

Задачи със системни примитиви

1. Да се напише програма, която прочита от стандартния вход 4-цифрено цяло число и го извежда на стандартния изход.
2. Дадена е програмата:

```
#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>

int main()
{
    int a;
    read(0, &a, sizeof(a));
    write(1, &a, sizeof(a));
    printf("%d\n", a);
    return 0;
}
```

Компилираме я в изпълним файл с името a.out.

Изпълняваме я 4 пъти, подавайки съответно числата 34, 123, 1234, 12345 на стандартния вход:

```
dvd@BlackPearl:~/hw/os/syscalls_tests$ ./a.out
34
34
668723
dvd@BlackPearl:~/hw/os/syscalls_tests$ ./a.out
123
123
171127345
dvd@BlackPearl:~/hw/os/syscalls_tests$ ./a.out
1234
1234875770417
dvd@BlackPearl:~/hw/os/syscalls_tests$
dvd@BlackPearl:~/hw/os/syscalls_tests$ ./a.out
12345
1234875770417
dvd@BlackPearl:~/hw/os/syscalls_tests$ 5
bash: 5: command not found
```

```
dvd@BlackPearl:~/hw/os/syscalls_tests$
```

Обяснете какво се е изписало на стандартния изход.

3. Да се напише програма, която на стандартния си вход ще получи 1 ред, съдържащ име на изпълним файл. Програмата трябва да прочете това име и да ехесите-не указания файл:
 - a) в текущия процес
 - b) в процес-дете на текущия

4. Да се напише програма `mywc`, която има поведение, подобно на програмата `wc`:

```
a) dvd@BlackPearl:~/Desktop$ ./mywc
standard input, ctrl-d to end
1  5  31
dvd@BlackPearl:~/Desktop$ ./mywc mywc.c
221 498 6369mywc.c
dvd@BlackPearl:~/Desktop$
```

```
b) към поведението от а) добавяме:
dvd@BlackPearl:~/Desktop$ ./mywc mywc.c
~/hw/os/test/1.sh ~/try/try.py
221 498 6369mywc.c
16  64  360 /home/dvd/hw/os/test/1.sh
19  60  699 /home/dvd/try/try.py
256 622 7428total
dvd@BlackPearl:~/Desktop$
```

```
c) (*)
към поведението от а) и b) добавяме:
dvd@BlackPearl:~/Desktop$ ./mywc -l mywc.c
221 mywc.c
dvd@BlackPearl:~/Desktop$ ./mywc -c mywc.c
6369mywc.c
dvd@BlackPearl:~/Desktop$ ./mywc -w mywc.c
498 mywc.c
dvd@BlackPearl:~/Desktop$ ./mywc -m mywc.c
mywc: unrecognized option 'm'
dvd@BlackPearl:~/Desktop$
```

```
d) (*)
```

има изцяло същото действие като wc (с възможности за броене на редове, думи, байтове, т.е. без да брои символите във файла)

5. Да се провери кои числа x от 0 до $2^{20}-1$ изпълняват условието $((((x\%179)*12)\%37)\&10)\neq 0$
Сложността по време трябва да бъде константна. Сложността по брой процеси е неограничена.

6. Да се напише програма, която стартира процес, който:
 1. приключва работа, когато размерът на файла kikufigu в текущата директория стане по-голям от 4K (приемаме, че файлът съществува и имаме нужните права за достъп до него)
 2. има за родителски процес init