Lab-2. Dretve i semafori

Dretve

Višedretvenost omogućava bolju iskorištenost računalnog sustava paralelnim radom dretvi istog ili različitih procesa. Tako se mogu iskoristiti višeprocesorski sustavi, ali i razni drugi elementi računala (disk, mreža, ulazno-izlazne naprave).

U nastavku je opisano stvaranje dretvi prema POSIX sučelju.

Stvaranje dretve obavlja se funkcijom `pthread_create` koja prima nekoliko argumenata:

1. Prvi argument je kazaljka na element tipa `pthread_t` – „opisnik dretve“
2. Drugi argument se može koristiti da se detaljnije definiraju parametri dretve, primjerice, stog, način raspoređivanja i slično. Postavljanjem vrijednosti `NULL` koriste se uobičajene vrijednosti.
3. Treći argument je kazaljka na funkciju koja će biti početna funkcija nove dretve.
4. Četvrti argument je kazaljka koja će se poslati kao argument početne funkcije dretve.

Primjer jednostavnog programa koji stvara nekoliko dretvi te potom čeka da one završe s radom je u nastavku.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>

#define BROJ_DRETVI 3
#define BROJ_ITERACIJA 5

void *dretva (void *rbr)
{
    int i, *d = rbr;
    for (i = 1; i <= BROJ_ITERACIJA; i++) {
        printf("Dretva %d; iteracija=%d\n", *d, i);
        sleep(1);
    }
    return NULL;
}
int main(int argc, char *argv[])
{
    pthread_t opisnik[BROJ_DRETVI];
    int i, j, id[BROJ_DRETVI];

    for (i = 0; i < BROJ_DRETVI; i++) {
        id[i] = i;
        if (pthread_create(&opisnik[i], NULL, dretva, &id[i])) {
            fprintf(stderr, "Ne mogu stvoriti novu dretvu!\n");
            exit(1);
        }
    }

    for (j = 0; j < BROJ_DRETVI; j++)
        pthread_join(opisnik[j], NULL);

    return 0;
}
```

Stvorena dretva radi dok ne završi sa zadanom funkcijom – dok ne izađe iz nje, ili to može napraviti i prije pozivom `pthread_exit()`.

Početna dretva nakon stvaranja novih, može preko `pthread_join()` čekati na završetak stvorene dretve.

Semafori `sem_wait/post`

Sučeljem `sem_*` ostvaruje se opći semafor (OSEM s predavanja). Ulazak i izlazak iz kritična odsječka može se ostvariti ovim semaforom uz početnu vrijednost 1.

```
int sem_init(sem_t *sem, int mprocesi, unsigned int koliko);
int sem_post(sem_t *sem); – PostaviOSEM
int sem_wait(sem_t *sem); – čekajOSEM
int sem_destroy(sem_t *sem); – obriši semafore
```

Koncept korištenja

```
sem_t s; //globalna varijabla
      //inicijalizacija, npr. u funkciji main() prije stvaranja dretvi
      sem_init(&s, 0, 5); //početna vrijednost = 5

            //u novim dretvama
            sem_wait(&s); //ČekajOSEM
            ...
            sem_post(&s); //PostaviOSEM

      //u funkciji main, nakon završetka ostalih dretvi
      sem_destroy(&s);
```

Primjeri korištenja dretvi i semafora na webu.

Zadatak

Nadograditi program iz prve vježbe (ali u `lab2`) tako da se koristi više dretvi za izračun (barem tri). Korištenje zajedničkih sredstava (međuspremnika, generiranje slučajnih brojeva) zaštititi semaforima. Sinkronizaciju ubaciti tamo gdje je neophodna, ne nepotrebno zaključavati veće dijelove koda (npr. dodati ju u kod međuspremnika i generatora, tj. u već postojeće funkcije `zaključaj`/`otključaj` ostvarene u `postavke.h`).

Mogući problemi i rješenja

Najjednostavnije rješenje uključuje stvaranje dretvi koje preuzimaju petlju koja je prije bila u `main` dok glavna dretva samo stvara nove te potom čeka na njihov kraj. Problem s navedenim jest da se pri popunjenju međuspremnika dretve zaustavljaju s `pause()` koji zaustavlja dretvu do primitka bilo kojeg signala. Međutim, signale upućene procesu prihvaća glavna (početna) dretva te se ove stvorene ne odblokiraju na svojem `pause()`. To se može riješiti tako da se u obradi signala svim dretvama pošalje signal, ali ne isti (da ne rade opet isti posao). Npr. maskirati i signal `SIGCONT` (istom funkcijom, ali bez ikakve akcije) te taj signal poslati svim stvorenim dretvama na svaki signal upućen procesu.

Drugo rješenje može biti korištenjem dodatnih semafora, za svaku dretvu po jedan. Tada, umjesto `pause()` staviti `sem_wait(&sem[id])` a na svaki signal svima postaviti taj semafor.