Operating Systems – 2025/2026

Informatics Engineering - ESTiG / IPB

Practical Assignment 2

1 Objective

The objective of Practical Assignment No. 2 is the implementation of a *chat* application, in which communication (message passing) between the various components shall be based on *named pipes* (see Chapter 44 of *The Linux Programming Interface*, by Michael Kerrisk, No Starch Press, 2010, as well as Section 4.9 of the Portuguese textbook). The application will consist of a small set of programs that, together, provide the basic functionalities of a *chat* room. Each running instance of the application will support only a single *chat* room. To operate multiple rooms simultaneously, multiple instances of the application must be launched.

2 Application Architecture

The application will have two components: a *server* and a *client*. Each component will include one or more processes, which will be executed in individual terminals.

2.1 Server

The *server* component is responsible for managing one (and only one) *chat* room (each additional room requires the execution of an additional *server* and its corresponding *clients*). This management encompasses two fundamental functionalities: (S1) registering new *clients*; and (S2) disseminating, to all registered *clients*, the messages sent to the *server* by each *client*.

2.2 Client

The *client* component is responsible for interacting with the *chat* room. This interaction is based on two fundamental functionalities: (C1) the introduction of messages via the keyboard, which are sent to the *server* (to be disseminated to all *clients*); and (C2) the reception and on-screen display of all messages originating from the *server*.

Ideally, the *client* should run in a single terminal, divided into two areas, one dedicated to functionality C1 and the other to functionality C2, similarly to a traditional *chat client*. However, for the purposes of this assignment, it is sufficient to split the

client into two subcomponents: a *client-writer*, which implements functionality C1, and a *client-reader*, which implements functionality C2. Each subcomponent shall be supported by an independent process and executed in its own terminal. Thus, running the *chat* application requires one terminal for the *server* and two terminals per *client*.

3 Development

Considering that the application is targeted at the Linux environment, it may be developed either on the virtual machine provided within the scope of the course or on any other Linux system.

Based on the architecture, the assignment will consist of three C source code files: `server.c`, `client-reader.c`, and `client-writer.c`. A text file, `README.txt`, shall also be included, containing instructions for compiling and running the application, as well as relevant notes on the work developed. All files must clearly indicate, in their respective headers, the authors of the work (name and student number).

4 Evaluation

Evaluation will be based on the defense and demonstration of the assignment, as well as the analysis of its source code.

5 Groups and Deadlines

The assignment must be carried out in groups of two students (exceptions are only allowed for working students or other authorized cases).

A compressed file (`assignment2.zip`) containing the four produced files (`server.c`, `client-reader.c`, `client-writer.c`, and `README.txt`) must be submitted for evaluation. The assignment should be submitted through `virtual.ipb.pt`, in the Operating Systems Assignments area, by 24:00 on 11/01/2026. If this functionality is unavailable, or if submission fails, the compressed file may be sent via email to `arnaldo@ipb.pt` until 00:30 on 12/01/2026.

The defense (demonstration and oral evaluation) of the assignment will take place on a date to be announced. Individual grades within each group may differ depending on each student's performance during the defense.