

# Bare Metal Operating Systems

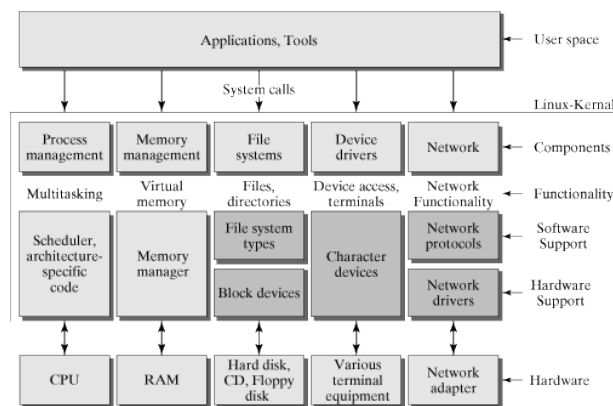
More metal, less threats, higher speed

*Proposta de Projeto*

Orientador: Paul Crocker

## 1 Objetivos

Tipicamente os sistemas operativos *full-fledged* oferecem uma interface muito expressiva à arquitetura máquina sobre a qual assenta permitindo assim uma abstração expressiva que tira o melhor proveito do *hardware* disponível e uma arquitetura de serviços rica aos seus utilizadores (e aplicações).



(arquitetura clássica de um S.O. Linux)

Um sistema de software que se pretende seguro ou eficiente tem de cuidar e assegurar estes requisitos durante a sua conceção e ao mesmo nível do código que o define. Mas estes requisitos dependem necessariamente também do contexto onde vai ser executado.

Se um sistema de ficheiro ou o gestor de memória está comprometido de nada serve ter garantias ao nível da aplicação, esta está também comprometida !

Do ponto de vista da engenharia de sistemas operativos, uma solução passa por propor *sistemas operativos à la carte*, onde o aplicativo por instalar trata ele próprio da interface com o *hardware* mediante as facilidades que o sistema operativo lhe fornece e assim minorar o espaço de exposição a ameaças. Outra vantagem passa pela modularidade da verificação de segurança e de correção funcional.

Estes sistemas operativos são designados de *Bare Metal Operating System*. O *Mirage OS*<sup>1</sup> é um dos sistemas referência deste tipo e está preparado para

<sup>1</sup><https://mirage.io/docs/>

disponibilizar um SO completo aos aplicativos na forma de uma API. Só as componentes essenciais são integradas no aplicativo o que lhe permite por exemplo suportar aplicações complexas em dispositivos limitados computacionalmente.

O Mirage OS é uma das soluções utilizadas pela Docker, Xen Hypervisor etc. Tem recebido contribuições da Citrix, Google, ARM etc.

O objetivo deste projeto é explorar a construção de aplicativos que fazem uso de serviços de OS bare metal (aqui, o Mirage OS) em ambientes computacionais (e comunicantes) limitados (como os sistemas embutidos).

## 2 Tarefas a Realizar e Cronograma

**T1** Estudo dos conceitos envolvidos. 1 mês

**T2** Estudo das tecnologias por utilizar. 1 mês

**T3** Desenho e Implementação da solução. 2 mês

**T4** Validação, teste e análises dos resultados. 0,5 mês

**T5** Escrita do relatório de projeto. 0,5 mês

## 3 Requisitos Técnicos

Gosto em programar e em resolver com rigor problemas de natureza informática. Gosto em desenvolver capacidade em programação, algoritmos e estruturas de dados. Vontade de aprender novas conceitos e novas tecnologias.

## 4 Requisitos Académicos

UE. de Matemáticas, Programação, Algoritmos e Estruturas de Dados, Engenharia de Software, Processamento de Linguagens, Sistemas Operativos.

## 5 Resultados esperados

- 1 protótipo.
- 1 relatório de projeto.

## 6 Contactos

Paul Andrew Crocker (crocker@di.ubi.pt)