

## Zadatak za lab1-signali, student: \_\_\_\_\_

Zadani program (desno) je pokrenut u  $t = 0$  sekundi. Navesti kako će se odvijati izvođenje programa kada signali dolaze zadanim redoslijedom, tj. što će program ispisati tijekom rada, npr.:

```
0: main: pocetak
5: obrada 1: pocetak
...
```

Pokazati ponašanje u tri scenarija:

1. u  $t = 5$  SIGUSR1; u  $t=10$  SIGUSR2

```
0: main: pocetak
5: obrada 1: pocetak
10: obrada 2: pocetak
20: obrada 2: kraj
25: obrada 1: kraj
30: main: kraj
```

2. u  $t = 5$  SIGUSR1; u  $t=10$  SIGUSR2; u  $t=15$  SIGUSR2

```
0: main: pocetak
5: obrada 1: pocetak
10: obrada 2: pocetak
20: obrada 2: kraj
20: obrada 2: pocetak
30: obrada 2: kraj
35: obrada 1: kraj
40: main: kraj
```

3. u  $t = 5$  SIGUSR1; u  $t=10$  SIGUSR2; u  $t=15$  SIGUSR1

```
0: main: pocetak
5: obrada 1: pocetak
10: obrada 2: pocetak
20: obrada 2: kraj
25: obrada 1: kraj
25: obrada 1: pocetak
35: obrada 1: kraj
40: main: kraj
```

```
void obrada1(int sig)
{
    printf("obrada 1: pocetak\n");
    "neki posao (proračun)
    koji traje 10 s,
    ne koristi se sleep";
    printf("obrada 1: kraj\n");
}

void obrada2(int sig)
{
    printf("obrada 2: pocetak\n");
    "neki posao (proračun)
    koji traje 10 s,
    ne koristi se sleep";
    printf("obrada 2: kraj\n");
}

int main()
{
    struct sigaction act;
    sigemptyset(&act.sa_mask);
    act.sa_flags = 0;
    act.sa_handler = obrada1;
    sigaction(SIGUSR1, &act, NULL);
    act.sa_handler = obrada2;
    sigaction(SIGUSR2, &act, NULL);
    printf("main: pocetak\n");
    "neki posao (proračun)
    koji traje 10 s,
    ne koristi se sleep";
    printf("main: kraj\n");
    return 0;
}
```