

## Operacijski sustavi za ugrađena računala – *međuispit*

1. (1) Git repozitorij nalazi se na adresi (poslužitelju) `git@adresa:rep`. Prikazati naredbe kojima će se dohvatiti repozitorij s zadane adrese (uz pretpostavku da prethodno nije bio dohvaćen) i nakon promjene pohraniti ga u repozitorij na poslužitelju. Promjenu u nizu naredbi označiti s PROMJENA.
2. (1) Što radi prevoditelj (*compiler*, npr. pokrenut sa: `gcc -c test.c +zastavice`), a što poveziivač (*linker*, npr. sa: `gcc test.o -o test +zastavice`)? Koji su izlazi prevoditelja, a koji poveziivača?
3. (1) Čemu služi sklop za prihvat prekida (*programmable interrupt controller*)? Koje mogućnosti nam on nudi pri upravljanju ulazno-izlaznim napravama?
4. (1) Neki se sustav sastoji od datoteka: `a.h`, `a.c`, `b.h`, `b.c`, `c.h` i `c.c`. Napisati `Makefile` koji će prevesti navedene datoteke i stvoriti program `abc` (predmetom `all`) ali omogućiti i pokretanje programa `abc` (predmetom `install`). Ni prevoditelj ni poveziivač ne trebaju dodatne zastavice.
5. (1) Korištenjem POSIX sučelja za rad s alarmima stvoriti (i aktivirati) alarm koji će svake tri sekunde pozvati funkciju `aktivacija`. Postavljanje elemenata strukture `sigevent_t` se može opisati (nije potrebno poznavati elemente i konstante).
6. (1) Napisati makro koji za ulazne parametre  $a$  i  $b$  računa vrijednost  $\sqrt{2a-3b}$ . Koristiti funkciju `sqrt` za računanje drugog korijena. Neka se makro zove `F1(A,B)`. Pripaziti na ispravnost makroa i u slučaju složenih ulaza za parametre!
7. (1) Neka je zadana struktura:
 

```
struct prva {
    int x;
    struct druga {
        int b;
        struct treca {
            int c;
        } z;
    } y;
};
```

 Napisati makroe `A`, `B` i `C` tako da se za varijable tipa `struct prva` mogu koristiti skraćeni zapisi `.A`, `.B` i `.C`. Npr. kôd `struct prva var; var.A = 5; var.B = 6; var.C = 7;` treba postaviti vrijednosti `var.x` na 5, `var.y.b` na 6 te `var.y.z.c` na 7.
8. (2) U nekom sustavu postoje razni podsustavi (npr. `fs`, `net`, `ui`) od kojih svaki ima ostvarene svoje funkciju za razne operacije, ali istih sufiksa (npr. `prvi_ispisi()`, `drugi_ispisi()` te `treći_ispisi()`). Napisati makro `OP(MODUL,OP,...)` koji će pozvati zadanu operaciju (sufiks na ime modula), ali tako da se i ispiše rezultat operacije (povratna vrijednost funkcije). Primjerice, slijedeći pozivi makroa (lijevo) trebaju generirati kod (desno):
 

```
OP(fs,flush,fp);      => printf("fs_flush:%d\n", fs_flush(fp));
OP(net,close,id,fl);   => printf("net_close:%d\n", net_close(id,fl));
```

9. (3) Napisati skriptu za povezivanje (*linker skriptu*) za sustav koji treba imati odvojene instrukcije od podataka. Instrukcije treba pripremiti za adresu ROM1=0x10000, konstante za ROM2=0x20000, a sve ostalo učitati na ROM3=0x30000 ali pripremiti za adresu RAM=0x40000.
10. (2) Neki sustav ima tri naprave koje izazivaju prekide. Neka se u tom sustavu pri prihvatu prekida skače na adresu 0x04 (ta se vrijednost stavlja u PC nakon što je R0, PC i SR pohranjen na stog) te u registru R0 se postavlja indeks prekida (1, 2 ili 3). Skicirati ostvarenje prekidnog podsustava, tj. samo njegove inicijalizacije `int_init()` tako da se na prekid 1 pozove funkcija `prekid_1`, na prekid 2 `prekid_2` te `prekid_3` na prekid 3. Instrukcije na adresi 0x04, kao i inicijalizacijsku funkciju opisati pseudokodom.
11. (3) Neki sustav posjeduje 32 bitovni registar (brojilo) na adresi 0x10000 koji odbrojava frekvencijom  $10^9$  Hz prema nuli (od zadane vrijednosti). Ostvariti podsustav za praćenje vremena, tj. funkcije:
- ```
void dohvati_vrijeme ( timespec_t *t )
void postavi_vrijeme ( timespec_t *t ) te
void prekid_sata() koji se poziva kad brojilo dođe do nule (brojilo tada učitava zadnje
učitanu vrijednost i nastavlja s brojanjem).
```
- Struktura `timespec_t` ima dva elementa `int tv_sec` i `int tv_nsec`.
12. (2) Zadan je skup izvornih datoteka sa sadržajima:

#### main.c

```
void alarm();
void jedan();
void dva();
int main () {
    postavi_alarm(svakih 2s, alarm);
    while (1) {
        sleep(1);
        jedan();
        dva();
    }
}
```

#### alarm.c

```
int br_alarma = 0;
void alarm () {
    br_alarma++;
    printf ("ALARM\n");
}
```

#### jedan.c

```
int br_alarma;
int brojac = 0;
void jedan(){
    if ( br_alarma % 2 == 0 )
        brojac++;
    printf ( "1:%d\n", brojac );
}
```

#### dva.c

```
int br_alarma;
int brojac = 0;
void dva(){
    if ( br_alarma % 2 != 0 )
        brojac++;
    printf ( "2:%d\n", brojac );
}
```

Nadopuniti kod oznakama `static`, `volatile` i `extern`, po potrebi, a da bi se sustav mogao prevesti i ispravno raditi, uz pretpostavku da varijabla `br_alarma` treba biti zajednička te da funkcija `postavi_alarm` postoji i da za zadani poziv stvara alarm koji svakih 2s poziva funkciju `alarm`.