Goals

In the previous sessions, multiple new tools have been used: file descriptors, pipes, sockets, signals, forks, threads, semaphores, shared memory, message queues, etc.

In this last session, the aim is to give a global vision of the tools used by carrying out a globalizing exercise where the most important part will be the design of the solution based on a statement.

Motivation

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

- File descriptors and operations like dup, dup2, execv, execvp, execl.
- Signals.
- Threads and/or forks.
- Pipes.
- Shared Memory.
- Sockets.
- Semaphores and/or mutex.
- Messages Queues.

Previous documentation

To complete this session, it is recommended to read of 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.

Celebration in Starfleet

The holiday spirit has arrived in Starfleet and all the crew members of the ships are excited to celebrate Christmas in space. As a leading starship programmer, you have been assigned the mission of creating a galactic program that meets the following guidelines:

- The program must receive an integer as a parameter.
- If you forget to enter the parameter when running the program, it will be displayed on the screen: "Christmas spirit cannot be calculated without precise coordinates."
- If the parameter is provided, the program should display a message of holiday cheer along with a special ornament in ASCII art depicting the iconic USS Enterprise starship surrounded by holiday lights.
- If the parameter is 1701 (the launch year of the USS Enterprise), in addition to the message of joy and ASCII art, you must include your wishes for the crew this season.
- If the parameter is 2024, the program should mention your favorite Star Trek Christmas episode, in addition to the holiday message and ASCII art.

Considerations

• The use of any makefile is not allowed.
• In this session YOU ARE NOT ALLOWED TO ASK QUESTIONS
• The use of #defines is not allowed
• The use of any type of variables is not allowed.
• The use of loops (for, while, do while) is not allowed.
• The use of forks or threads is not allowed.
• The use of shared memory or semaphores is not allowed.
• The use of pipes, sockets or message queues is not allowed.
• The use of signals is not allowed.
• The use of text files is not permitted.
• The use of read or write is not allowed
• The use of file descriptors is not allowed
• The use of scanf, FILE*, getchar is not allowed.
• The group names must be commented above.
• It must be compiled using the –Wall and –Wextra flags.
• Any practice that contains warnings or errors when compiling will have a wonderful double duck.
• Any practice that does not follow the restrictions will have a double duck.