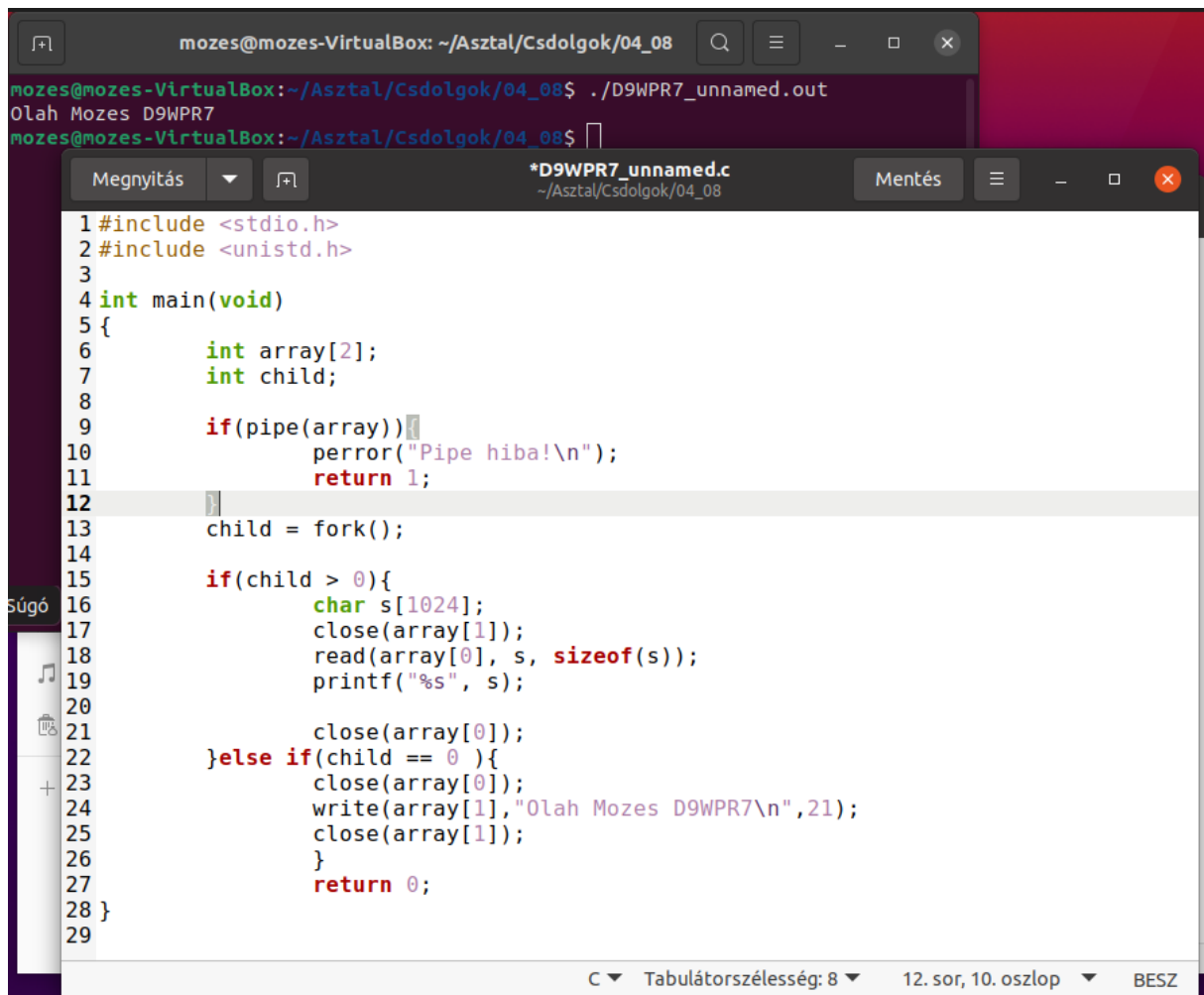


1.



The image shows a virtual machine environment with a terminal window and a code editor. The terminal window at the top shows the execution of a program named `D9WPR7_unnamed.out`, which outputs `Olah Mozes D9WPR7`. Below the terminal is a code editor window titled `*D9WPR7_unnamed.c` showing the source code of the program. The code is a C program that uses `pipe` and `fork` to create a child process that reads from a pipe and prints the output.

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void)
5 {
6     int array[2];
7     int child;
8
9     if(pipe(array)){
10         perror("Pipe hiba!\n");
11         return 1;
12     }
13     child = fork();
14
15     if(child > 0){
16         char s[1024];
17         close(array[1]);
18         read(array[0], s, sizeof(s));
19         printf("%s", s);
20
21         close(array[0]);
22     } else if(child == 0){
23         close(array[0]);
24         write(array[1], "Olah Mozes D9WPR7\n", 21);
25         close(array[1]);
26     }
27     return 0;
28 }
29
```

The code editor window has a sidebar on the left with icons for file management and a search bar. The status bar at the bottom indicates the current file is `C`, the tab width is 8, and the cursor is at line 12, column 10.

2.

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_08$ ./D9WPR7_named.out
Rendszer: Gyerek processz sikeresen elindult!
Rendszer: Gyerek processz sikeresen elindult!
Gyerek: FIFO írásra megnyitva, írás folyamatban
Rendszer: A gyerek processz sikeresen befejezte az írást,
A gyerek processz kilep.

Rendszer: Szülő processz indítása
5 Ubuntu Software asásra megnyitva, olvasás folyamatban
Szülő: Kiolvasott adat: Olah Mozes.

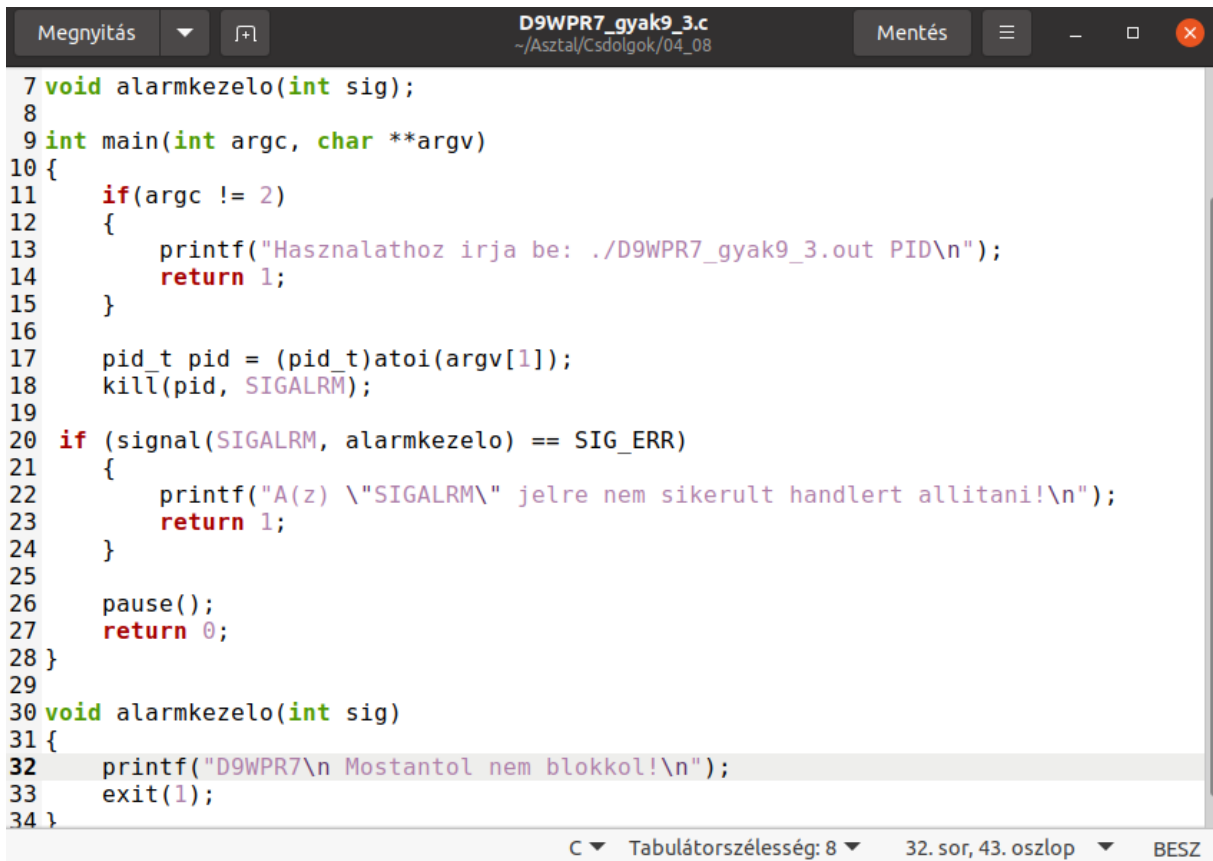
Rendszer: A szülő processz sikeresen befejezte az olvasást.
Rendszer: FIFO sikeresen törölve, kilepes.
```

```
Megnyitás  D9WPR7_named.c  Mentés
~/Asztal/Csdolgozok/04_08

1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/types.h>
4 #include <sys/stat.h>
5 #include <fcntl.h>
6 #include <stdlib.h>
7 #include <string.h>
8
9
10 int main(){
11
12     int fd;
13     pid_t gyerek;
14
15     if((mkfifo("D9WPR7", S_IRUSR | S_IWUSR)) < 0)
16 {
17     printf("Rendszer: FIFO létrehozás sikertelen!\n");
18     exit(1);
19 }
20
21     if((gyerek = fork()) < 0){
22     printf("Rendszer: Gyerek fork processz létrehozás sikertelen!\n");
23     exit(1);
24     }
25 else
26 {
27 printf("Rendszer: Gyerek processz sikeresen elindult!\n");
28 }
29
30 //Gyerek processz
31     if(!gyerek){
32         char *uzenet = "Olah Mozes";
33
34
35         if((fd = open("D9WPR7", O_WRONLY)) < 0){
```

3.

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_08$ cc D9WPR7_gyak9_3.c -o D9WPR7_gy
ak9_3.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_08$ ./D9WPR7_gyak9_3.out
Hasznalathoz irja be: ./D9WPR7_gyak9_3.out PID
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_08$ ./D9WPR7_gyak9_3.out PID
Időzítő óra
mozes@mozes-VirtualBox:~/Asztal/Csdolgok/04_08$
```



```
D9WPR7_gyak9_3.c
~/Asztal/Csdolgok/04_08

7 void alarmkezeselo(int sig);
8
9 int main(int argc, char **argv)
10 {
11     if(argc != 2)
12     {
13         printf("Hasznalathoz irja be: ./D9WPR7_gyak9_3.out PID\n");
14         return 1;
15     }
16     pid_t pid = (pid_t)atoi(argv[1]);
17     kill(pid, SIGALRM);
18
19     if (signal(SIGALRM, alarmkezeselo) == SIG_ERR)
20     {
21         printf("A(z) \"SIGALRM\" jelre nem sikerult handlert allitani!\n");
22         return 1;
23     }
24
25     pause();
26     return 0;
27 }
28
29 void alarmkezeselo(int sig)
30 {
31     printf("D9WPR7\n Mostantol nem blokkol!\n");
32     exit(1);
33 }
34
```

C ▾ Tabulátorszélesség: 8 ▾ 32. sor, 43. oszlop ▾ BESZ

4.

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_08$ ./D9WPR7_gyak9_41.out
Varakozok..
Varakozok..
Varakozok..
Varakozok..
Varakozok..
^Z
[2]+  Megállítva                  ./D9WPR7_gyak9_41.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_08$
```

```
Megnyitás ▼ [F1] D9WPR7_gyak9_4.c ~/Asztal/Csdolgozok/04_08 Mentés ☰ - □ ×

3 #include <unistd.h>
4 #include <signal.h>
5
6
7 void terminal(int sig);
8
9 int main(void)
10 {
11     if (signal(SIGTERM, terminal) == SIG_ERR)
12     {
13         printf("A SIGTERM jelre nem sikerult handlert allitani!\n");
14         return 0;
15     }
16
17     while(-1)
18     {
19         printf("Varakozok..\n");
20         sleep(3);
21     }
22
23     return 0;
24 }
25
26 void terminal(int sig){
27     signal(sig, SIG_IGN);
28     printf("Jel: %d\n", sig);
29 }
30 }
```

C ▼ Tabulátorszélesség: 8 ▼ 20. sor, 18. oszlop ▼ BESZ

```
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_08$ cc D9WPR7_gyak9_4.c -o D9WPR7_gy
ak9_4.out
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_08$ ./D9WPR7_gyak9_4.out
A SIGKILL jelre nem sikerult handlert allitani!
mozes@mozes-VirtualBox:~/Asztal/Csdolgozok/04_08$
```

Megnyitás   D9WPR7_gyak9_4.c
~/Asztal/Csdolgozok/04_08

Mentés    

```
3 #include <unistd.h>
4 #include <signal.h>
5
6
7 void terminal(int sig);
8
9 int main(void)
10 {
11     if (signal(SIGKILL, terminal) == SIG_ERR)
12     {
13         printf("A SIGKILL jelre nem sikerult handlert allitani!\n");
14         return 0;
15     }
16
17     while(-1)
18     {
19         printf("Varakozok..\n");
20         sleep(3);
21     }
22
23     return 0;
24 }
25
26 void terminal(int sig){
27     signal(sig, SIG_IGN);
28     printf("Jel: %d\n", sig);
29 }
30 }
```

C ▾ Tabulátorszélesség: 8 ▾ 20. sor, 18. oszlop ▾ BESZ