Instituto Superior de Engenharia do Porto

Master in Critical Computing Systems Engineering

Real-Time Operating Systems Programming (RTOSP)

RTOSP: Assignment: M2

Linux Kernel Development

Paulo Baltarejo Sousa and Cláudio Maia
{pbs,crr}@isep.ipp.pt

2024/25

1 Introduction

The purpose of this assignment is to give students the opportunity to work with Linux kernel at a deeper level and, consequently, develop skills in such area.

The assignment is to be made in teams of two students (exceptions must be approved), and delivered until week 11. Project presentation/demo will be at week 11.

Each team must choose a topic for this assignment and it must be communicated to the professor by email (`pbs@isep.ipp.pt`) until 14/04/2024. Such communication must include a brief description of the topic. If the chosen topic is accepted, it will appear Section 7. If not accepted, then the students must choose another topic or reformulate the previous one.

A portion of classes (T and PL) is reserved to provide support to this assignment.

2 Topic

Given the purpose of this course, real-time operating systems programming, the topics for this assignment should be related to the real-time area and, more specifically, related to task scheduling algorithms and resource-shared mechanisms for real-time systems. However, other topics could be accepted.

Here are some examples for single core systems:

- Rate Monotonic/Deadline Monotonic
  - C.L.Liu,J.Layland."Scheduling algorithms for multiprogramming in a hard–real--time environment",Journal of the ACM(20),p.40–61,1973.
- Least Slack Time First
  - Joseph Y.--T.Leung."A new algorithm for scheduling periodic, real--time tasks", Algorithmica(4), p.209--219,1989
- Protocols for shared resources (Priority Inheritance, Priority Ceiling, Stack Resource Policy)
  - Lui Sha, Rajkumar,R., Lehoczky,J.P., "Priority inheritance protocols:an approach to real--time synchronization," IEEE Transactions on Computers(39), no.9, pp.1175--1185,1990.

  - T.P.Baker."Stack-based scheduling of real–time processes". Real--‑time Systems:The International Journal of Time--Critical Computing(3),1991.

- Constant Bandwidth Server
  - L.Abeni,G.Buttazzo."Integrating multimedia applications in hard real--time systems", Proceedings of the 19th IEEE Real--Time Systems Symposium, p.4 ,1998.

- Any extension to CBS (e.g. GRUB, CASH , CSS , ...)
  - G.Lipari, S.Baruah."Greedy reclamation of unused bandwidth in constant--bandwidth servers", Proceedings of the 12th EuroMicro Conference on Real--Time Systems, pp.193–200,2000.
  - M.Caccamo,G.Buttazzo,L.Sha."Capacity sharing for over run control", Proceedings of 21th IEEE Real Time System Symposium, pp.295–304,2000.
  - Luís Nogueira, Luís Miguel Pinho."A Capacity Sharing and Stealing Strategy for Open Real--time Systems". Journal of Systems Architecture (56),Issues 4--6, pp. 163--179,2010

3 Contents and Delivery

Each team has to delivery three components: Technical Report, Developed code, and Presentation and Discussion.

3.1 Technical Report (TR)

The assignment should be delivered in the form of a report conforming to the Survey Template as available in Moodle (**SurveyFormat.doc** or **SurveyFormatTex.zip**). The report can be in English or in Portuguese, about 10 to 30 pages long. The document has to be submitted **in PDF format** in the moodle and the **document name** has to be in the form **M2_TR_stnr_stnr**.pdf, where **stnr** are the student's numbers.

3.2 Developed Code (DC)

Each team should deliver a Linux kernel patch file containing the add on developed as well as the user-space application for testing such add on. This must be submitted in a zip file in the moodle and **document name** has to be in the form **M2_DC_stnr_stnr**.zip, where **stnr** are the student's numbers.

3.3 Presentation and Discussion (PD)

Each team has to present and discuss the work performed. The discussion can be individually for each team member.

4 Assessment

The M2 assignment will be graded as follows:

$$M2 = 0.20 * TR + (0.80 * DC * PD)$$

, where TR is the Technical Report, DC is the Developed Code and PD is the presentation and discussion.

- TR is graded [0.0,20.0];
- DC is graded [0.0,20.0];
- PD is graded [0.0,1.0];

In the case of a team member does not attend the PD, he will be graded with zero to all components.

5 Planning

5.1 Week mapping

We consider the first week day is on **Monday** and the last week day is on **Sunday**.

Week	Dates	Week	Dates
1	24/02 – 02/03/2025	9	28/04 – 04/05/2025
2	03/03 – 09/03/2025		**Queima**
3	10/03 – 16/03/2025	10	12/05 – 18/05/2025
4	17/03 – 23/03/2025	11	19/05 – 25/05/2025
5	24/03 – 30/03/2025	12	26/05 – 01/06/2025
6	31/03 – 06/04/2025		
7	07/04 – 13/04/2025		
8	14/04 – 20/04/2025		
	Easter		

- Easter holiday: 17/04 – 27/04/2025
- Queima: 04/05 – 11/05/2025

6 Schedule

Week nr	Milestone
1	
2	
3	
4	
5	
6	
7	
8	M2: Team Composition and topic M2: Start
Easter	
9	
Queima	
10	
11	M2: End, Presentation and Discussion

7 Themes/Topics and Teams

All approved themes/topics and teams are in the following list:

Team Number	Team elements	Topic
1	• Luís Ribeiro, 1211555 • Ricardo Freitas, 1210828	GRUB "Greedy reclamation of unused bandwidth in constant-bandwidth servers"
2	• André Gonçalves, 1210804 • Jorge Moreira, 1201458	Least Slack Time First.
3	• Francisco Santos Seabra Mendes, 1232041 • Cristiano Manuel Garcês Coelho, 1182092	Rate Monotonic
4	• Anaísa Carvalho, 1150389 • Luís Carvalho, 1141180	Constant Bandwidth Server
5	• Cláudio Coelho, 1211435 • José Trigo, 1211023	Rate Monotonic / Deadline Monotonic.
6	• Clarisse Sousa, 1211434 • Filipe Duarte, 1210959	CBS - CASH.
7	• Ângelo Oliveira, 1221818 • João Rodrigues, 1211016	Protocol for shared resources (SRP - stack resource policy)
8	• Nuno Castro, 1240160 • Reinaldo Reis, 1201560	Protocols for Shared Resources - Priority Inheritance.
9	• Rúben Seabra, 1181865 • Tiago Costa, 1201329	Constant Bandwidth Server.
10	• Shijo George, 1240374	Constant Bandwidth Server using Rust and BPF in Linux
11	• Filipe Ferreira, 1130425 • Diogo Pereira, 1201599	Protocols for shared resources (Priority Inheritance, Priority Ceiling)