//Eric Goulart da Cunha - 2110878
//João Pedro Biscaia Fernandes - 2110361

Relatório Laboratório 3A Sistemas Operacionais:

Questão 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.

R: O resultado desse programa é que o terminal não encerra o programa quando utilizamos Ctrl-C como normalmente, ele encerra quando digitamos Ctrl-\. Quando removemos o comando signal() o que acontece é que paramos de interceptar o Ctrl-C então ele volta a funcionar, o Ctrl-\ continua funcionando.

Questão 2: Tente fazer um programa para interceptar o sinal SIGKILL. Você conseguiu? Explique.

R:Não conseguimos. Tentamos utilizar um signal(SIGKILL, killHandler) porém é impossível interceptar o SIGKILL.

```c
void killHandler(int sinal);

int main (void) {
  void (*p)(int);// ponteiro para função que recebe int como parâmetro
  pid_t pid = getpid();
  printf("pid é === %d\n",pid);
  p = signal(SIGKILL, killHandler);
  printf("Endereco do manipulador anterior %p\n", p);
  for(EVER);
}

void killHandler(int sinal) {
  printf ("Você tentou dar sigkill (%d)\n", sinal);
}
```

Questão 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 */

R:

a) O programa ficaria em loop eterno já que o filho nunca seria terminado, isso não acontece pois colocamos um delay máximo de 10 segundos e então o programa termina.
b) Após 3 segundo o processo filho encerra, enviando o sinal para a função signal, chamando a função childHandler que encerra o programa antes que o processo pai

consiga usar o comando kill no processo que encerraria o programa da mesma forma que a questão anterior.

c) Nesse caso o processo filho utiliza o execvp, chamando o comando sleep5, que simplesmente "dorme" por 5 segundos, e ao encerrar, retorna para o processo filho, que também imediatamente acaba, assim terminando em menos de 10 segundos, chamando o Handler para o sinal do filho da mesma forma que na questão anterior. Prompt do terminal:

```
jp@jp-A520M-S2H:~/Documents/PUCRIO/SistemasOperacionais/lab3$ ./ex3 10 ./sleep5
indo dormir...
Acordei!
Child 61618 terminated within 10 seconds com estado 0.
```

d)

Nesse caso, diferente da questão anterior, o sleep é superior ao tempo limite de espera recebido como parâmetro(10 segundos.), logo não teve tempo de acabar antes que o pai encerrasse o processo filho forçadamente, assim como encerrou na questão A.

```
jp@jp-A520M-S2H:~/Documents/PUCRIO/SistemasOperacionais/lab3$ ./ex3 10 ./sleep15
indo dormir...
Program ./sleep15 exceeded limit of 10 seconds!
```