

# Projet de système d'exploitation — Rock and Roll —

Benjamin Bonneau & Maxime Flin

13 mai 2020

## Sommaire :

1. Introduction
2. Boot
3. Mémoire
4. Appels systèmes
5. Processus
6. Système de fichier
7. Libc
8. Conclusion et améliorations

# Introduction

`ecos` est un système d'exploitation conçu pour une architecture Intel x86 64 bits. Il propose une interface très proche des normes POSIX et fourni

- ▶ une gestion de processus concurrent
- ▶ une gestion de plusieurs systèmes de fichiers
- ▶ une implémentation de la librairie standard C
- ▶ un shell avec des programmes utilitaires courants (`cat`, `ls`, etc. . .)

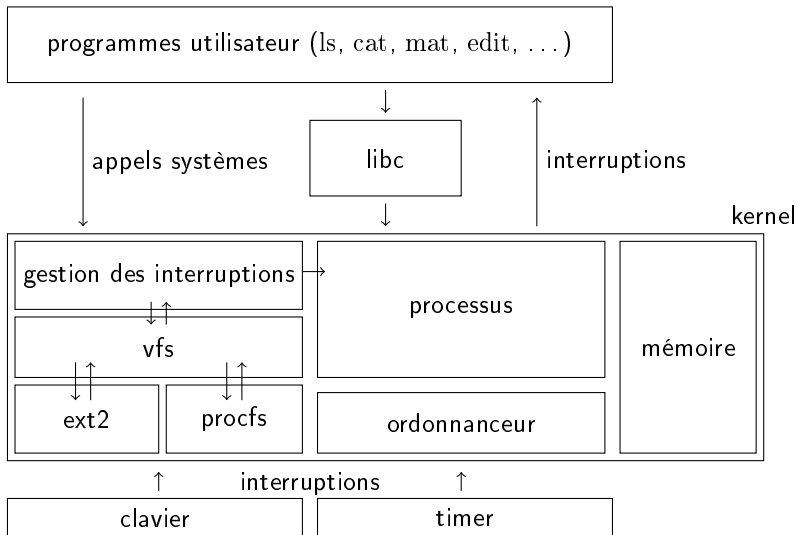


Figure – une vue globale du système

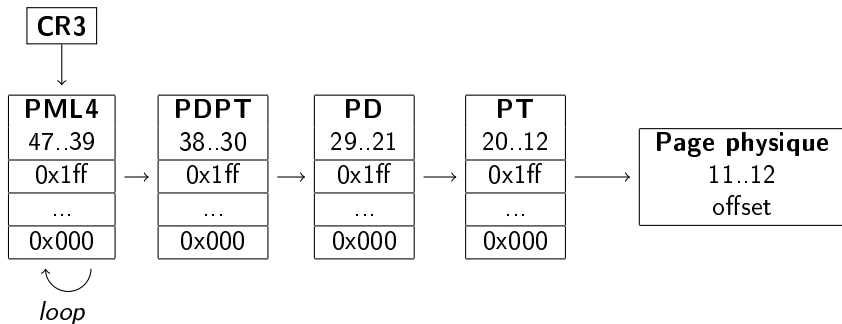
# Boot

src/boot

- ▶ GRUB
  - ▶ kernel (ELF)
  - ▶ carte de la mémoire
  - ▶ affichage
- ▶ *protected mode 32 bits vers long mode 64 bits*
- ▶ Control Registers
- ▶ Paging
- ▶ Global Descriptor Table
- ▶ Chargement du kernel

# Mémoire

Paging 4 niveaux : 48 bits d'adresse



# Mémoire

## Allocation des pages physiques disponibles

- ▶ découpage de la mémoire en blocs de 2Mo
- ▶ au niveau d'un bloc : arbre d'arité 8 (3 niveaux)
- ▶ pour gérer l'ensemble des blocs : 2 arbres d'arité 64. Blocs entièrement libres et blocs partiellement libres.

# Mémoire

- ▶ chaque processus dispose de son PML4
- ▶ partie basse : userspace
- ▶ partie haute : kernel
- ▶ entrées spéciales dans la partie userspace :
  - ▶ Pages partagées (libc)
  - ▶ Pages allouées lors de l'accès (.bss, pile)
  - ▶ Pages copiées lors de l'accès (fork)

actions effectuées lors de l'accès détecté par un page fault



# Execve

- ▶ effectué par un processus auxiliaire avec privilèges de niveau 1
- ▶ gestion des fichiers spécifiant un interpréteur (!)
- ▶ chargement de fichiers ELF
- ▶ création d'un nouveau paging
- ▶ allocation et chargement de sections depuis le fichier : .text, .data, .rodata
- ▶ marquage des pages à allouer lors de l'accès : .bss (initialisé à 0), pile
- ▶ ajout de la libc
- ▶ copie des arguments et paramètres d'environnement