

DCC605 - Sistemas Operacionais – 2021/2
Primeiro trabalho prático – Comando time no xv6
Samuel Fantini Braga
2018046637

O programa inicia buscando o timestamp atual e armazenando em uma variável utilizando a função uptime, dessa forma, essa variável irá marcar o início do programa. Utilizando o comando fork que já está no xv6 eu segrego os processos. Portanto, o programa irá rodar separadamente do que foi passado como argumento.

É feito, algumas análises sobre o status do processo solicitado, onde temos algumas condicionantes:

Status menor que 0: O processo solicitado é inválido e não está presente na arquitetura do xv6 no momento.

Status maior que 0: O processo está em execução.

status igual a 0: O processo ainda não foi executado.

Portanto, realizando essas análises, é possível executar o processo solicitado, passando os parâmetros especificados com a função exec que já está no xv6.

Ademais, no final do processo eu busco o timestamp atual e armazeno em uma variável, dessa forma, a variável marcará o final do processo. Com duas variáveis, uma marcando o início e outra o fim do processo, basta calcular a diferença e transformar em segundos.

Um ponto a ser considerado foi a dificuldade em desenvolver um programa de teste para a função time. Dessa forma, é compreensível e esperado algum desconto de pontos no TP.

```
$ time zombie
zombzoimbei!e
demorou 0.06 segundos
$ █
```

```
bigfile test ok
subdir test
subdir ok
linktest
linktest ok
unlinkread test
unlinkread ok
dir vs file
dir vs file OK
empty file name
empty file name OK
fork test
fork test OK
bigdir test
bigdir ok
uio test
pid 594 usertests: trap 13 err 0 on cpu 1 eip 0x3593 addr 0x801dc130--kill proc
uio test done
exec test
ALL TESTS PASSED
usertests demorou 51.56 segundos
```