

# PandOS+

Fase 2

**Cheikh Ibrahim · Zaid**

Matricola: 0000974909

**Lee · Qun Hao Henry**

Matricola: 0000990259

**Xia · Tian Cheng**

Matricola: 0000975129

**Paris · Manuel**

Matricola: 0000997526

Anno accademico

2021 — 2022

Corso di Sistemi Operativi

Alma Mater Studiorum · Università di Bologna

# 1 Introduzione

La seconda fase del progetto PandOS+ consiste nell'implementazione del livello 3 dell'architettura astratta di un sistema operativo proposta da Dijkstra.

In particolare è necessario implementare le funzionalità del kernel per gestire la schedulazione dei processi e la gestione delle eccezioni.

## 1.1 Organizzazione dei file

I file coinvolti per l'implementazione delle specifiche sono i seguenti:

<code>pcb</code>	È stata aggiunta la gestione dei <code>pid</code> dei processi.
<code>asl</code>	Sono state aggiunte le operazioni P/V sui semafori.
<code>initial</code>	Gestisce la fase di inizializzazione del sistema e fornisce alcune funzioni ausiliare di utilità generale.
<code>scheduler</code>	Implementa lo scheduler.
<code>exceptions</code>	Gestisce lo "smistamento" le eccezioni ai relativi gestori. Implementa il gestore delle system call.
<code>interrupts</code>	Implementa il gestore delle interrupt.
<code>utilities</code>	Contiene l'implementazione di <code>memcpy</code> .

## 2 Gestione dei pid

I pid vengono generati in ordine crescente a partire da 1.

La mappatura dei pid validi in un dato istante avviene tramite una lista bidirezionale implementata attraverso un nuovo campo aggiunto della struttura `pcb_t`. Tale lista è ordinata in senso crescente per migliorare le prestazioni di ricerca e permette l'inserimento in tempo costante per il primo ciclo di pid (dopo un wraparound non è garantita questa proprietà).

### 2.1 Possibili miglioramenti

Una possibile alternativa a tale implementazione è basata sugli alberi di ricerca che permettono di ottenere maggiori prestazioni nella fase di ricerca.

La scelta di utilizzare una lista ordinata è stata fatta considerando un trade-off tra semplicità di implementazione e prestazioni che rimangono valide in quanto il numero massimo di processi attivi ha un limite superiore molto contenuto.

- 3 Gestione dei semafori dei device
- 4 Gestione contabilizzazione del tempo di CPU
- 5 Inizializzazione del sistema
- 6 Scheduling dei processi
- 7 Gestione delle eccezioni