

1. Feladat :

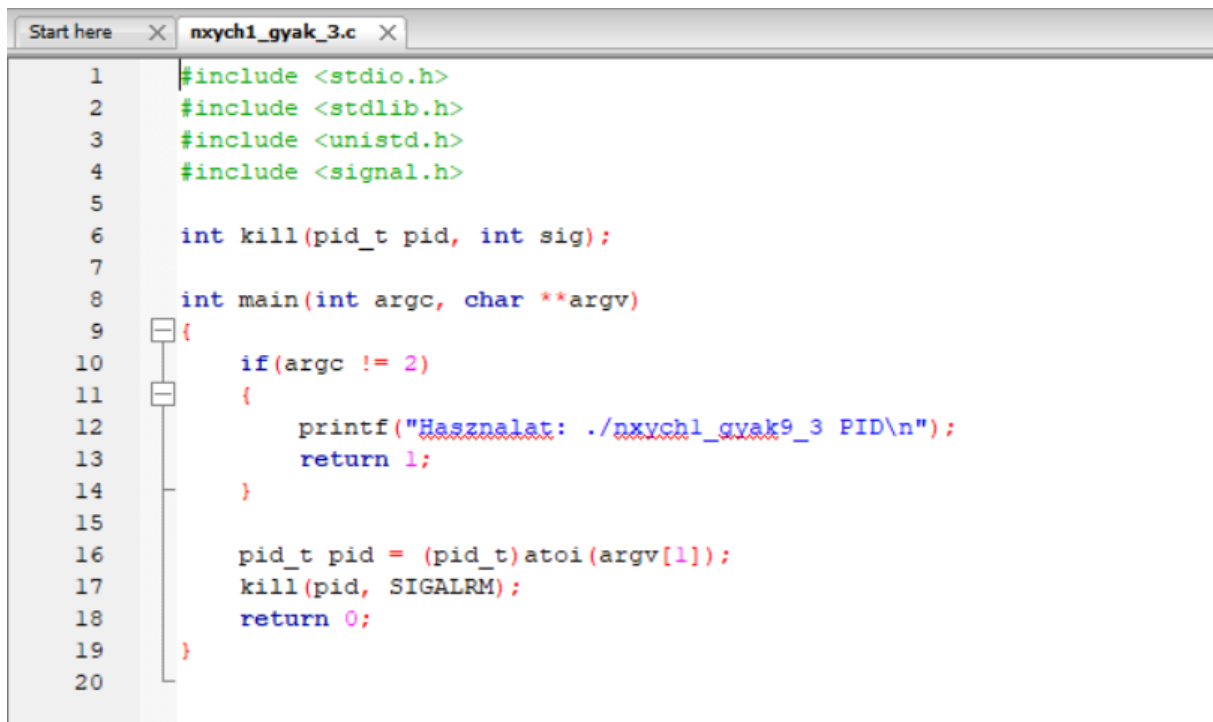
. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetékét, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre. Mentés: neptunkod_unnamed.c

2. Feladat: .

Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetékét (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve:pl. Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre. Mentés: neptunkod_named.c

3. Feladat:

Írjon C nyelvű programot, amelyik kill() seg.-vel SIGALRM-et küld egy argumentumként megadott PID-u processznek, egy másik futó program a SIGALRM-hez rendeljen egy fv.-t amely kiírja pl.neptunkodot, továbbá pause() fv.-el blokkolódjon, majd kibillenés után jelezze, hogy kibillent és terminálódjon. Mentés. neptunkod_gyak9_3.c



```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <signal.h>
5
6  int kill(pid_t pid, int sig);
7
8  int main(int argc, char **argv)
9  {
10     if(argc != 2)
11     {
12         printf("Hasznalat: ./nxychl_gyak9_3 PID\n");
13         return 1;
14     }
15
16     pid_t pid = (pid_t)atoi(argv[1]);
17     kill(pid, SIGALRM);
18     return 0;
19 }
20
```

```
Start here x nxych1_gyak_3_2.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <signal.h>
5
6  void AlarmHandler(int sig);
7
8  int main(void)
9  {
10     if (signal(SIGALRM, AlarmHandler) == SIG_ERR)
11     {
12         printf("Nem sikerult handlert allitani a(z) \"SIGALRM\" jelre!\n");
13         return 1;
14     }
15
16     pause();
17     return 0;
18 }
19
20 void AlarmHandler(int sig)
21 {
22     printf("NXYCH1\n Mostantol nem blokkol !\n");
23     exit(1);
24 }
25
```

4. Feladat:

Írjon C nyelvű programot, amelyik a SIGTERM-hez hozzárendel egy fv-t., amelyik kiírja az int paraméter értéket, majd végtelen ciklusban fusson, 3 sec-ig állandóan blokkolódva elindítás után egy másik shell-ben kill paranccsal (SIGTERM) próbálja terminálni, majd SIGKILL-el.

Mentés. neptunkod_gyak9_4.c

```
Start here x nxych1_gyak_4.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #include <unistd.h>
5  #include <signal.h>
6
7  void TerminationHandler(int sig);
8
9  int main(void)
10 {
11     if (signal(SIGTERM, TerminationHandler) == SIG_ERR)
12     {
13         printf("Nem sikerult a handlert allitani a \"SIGTERM\" jelre!\n");
14         return 0;
15     }
16
17     while(1)
18     {
19         printf("Varakozom...\n");
20         sleep(3);
21     }
22
23     return 0;
24 }
25
26 void TerminationHandler(int sig)
27 {
28     signal(sig, SIG_IGN);
29     printf("SIGTERM signal: %d\n", sig);
30 }
31
```