

## MAC0422 - Sistemas Operacionais

Matheus Santos - 10297672

Vitor Barbosa S rio - 7627627

Esse trabalho simula uma shell capaz de interpretar 5 comandos distintos: `protegepracaramba`, `liberageral`, `rodeveja`, `rode` e `quit`, que funcionam das seguintes formas:

- `protegepracaramba <arquivo>`: coloca prote  o 000 em `arquivo`, utilizando as chamadas `fork`, que cria um processo filho, `execve`, para executar outro programa no lugar do processo filho, e `chmod`, que   a chamada executada por `execve` e que altera as permiss es de `arquivo`.
- `liberageral <arquivo>`: coloca prote  o 777 em `arquivo`, utilizando as chamadas `fork`, que cria um processo filho, `execve`, para executar outro programa no lugar do processo filho, e `chmod`, que   a chamada executada por `execve` e que altera as permiss es de `arquivo`.
- `rodeveja <programa>`: executa `programa` e imprime o valor de retorno do mesmo ao final, utilizando as chamadas `fork`, que cria um processo filho, `execve`, para executar `programa` no lugar do processo filho, e `wait`, para que o processo pai espere o processo filho terminar. Dessa forma, como o processo filho est  executando em primeiro plano (foreground) a shell n o consegue executar outro comando enquanto o processo filho n o terminar (ctrl+c encerra o processo filho).
- `rode <programa>`: executa `programa`, sem imprimir o valor de retorno e tamb m monopoliza o teclado e o mouse, ou seja, o processo filho estar  executando em segundo plano (background) e permitindo que a shell execute outros comandos. Isso   feito utilizando as chamadas `fork`, que cria um processo filho, e `execve`, para executar `programa` no lugar do processo filho criado (ctrl+c n o encerra o processo filho).
- `quit`: encerra a shell.