

## Procedimento

Abaixo, segue a lista de projetos para o semestre 25.1. Caso tenha interesse em um dos projetos, favor contatar o professor. Caso tenham interesse em fazer um projeto que não esteja listado aqui, favor contactar o professor. Alguns dos projetos podem ser muito complexos para serem feitos em dois meses por um único aluno; nestes casos, discutir com o professor sobre o escopo (e tamanho da equipe). Os alunos devem escrever uma promessa do que será feito. Essa promessa será usada tanto para avaliação do trabalho como guia de acompanhamento.

## General

- Quantificar algum trade-off visto em sala. P.ex, quando discutimos os modelos  $M-\{0, 1, 2, 3\}$  de gerenciamento de memória, discutimos que alguns deles resolvem problemas de desempenho dos anteriores. Quanto mais eficiente os melhores modelos podem ser?
- Em muitas discussões de design, usamos estruturas de dados clássicas. Há estruturas que podem funcionar melhor do que as vistas em sala de aula. Por exemplo, *roaring bitmaps* podem ser uma alternativa para *bitmaps* clássicos em nosso contexto?

## Build

- Construir uma [máquina simples \(de 8 bits\) do zero](#)

## system software

- Implementar uma política/mecanismo relacionado com page cache usando [cachebpf](#)

## scheduling

- página web que permita simulações sobre trade-offs de algoritmos de escalonamento de processos

## memory management

- um relatório (submetido para o arxiv.org) sobre como funciona memory allocation no linux. o trabalho deve se basear no código fonte atual do sistema
- um relatório (submetido para o arxiv.org) sobre como funciona page eviction no linux. o trabalho deve se basear no código fonte atual do sistema

- Scripts para observar (ou coletar traces) fragmentação interna no linux (ou outras métricas relevantes de mem mgmt)
- Scripts para observar (ou coletar traces) do uso de endereços virtuais por processo. Quão grandes são os processos?
- Avaliação de impacto no desempenho do metadado dirty-bit - Francisco Filho

## I/O

1. página web que permita simulações sobre trade-offs de algoritmos de escalonamento de I/O
2. Scripts para observar (ou coletar traces) escalonamento de I/O requests no linux

## file systems

### [zepa machine](#)

1. extensão da spec da ISA (e implementação)
2. software básico
  - a. extensão do assembly/assembler
    - i. adição de labels
    - ii. adição de saltos condicionais
  - b. implementação de um assembler escrito na própria linguagem definida para a máquina (hoje, temos um cross-assemble)
  - c. bootloader
3. spec e impl de syscalls - ?
4. batch OS
5. small multiprogram interactive OS
  - a. sched - Livia
  - b. mem mgmt - ?
6. S0 extra
  - a. I/O & device drivers - ?
  - b. file system- ?
7. dev & userspace
  - a. high-level language & compiler
  - b. define elf
  - c. stdlib
  - d. shell & utilities
8. tooling
  - a. OS execution debugger (web ou cli)
  - b. zepa on qemu - Julio