Međuispit iz OSZUR-a, 24. 4. 2012.

_____

ime i prezime

1. a) (3 boda) Nadopuniti slijedeći kod potrebnim ključnim riječima (`include`, `static`, ...) tako da se kod može prevesti naredbom: `gcc main.c device1.c device2.c -o d2d`.

"device.h"

```
struct device_t {
        int (*init) ();
        int (*recv) ( void *data, size_t size );
        int (*send) ( void *data, size_t size );
}
```

"main.c"

```
#define M       80
int main () {
        char buffer[M];
        size_t size;
        device1.init();
        device2.init();
        while(size = device1.recv ( buffer, M ))
                device2.send ( buffer, size );
        return 0;
}
```

"device1.c"

```
int init () { ... }
int recv ( void *data, size_t size )
{ ... }
int send ( void *data, size_t size )
{ ... }

struct device_t device1 = (struct device_t)
{ .init = init, .recv = recv, .send = send };
```

"device2.c"

```
int init () { ... }
int recv ( void *data, size_t size )
{ ... }
int send ( void *data, size_t size )
{ ... }

struct device_t device2 = (struct device_t)
{ .init = init, .recv = recv, .send = send };
```

b) (2 boda) Napisati *Makefile* za prevođenje gornjih datoteka.

c) (2 boda) Navesti izlazne odjeljke koji će se pojaviti prevođenjem gornjih datoteka te navedite sadržaje tih odjeljaka (koji elementi gornjih datoteka će biti u njima).

2. (3) Napisati makroe (sa `#define IME`) naziva `INC1(N)`, `INC2(N,X)` te `INC3(N,X)` tako da:
   * `INC1(N)` vraća vrijednost za jednu veću od `N`,
   * `INC2(N,X)` vraća vrijednost za jednu veću od `N` ako je `N<X-1` te 0 inače, te
   * `INC3(N,X)` koji povećava varijablu `N` za jedan ako je `N<X-1`, odnosno postavlja ju u 0 inače.

   Makroe napisati tako da budu uporabljivi u svim primjenama (kontekstu) koje imaju smisla (poslani parmetri N i X mogu biti i složeniji izrazi; sam makro može biti dio složenijih izraza, primjerice `INC1` može se koristiti u `INC2` a `INC2` u `INC3`). Po potrebi koristiti "uvjetno" dodjeljivanje:

   `( uvjet ? vrijednost_za_DA : vrijednost_za_NE ).`

3. (2) Neki zamišljeni procesor ima 4 registara opće namjene `R0-R3` te programsko brojilo `PC`, registar stanja `RS` i kazaljku stoga `SP`. Za rad sa stogom ima instrukcije `PUSH registar` i `POP registar` koje stavljaju zadani registar na stog i obnavljaju vrijednost registra sa stoga. Pri prihvatu prekida procesor sam stavlja na stog `PC` i `RS`. Ukoliko sve prekida treba obraditi funkcijom obradi_prekid (`CALL obradi_prekid`), te ukoliko se iz prekida vraćamo instrukcijom `IRET` (koja obnavlja `RS` i `PC` sa stoga i omogućuje prekide) napisati niz instrukcija koje slijede labelu `prihvati_prekid` a koje se izvode po prihvatu prekida (procesor nastavlja obradu prekida tim instrukcijama nakon što je sam na stog pohranio `PC` i `RS`).

```
prihvati_prekid:

        CALL  obradi_prekid

        IRET
```

4. (8 bodova) Ostvariti podsustav za upravljanje vremenom koji omogućuje postavljanje jednog alarma (jedina funkcionalnost). Neka sučelje koje treba ostvariti bude:

   `postavi_alarm ( vrijeme_do_aktiviranja, funkcija ).`

   Nakon isteka zadanog vremena (`vrijeme_do_aktiviranja`, u mikrosekundama, računanog od trenutka postavljanja alarma – poziva `postavi_alarm`) treba pozvati funkciju `funkcija`. Na raspolaganju stoji brojilo koje odbrojava u taktu jedne mikrosekunde, a čija se vrijednost (u mikrosekundama) postavlja sa `postavi_brojilo ( int broj )` (sa `broj=0` se brojanje isključuje) i čita sa `pročitaj_brojilo ( int *broj )` (na adresu `broj` se upisuje trenutna vrijednost brojila). Po dostizanju vrijednosti nula, brojilo izaziva prekid `PREKID_BROJILA` koji se može povezati funkcijom za obradu prekida pozivom `registriraj_prekid ( ID_PREKID, funkcija_obrade )`. Neka se podsustav, tj. sve funkcije koje ga sačinjavaju, od `postavi_alarm`, `obrada_prekida_sata` te `inicijaliziraj()`, kao i sve potrebne varijable nalaze u datoteci `alarm.c`. Napisati sadržaj te datoteke. Radi jednostavnosti vrijeme izražavati u mikrosekundama i pretpostaviti da neće doći do prekoračenja opsega brojeva tipa `int` pri njegovu korištenju za tu svrhu te da je brojilo dovoljno veliko da prihvati sve intervale.