

## Roteiro 1 - Processos

- Orientações:
    - Ao elaborar o relatório desse roteiro, **FAÇA PRINTS DA TELA SEMPRE QUE POSSÍVEL.**
    - **Mande apenas um único arquivo compactado (zip, rar ou tar.gz)**
- 
1. Faça *download* do [arquivo](#) para a prática
  2. Abra um terminal
  3. Descompacte o arquivo. No terminal digite: "tar xvfz process\_exercise.tar.gz"
  4. Encontre o arquivo processdemo.c dentro da pasta descompactada. O programa usa a chamada fork() para criar um processo filho. Os processos pai e filho são executados separadamente e cada um chama a função adjustX() com parâmetros diferentes em cada processo
  5. Vamos examinar o código-fonte e tentar determinar o que ele faz. (Você pode fazer isso usando o programa Gedit.)
  6. Faça um exercício de tentar prever o que o código vai fazer
  7. Compile o código-fonte. No terminal digite:  
  
`gcc processdemo.c -o processdemo`
  8. Rode o programa. No terminal digite: `./processdemo`
  9. **Descreva a saída e explique por que ela é dessa forma.**
  10. Enquanto o programa roda, execute (em outro terminal) o comando `ps xl`
  11. **Qual o processo pai e qual o processo filho? (Dica, verifique a coluna PID e PPID. Se não souber o que é PID e PPID, procure no Google). Justifique.**

12. Use o comando "kill -9 PID" para matar o processo filho. O que aconteceu?
13. Use o comando "kill -9 PID" para matar o processo pai. O que aconteceu?
14. Rode o programa novamente. Identifique e mate o processo pai primeiro em seguida o filho. O que aconteceu?
15. Faz diferença matar o pai ou o filho antes?
16. Compile o programa threaddemo.c com o comando "gcc threaddemo.c -lpthread -o threaddemo"
17. Esse programa usa a biblioteca *POSIX threads library* e é muito similar a processdemo.c, mas usa *threads* em vez de processos
18. Rode o programa. O que ele faz? Qual a diferença dele para o programa processdemo.c?
19. Qual a diferença de velocidade de saída (medido em linhas por segundo) comparado a processdemo? Quem é mais rápido? Você tem uma idéia do porquê?
20. Use o comando ps xl para verificar que há apenas 1 processo threaddemo
21. Investigue o efeito de remover o *loop* infinito no fim do main(). O que acontece? Por que?
22. Modifique o programa threaddemo.c para ele fazer a mesma coisa que processdemo.c
23. Escreva um relatório do experimento respondendo cada **pergunta em negrito** deste roteiro. Seja honesto, você aprenderá mais dessa forma.