laSalle
UNIVERSITAT RAMON LLULL

Sistemas Operativos 2024-2025

Sesión 9: Semáforos I 04/12/2024

Goals

In previous sessions, we worked on the concept of threads and forks for parallelism and concurrency of tasks. The problem with working with shared resources and multi-process systems is that two processes try to access the same resource simultaneously, thus generating unpredictable or random results. On the other hand, sometimes the correct sequencing of tasks (synchronization) must also be guaranteed. The goal of this session is to introduce Operating Systems students to one of the most common synchronization and mutual exclusion mechanisms: **semaphores**. With the development of this session, it is intended that the student clarify the theoretical concepts of the use of semaphores and solve different problems.

Motivation

More specifically, with this session, the student will exercise:

- Semaphore creation (*SEM_constructor, SEM_init*)
- Control and destruction of semaphores (*SEM_destructor*)
- Operations on semaphores (*SEM_signal, SEM_wait*)
- Our custom library semaphore.h
- Use of the mutex library to use semaphores with threads.

Previous documentation

To complete this session, it is recommended to read the following references:

SALVADOR, J. (2014). *Programació en UNIX per a pràctiques de Sistemes Operatius*.

STEVENS, W. R. & RAGO S.A. (2008). Advanced Programming in the UNIX Environment, 2nd edition.

TANENBAUM, A. S. (2009). Modern Operating Systems, 3rd Edition.

La Salle Moto GP

This year, the Motorcycle World Championship comes to La Salle, and we need your help to manage the qualifying session. The objective of the program is to simulate this qualifying session, record the best lap times of each rider, and determine the starting order for the Saturday and Sunday races.

Specifications

The program must simulate a qualifying session in which the riders try to improve their lap times. At the end, the program will display the final standings, which will determine the starting order of the riders.

Program Details:

The program should be started from the command line, receiving two parameters:

1. The first parameter will be the file containing the rider information, called pilots.txt.
2. The second will be the maximum number of riders that can be on track simultaneously.

The rider file must have the following format:

<first_name>,<last_name>,<number>,<team>\n, where each line represents a rider. Example content of the file:

Example of file content:

Francesco,Bagnaia,1,Ducati Lenovo Team

Enea,Bastianini,23,Ducati Lenovo Team

Luca,Marini,10,Repsol Honda Team

Once the program starts, it should read all the rider information from the specified file and load it into the system. Each line of the file contains the details of a rider, and all of them will be stored in memory. Next, the riders will start entering the track in groups, the size of which will be determined by the number specified as the second argument of the program, indicating the maximum number of riders that can be on track simultaneously. If the total number of riders exceeds this limit, those who cannot enter the track will have to wait their turn.

Once a rider enters the track, they can continue lapping and recording lap times until they complete a maximum of 5 laps. After the fifth lap, the rider must leave the track, allowing other riders to enter if they are waiting their turn.

The qualifying session has a total duration of 10 seconds, but each rider can only be on track for the time it takes them to complete their 5 laps. During each lap, a random lap time will be generated. If a lap time is better than their previous best time, this new time will be updated as the rider's best record.

Lap times will be composed of three sectors (sector1, sector2, and sector3), whose random times will be added together to obtain the total lap time.

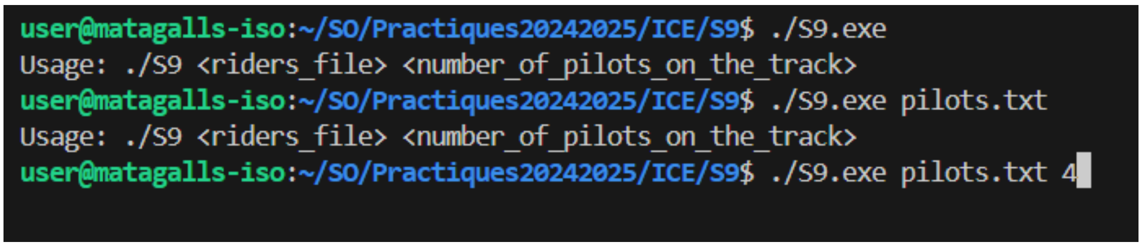
Instead of waiting a fixed time between laps, the time between one lap and the next will be the random lap time generated in each case. In other words, once the rider finishes a lap, they must "rest" for the equivalent of the lap time generated, simulating the time it takes them to complete the lap, before starting the next one. This process will continue until the rider completes their 5 laps, after which they will leave the track.

During the session: The program must display the updated standings of the riders every 3 seconds. These standings should be ordered according to the best time recorded by each rider, from fastest to slowest. The times should be displayed in a format that includes seconds, milliseconds, and nanoseconds. For example, the standings should look like this:

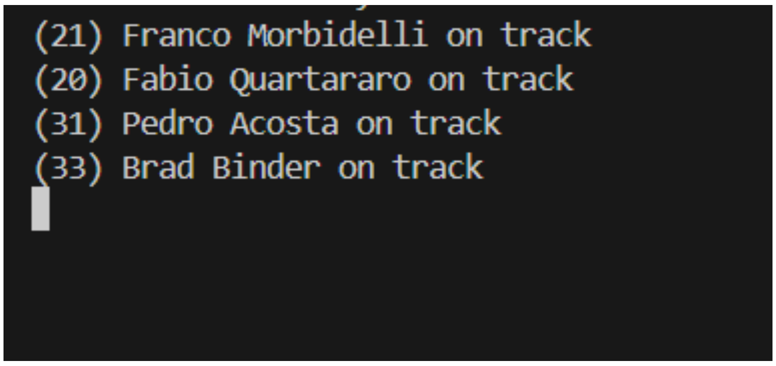
```
=== Actual Classification ===

1. (1) Francesco Bagnaia: 01:32:453

2. (23) Enea Bastianini: 01:34:672

3. (5) Johann Zarco: 01:35:123
```

Once the 10 seconds of the session have elapsed, or when all riders have completed their 5 laps, the session will end. At that point, the program must display the final standings of the riders, again ordered by their best times, from fastest to slowest. After displaying this final classification, the program will end its execution.

Execution Example:

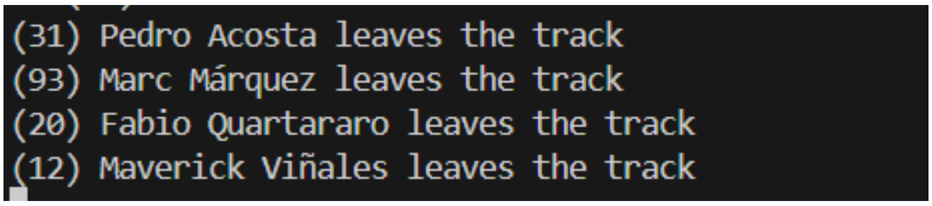


```
user@matagalls-iso:~/SO/Practiques20242025/ICE/S9$ ./S9.exe
Usage: ./S9 <riders_file> <number_of_pilots_on_the_track>
user@matagalls-iso:~/SO/Practiques20242025/ICE/S9$ ./S9.exe pilots.txt
Usage: ./S9 <riders_file> <number_of_pilots_on_the_track>
user@matagalls-iso:~/SO/Practiques20242025/ICE/S9$ ./S9.exe pilots.txt 4
```



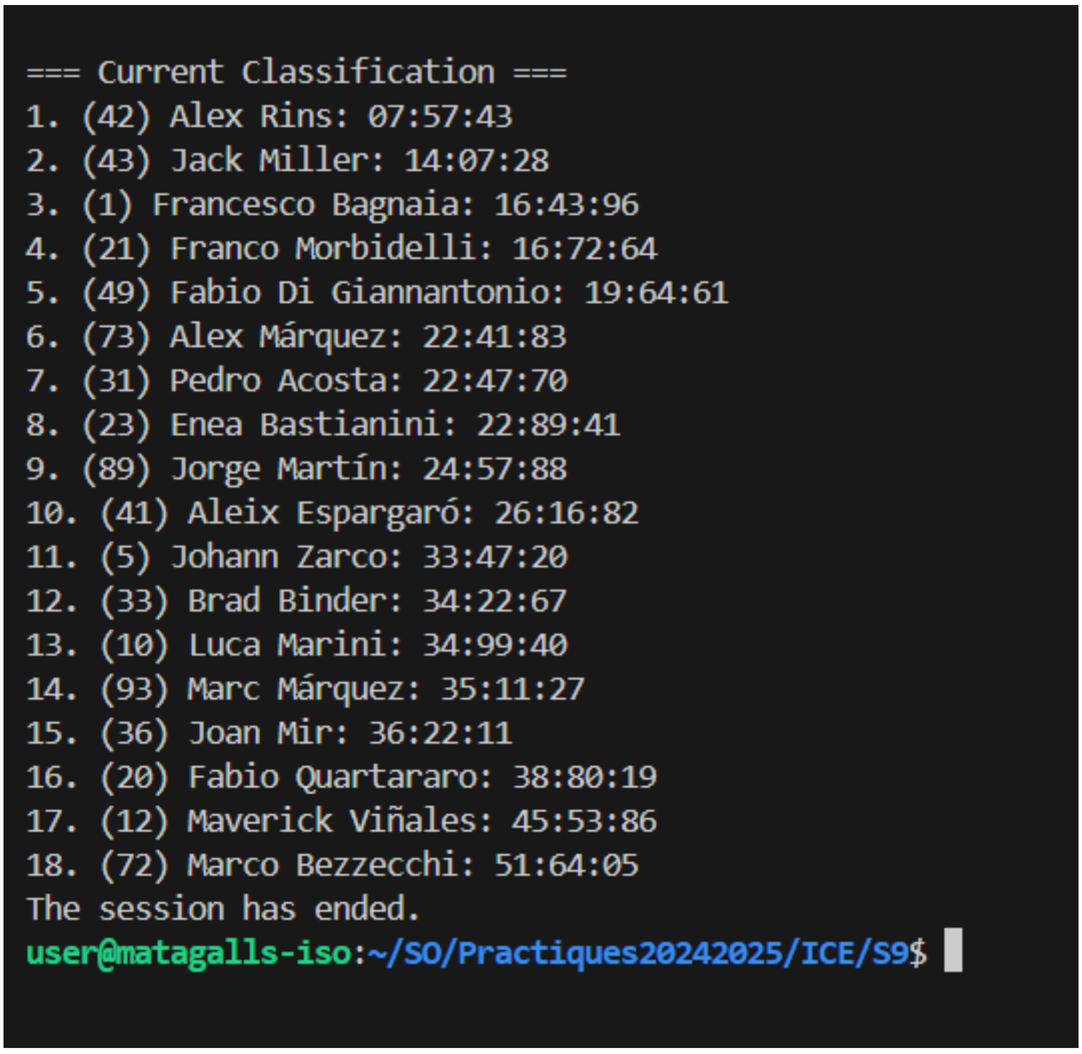
```
(21) Franco Morbidelli on track
(20) Fabio Quartararo on track
(31) Pedro Acosta on track
(33) Brad Binder on track
```

Every time a driver enters the track, the message shown in this image must be displayed.



```
(31) Pedro Acosta leaves the track
(93) Marc Márquez leaves the track
(20) Fabio Quartararo leaves the track
(12) Maverick Viñales leaves the track
```

Every time a pilot leaves the track, the message shown in this image must be displayed.



```
=== Current Classification ===
1. (42) Alex Rins: 07:57:43
2. (43) Jack Miller: 14:07:28
3. (1) Francesco Bagnaia: 16:43:96
4. (21) Franco Morbidelli: 16:72:64
5. (49) Fabio Di Giannantonio: 19:64:61
6. (73) Alex Márquez: 22:41:83
7. (31) Pedro Acosta: 22:47:70
8. (23) Enea Bastianini: 22:89:41
9. (89) Jorge Martín: 24:57:88
10. (41) Aleix Espargaró: 26:16:82
11. (5) Johann Zarco: 33:47:20
12. (33) Brad Binder: 34:22:67
13. (10) Luca Marini: 34:99:40
14. (93) Marc Márquez: 35:11:27
15. (36) Joan Mir: 36:22:11
16. (20) Fabio Quartararo: 38:80:19
17. (12) Maverick Viñales: 45:53:86
18. (72) Marco Bezzecchi: 51:64:05
The session has ended.
user@matagalls-iso:~/SO/Practiques20242025/ICE/S9$
```

Considerations:

- Mutual exclusion and/or synchronization must be guaranteed using semaphores.
- Check that you do not leave shared resources (semaphores, etc.) on the system. Use ipcs, ipcrm and other available scripts.
- All input and output must be done with file descriptors. The use of printf, scanf, FILE*, getchar, or similar is not allowed.
- Compile using the –Wall and –Wextra flags.
- Any practice that contains warnings will be unsuitable.
- All resources must be released, otherwise the practice will be unsuitable.
- At the beginning of the created .c, the logins and names and surnames of the students who are taking part in the session must be commented.
- The program must ignore the CTRL+ C in this way if the user presses CTRL+ C, the program must continue normally.