

### Laboratório 3 – Sinais

- 1) Execute o programa “ctrl-c.c”. Digite Ctrl-C e Ctrl-\. Analise o resultado. Neste mesmo programa, remova os comandos `signal( )` e repita o teste anterior observando os resultados. **Explique o que ocorreu no relatório.**
- 2) Tente fazer um programa para interceptar o sinal SIGKILL. Você conseguiu? **Explique.**
- 3) Execute e explique o funcionamento de `filhocidio.c`, com as 4 opções:  
a- `for(EVER)`                                `/* filho em loop eterno */`  
b- `sleep(3)`                                    `/* filho dorme 3 segundos */`  
c- `execvp(sleep5)`                           `/* filho executa o programa sleep5 */`  
d- `execvp(sleep15)`                        `/* filho executa o programa sleep15 */`  
**Explique o que ocorreu em cada programa.**
- 4) Faça um programa que leia **2 números reais** e imprima o resultado das 4 operações básicas sobre estes 2 números. Verifique o que acontece se o 2º. número da entrada for 0 (zero). Capture o sinal de erro de floating point (SIGFPE) e repita a experiência anterior. Faça o mesmo agora **lendo e realizando as operações com inteiros**. **Explique o que ocorreu nas duas situações.**
- 5) Faça um programa que tenha um coordenador e dois filhos. Os filhos executam (`execvp`) um programa que tenha um loop eterno. O pai coordena a execução dos filhos realizando a preempção dos processos, executando um deles por 1 segundo, interrompendo a sua execução e executando o outro por 1 segundo, interrompendo a sua execução e assim sucessivamente. O processo pai fica então coordenando a execução dos filhos, é, na verdade, um escalonador. Faça o processo pai executar por 15 segundos e, ao final, ele mata os processos filhos e termina. **Explique como realizou a preempção**, se o programa funcionou a contento e as dificuldades encontradas.