	2	0	1	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	2	3	3	
2	2	0	2	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	2	3	3	
Flight 1 : Passenger 1 checked																										
2	2	0	2	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	4	4	
2	1	0	2	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	4	4	
2	1	0	2	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	4	4	
2	2	0	2	0	1	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	4	4	
2	2	0	2	0	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	1	4	4	
Flight 1 : Passenger 3 checked																										
2	2	0	2	0	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	5	5	
2	3	0	2	0	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	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	2	0	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	5	5	
3	0	0	2	0	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	5	5	
3	0	0	2	0	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	1	0	0	1	5	5	
3	0	0	2	1	2	2	0	0	0	2	0	0	2	0	0	0	0	0	0	1	0	0	2	5	5	
3	0	0	2	1	2	2	0	0	0	2	0	0	2	0	1	0	0	0	0	1	0	0	3	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	2	1	2	2	0	0	0	2	0	0	2	0	1	0	0	0	0	1	0	0	3	5	5	
4	0	0	2	1	2	3	0	0	0	2	0	0	2	0	1	0	0	0	0	1	0	0	3	4	5	
4	0	0	2	1	2	3	0	0	0	2	0	0	3	0	1	0	0	0	0	1	0	0	3	3	5	
4	0	0	2	1	2	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	2	5	
4	0	0	3	1	2	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	1	5	
4	0	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	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	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	0	5
1	0	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	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	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	0	5
2	1	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	0	5
2	2	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	1	0	0	3	0	5
2	2	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	2	0	0	3	0	5

Flight 2 : Passenger 18 checked

2	2	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	2	0	0	2	1	6
2	1	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	2	0	0	2	1	6
2	1	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	0	2	0	0	2	1	6
2	1	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	3	1	6
2	2	0	3	1	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	3	1	6
2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	3	1	6

Flight 2 : Passenger 2 checked

2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	2	2	7
2	1	0	3	2	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	2	2	7
2	1	0	3	2	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	2	2	7
2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	1	0	0	0	1	2	0	0	2	2	7
2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	0	0	1	2	0	0	2	2	7

Flight 2 : Passenger 13 checked

2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	0	0	1	2	0	0	1	3	8
2	1	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	0	0	1	2	0	0	1	3	8
2	1	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	0	0	1	2	0	0	1	3	8
2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	0	0	1	2	0	0	1	3	8
2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	0	0	2	2	0	0	1	3	8
2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	1	0	2	2	0	0	2	3	8

Flight 2 : Passenger 17 checked

2	2	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	1	0	2	2	0	0	1	4	9
2	1	0	3	2	3	3	0	0	0	3	0	0	3	0	2	0	1	0	2	2	0	0	1	4	9
2	1	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	1	0	2	2	0	0	2	4	9
2	1	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	1	0	2	2	0	0	2	4	9
2	2	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	1	0	2	2	0	0	2	4	9
2	2	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	2	4	9

Flight 2 : Passenger 15 checked

2	2	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	1	5	10
2	1	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	1	5	10
2	1	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	1	5	10
2	2	0	3	2	3	3	1	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	1	5	10
2	2	0	3	2	3	3	2	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	1	5	10

Flight 2 : Passenger 5 checked

2	2	0	3	2	3	3	2	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	0	6	11
2	3	0	3	2	3	3	2	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	0	6	11

Flight 2 : Departed with 6 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	3	2	3	3	2	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	0	6	11
3	0	0	3	2	3	3	2	0	0	3	0	0	3	0	2	0	2	0	2	2	0	0	0	6	11
3	0	0	3	2	3	3	2	0	1	3	0	0	3	0	2	0	2	0	2	2	0	0	1	6	11

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	0	3	2	3	3	2	0	1	3	0	0	3	0	2	0	2	0	2	2	0	0	1	6	11
4	0	0	3	3	3	3	2	0	1	3	0	0	3	0	2	0	2	0	2	2	0	0	1	5	11
4	0	0	3	3	3	3	2	0	1	3	0	0	3	0	2	0	2	0	2	3	0	0	1	4	11
4	0	0	3	3	3	3	2	0	1	3	0	0	3	0	3	0	2	0	2	3	0	0	1	3	11
4	0	0	3	3	3	3	3	0	1	3	0	0	3	0	3	0	2	0	2	3	0	0	1	2	11
4	0	0	3	3	3	3	3	0	1	3	0	0	3	0	3	0	2	0	3	3	0	0	1	1	11
4	0	0	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	0	1	0	11
4	0	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	0	2	0	11

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	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	0	2	0	11
0	0	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	1	3	0	11
1	0	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	1	3	0	11

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	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	1	3	0	11
2	1	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	1	3	0	11
2	2	1	3	3	3	3	3	0	1	3	0	0	3	0	3	0	3	0	3	3	0	1	3	0	11
2	2	1	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	3	0	11

Flight 3 : Passenger 7 checked

2	2	1	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	2	1	12
2	1	1	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	2	1	12
2	1	1	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	2	1	12
2	2	1	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	2	1	12
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	2	1	12

Flight 3 : Passenger 0 checked

2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	1	2	13
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	1	2	13
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	1	2	13
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	1	1	2	13
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	2	1	2	13

Flight 3 : Passenger 20 checked

2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	2	0	3	14
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	2	0	3	14
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	0	2	0	3	14
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	1	2	1	3	14
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	1	2	1	3	14
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	2	2	1	3	14

Flight 3 : Passenger 19 checked

2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	2	2	0	4	15
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	2	2	0	4	15
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	0	3	3	2	2	0	4	15
2	1	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	1	3	3	2	2	1	4	15
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	1	3	3	2	2	1	4	15
2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	2	3	3	2	2	1	4	15

Flight 3 : Passenger 16 checked

2	2	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	2	3	3	2	2	0	5	16
2	3	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	2	3	3	2	2	0	5	16

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	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	2	3	3	2	2	0	5	16
3	0	2	3	3	3	3	3	0	2	3	0	0	3	0	3	0	3	2	3	3	2	2	0	5	16
3	0	2	3	3	3	3	3	0	2	3	0	0	3	0	3	1	3	2	3	3	2	2	1	5	16

```

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    2 3 3 3 3 3 0 2 3 0 0 3 0 3 1 3 2 3 3 2 2 1 5 16
4 0    3 3 3 3 3 3 0 2 3 0 0 3 0 3 1 3 2 3 3 2 2 1 4 16
4 0    3 3 3 3 3 3 0 3 3 0 0 3 0 3 1 3 2 3 3 2 2 1 3 16
4 0    3 3 3 3 3 3 0 3 3 0 0 3 0 3 1 3 2 3 3 3 2 1 2 16
4 0    3 3 3 3 3 3 0 3 3 0 0 3 0 3 1 3 2 3 3 3 3 1 1 16
4 0    3 3 3 3 3 3 0 3 3 0 0 3 0 3 1 3 3 3 3 3 3 1 0 16

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 3 3 3 0 3 3 0 0 3 0 3 1 3 3 3 3 3 3 1 0 16
0 0    3 3 3 3 3 3 0 3 3 0 1 3 0 3 1 3 3 3 3 3 3 2 0 16
1 0    3 3 3 3 3 3 0 3 3 0 1 3 0 3 1 3 3 3 3 3 3 2 0 16

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 3 3 3 0 3 3 0 1 3 0 3 1 3 3 3 3 3 3 2 0 16
2 1    3 3 3 3 3 3 0 3 3 0 1 3 0 3 1 3 3 3 3 3 3 2 0 16
2 1    3 3 3 3 3 3 0 3 3 1 1 3 0 3 1 3 3 3 3 3 3 3 0 16
2 2    3 3 3 3 3 3 0 3 3 1 1 3 0 3 1 3 3 3 3 3 3 3 0 16
2 2    3 3 3 3 3 3 0 3 3 1 1 3 0 3 2 3 3 3 3 3 3 3 0 16

Flight 4 : Passenger 14 checked
2 2    3 3 3 3 3 3 0 3 3 1 1 3 0 3 2 3 3 3 3 3 3 2 1 17
2 1    3 3 3 3 3 3 0 3 3 1 1 3 0 3 2 3 3 3 3 3 3 2 1 17
2 1    3 3 3 3 3 3 0 3 3 1 1 3 0 3 2 3 3 3 3 3 3 2 1 17
2 2    3 3 3 3 3 3 0 3 3 1 1 3 0 3 2 3 3 3 3 3 3 2 1 17
2 2    3 3 3 3 3 3 0 3 3 1 2 3 0 3 2 3 3 3 3 3 3 2 1 17

Flight 4 : Passenger 10 checked
2 2    3 3 3 3 3 3 0 3 3 1 2 3 0 3 2 3 3 3 3 3 3 1 2 18
2 1    3 3 3 3 3 3 0 3 3 1 2 3 0 3 2 3 3 3 3 3 3 1 2 18
2 1    3 3 3 3 3 3 0 3 3 1 2 3 0 3 2 3 3 3 3 3 3 1 2 18
2 2    3 3 3 3 3 3 0 3 3 1 2 3 0 3 2 3 3 3 3 3 3 1 2 18
2 2    3 3 3 3 3 3 0 3 3 2 2 3 0 3 2 3 3 3 3 3 3 1 2 18

Flight 4 : Passenger 9 checked
2 2    3 3 3 3 3 3 0 3 3 2 2 3 0 3 2 3 3 3 3 3 3 0 3 19
2 1    3 3 3 3 3 3 0 3 3 2 2 3 0 3 2 3 3 3 3 3 3 0 3 19
2 1    3 3 3 3 3 3 0 3 3 2 2 3 0 3 2 3 3 3 3 3 3 0 3 19
2 1    3 3 3 3 3 3 0 3 3 2 2 3 1 3 2 3 3 3 3 3 3 1 3 19
2 2    3 3 3 3 3 3 0 3 3 2 2 3 1 3 2 3 3 3 3 3 3 1 3 19
2 2    3 3 3 3 3 3 0 3 3 2 2 3 2 3 2 3 3 3 3 3 3 1 3 19

Flight 4 : Passenger 12 checked
2 2    3 3 3 3 3 3 0 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 4 20
2 1    3 3 3 3 3 3 0 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 4 20
2 1    3 3 3 3 3 3 0 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 4 20
2 1    3 3 3 3 3 3 1 3 3 2 2 3 2 3 2 3 3 3 3 3 3 1 4 20
2 2    3 3 3 3 3 3 1 3 3 2 2 3 2 3 2 3 3 3 3 3 3 1 4 20
2 2    3 3 3 3 3 3 2 3 3 2 2 3 2 3 2 3 3 3 3 3 3 1 4 20

Flight 4 : Passenger 6 checked
2 2    3 3 3 3 3 3 2 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 5 21
2 3    3 3 3 3 3 3 2 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 5 21

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
3 3    3 3 3 3 3 3 2 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 5 21

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 3    3 3 3 3 3 3 2 3 3 2 2 3 2 3 2 3 3 3 3 3 3 0 5 21
4 3    3 3 3 3 3 3 2 3 3 2 2 3 2 3 3 3 3 3 3 3 3 0 4 21
4 3    3 3 3 3 3 3 2 3 3 2 3 3 2 3 3 3 3 3 3 3 3 0 3 21
4 3    3 3 3 3 3 3 2 3 3 3 3 3 2 3 3 3 3 3 3 3 3 0 2 21
4 3    3 3 3 3 3 3 3 3 3 3 3 3 2 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

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

```

AirLift result

AirLift used 4 Flights

Flight 1 took 5 passengers

Flight 2 took 6 passengers

Flight 3 took 5 passengers

Flight 4 took 5 passengers

gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run\$ █

Alguns prints de AirLift results:

```
AirLift result
AirLift used 4 Flights
Flight 1 took 9 passengers
Flight 2 took 5 passengers
Flight 3 took 5 passengers
Flight 4 took 2 passengers
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$
```

```
AirLift result
AirLift used 3 Flights
Flight 1 took 10 passengers
Flight 2 took 6 passengers
Flight 3 took 5 passengers
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$
```

```
AirLift result
AirLift used 4 Flights
Flight 1 took 10 passengers
Flight 2 took 5 passengers
Flight 3 took 5 passengers
Flight 4 took 1 passengers
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$
```

```
AirLift result
AirLift used 4 Flights
Flight 1 took 6 passengers
Flight 2 took 5 passengers
Flight 3 took 5 passengers
Flight 4 took 5 passengers
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$
```

```
AirLift result
AirLift used 4 Flights
Flight 1 took 5 passengers
Flight 2 took 10 passengers
Flight 3 took 5 passengers
Flight 4 took 1 passengers
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$
```

```
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
gui23dias@DESKTOP-VPI6ECP:/mnt/d/GitHubProjects/SOProject2/semaphore_airLift/run$
```

Conclusão

Após a realização deste trabalho, ficamos a saber mais sobre Semáforos e sobre sincronização de processos e threads. Ficamos também a perceber melhor o funcionamento das funções C e todos os meandros de escrever/programar em C.

Bibliografia

<https://www.geeksforgeeks.org/mutex-vs-semaphore/?ref=gcse>

<https://riptutorial.com/c/example/31715/semaphores>