

Ping-Pong OS - v2

O **Sistema Operacional Ping-Pong**, ou PPOS (*Ping-Pong Operating System*), é um projeto acadêmico que visa construir um sistema operacional didático (*toy operating system*) em linguagem C.

Esta página descreve a versão 2 do PPOS. A versão 1 pode ser encontrada [nesta página](#).

Ao contrário de projetos similares, o PPOS executa inteiramente dentro de um processo do Linux. A maior vantagem em emular um SO dentro de um processo em modo usuário é a possibilidade de usar ferramentas padrão de depuração, como o Valgrind e o GDB no modo default, o que reduz a curva de aprendizado do aluno.

A versão atual do PPOS suporta múltiplas tarefas com preempção por tempo, escalonador baseado em prioridades, semáforos, filas de mensagens, alocador de memória *heap*, acesso a disco e sistema de arquivos. Devido à característica de executar em um processo Linux, o PPOS **não suporta** memória virtual, separação usuário/kernel e chamadas de sistema por *traps*.

O PPOS segue a estrutura típica de um RTOS (*Real-Time Operating System*), um sistema operacional com espaço de memória único, no qual o núcleo e as aplicações são compilados juntos, formando uma única imagem binária.



Eu estou preparando uma lista dos **cursos que usam este projeto** . Se você (ou seu professor) usa este projeto em suas aulas, peço a gentileza de me enviar um [e-mail](#) informando sua instituição e seu curso.

Obrigado!

Projetos

O desenvolvimento do PPOS está organizado em projetos incrementais relativamente simples, que visam implementar cada uma das funcionalidades do sistema, em sequência. Cada projeto geralmente depende das funcionalidades implementadas nos projetos anteriores.

Os projetos atualmente definidos são:

- [Fila genérica](#)
- [Mapa genérico](#)
- [Trocas de contexto](#)
- P1: [Tarefas cooperativas](#)
- P2: [Despachante de tarefas](#)
- P3: [Escalonador por prioridades](#)
- P4: [Preempção por tempo](#)
- P5: [Relógio e contabilização](#)
- P6: [Tarefas que esperam](#)
- P7: [Tarefas que dormem](#)
- P8: [Spinlocks e semáforos](#)
- P9: [Sistema produtor-consumidor](#)
- P10: [Filas de mensagens](#)
- P11: [Alocador de memória](#)
- P12: [Acesso ao disco](#)
- P13: [Escalonador de disco](#)
- P14: [Cache de blocos](#)

Projetos em estruturação:

- [Gestão de inodes](#)
- [Diretórios](#)
- [Nomes de arquivos](#)
- [Acesso à rede](#)
- [Acesso ao terminal](#)
- [Execução nativa em Risc-V](#)

Código-fonte

O

código-fonte do PPOS

versão 2 tem a seguinte estrutura:

```
ppos/
  hardware/      : emulação do hardware
    disk*        : disco rígido
    cpu.*         : operações da CPU/chipset
    serial.*      : porta serial (console)
    makefile
  kernel/        : núcleo do PPOS
    ppos.*        : núcleo (arquivo principal)
    macros.h      : macros de uso geral
    ctx.*         : trocas de contexto (assembly e C)
    tcb.h         : task control block (descriptor de tarefa)
    task.*        : gestão básica de tarefas
    dispatcher.*  : despachante de tarefas
    scheduler.*   : escalonador de tarefas
    time.*        : gestão do tempo
    semaphore.*   : spinlocks e semáforos
    mqueue.*      : filas de mensagens
    memory.*      : gestão de memória heap
    block.*       : acesso aos blocos do disco
    syscall.h     : funções acessíveis às aplicações
    makefile
  lib/
    map.*         : biblioteca de mapas genéricos
    queue.*       : biblioteca de filas genéricas
    pplibc.*      : mini-biblioteca C padrão
    makefile
  test/
    pingpong-*.c  : programas de teste dos projetos
    pingpong-*.txt : saídas esperadas dos testes
    makefile
  makefile
```



Alguns arquivos (sobretudo os *headers*, os arquivos no diretório `hardware`, a mini-biblioteca C e os `makefiles`) estão prontos e **não devem ser alterados** (essa restrição está claramente indicada no início de cada arquivo). Esses arquivos serão sobrescritos pelos originais durante a avaliação de cada projeto.

Entregas

Alguns esclarecimentos sobre a entrega dos projetos:

Quem deve entregar: apenas **um aluno** por grupo.

Como identificar os autores: incluir **comentários no início** de cada arquivo criado ou alterado pelos alunos, seguindo este modelo:

```
// GRR12345678 Homer Simpson  
// GRR12345679 Marge Simpson
```

O que entregar: execute `make tgz` e envie o arquivo TGZ produzido.

From:

<https://wiki.inf.ufpr.br/maziero/> - **Prof. Carlos Maziero**

Permanent link:

<https://wiki.inf.ufpr.br/maziero/doku.php?id=ppos-v2:start&rev=1787830331>

Last update: **2026/08/27 08:32**

