VJEŽBA 1: POSIX niti

Nazivom **POSIX** (engl. *Portable Operating System Interface*) označava se grupa IEEE standarda čiji je cilj održavanje kompatibilnosti između različitih operacijskih sustava. Jedan od najvažnijih standarda za paralelno procesiranje podataka i izradu niti (engl. *threads*) donesen je 1995. godine i pripada ovoj skupini standarda. Naziva se **IEEE Std POSIX 1003.1c-1995 standard**. Niti implementirane slijedeći pravila predložena ovim standardom nazivaju se **POSIX niti** ili **P-niti** (engl. *Pthreads*) [1].

U računarstvu, **niti** se najbolje mogu definirati kao dijelovi koda nekog računalnog programa koji se mogu izvršavati istovremeno jer ne ovise jedni o drugima. Niti su uglavnom uvijek vezane za neki **proces** (tj. računalni program koji se izvršava), i ne mogu postojati samostalno. Unutar jednog procesa može postojati više niti.

Ukoliko se višenitni proces izvršava na računalu sa jednim procesorom, onda se procesorsko vrijeme dijeli između niti koje je potrebno izvršiti, tj. procesor neprestalno prebacuje izvođenje sa jedne niti na drugu sve dok se one ne izvrše. Ovo prebacivanje se odvija jako brzo i korisnik ga percipira kao istovremeno izvršavanje svih niti (engl. *multithreading*). U stvarnosti, niti će se istovremeno izvršavati jedino ako se izvršavaju na višeprocesorskom računalu.

Kada koristiti niti? Evo nekoliko primjera:

- za istovremeno obrađivanje zahtjeva koji pristižu na server kod klijent-server arhitekture,
- ukoliko je potrebno vršiti neke duže proračune i istovremeno obavještavati korisnika o tijeku tih operacija,
- za skraćivanje vremena izvršavanja nekog programa,
- itd.

Bilo koji program koji napišete ima barem jednu nit (engl. *default thread*), koja odgovara *main()* funkciji i eventualnim funkcijama koje se iz nje pozivaju. Ako želite koristiti više niti u svom programu, morate ih eksplicitno definirati [1].

Da bi se mogle koristiti POSIX niti, na početku C programa potrebno je uključiti **<pthread.h>**. Neke od najvažnijih funkcija koje dolaze sa ovim *header* dokumentom i pripadajućom bibliotekom nalaze se u *Tablici 1*.

Tablica 1. Funkcije za rad sa POSIX nitima

Ime funkcije	**Opis**	**Argumenti**
pthread_create	kreira novu nit	int pthread_create(pthread_t *_thread_, const pthread_attr_t *_attr_, void *(*_start_routine_)(void*), void *_arg_); Funkcija prima: • pokazivač na pthread_t objekt gdje se može spremiti identifikator nove niti; • pokazivač na pthread_attr_t strukturu koja specificira atribute nove niti (NULL po *default-u*); • ime procedure u kojoj nova nit započinje izvršavanje (kada se procedura izvrši, implicitno se poziva pthread_exit() funkcija); • argument koji se šalje proceduri u kojoj nova nit započinje sa izvršavanjem. Funkcija vraća: • 0 ako je uspješno izvršena; • neki drugi broj ako nije uspješno izvršena (ovaj broj označava do kakve je greške došlo tijekom izvršavanja funkcije).
pthread_exit	eksplicitno zaustavlja nit	void pthread_exit (void *status); Funkcija prima: • pokazivač na svoj status, tako da može signalizirati drugim nitima da je njezino izvođenje završeno.
pthread_cancel	poništava nit	int pthread_cancel (pthread_t thread); Funkcija prima: • identifikator niti koju je potrebno poništiti.

Primjer kreiranja niti u C-u:

```c
#include <stdio.h>
#include <stdlib.h>     // potrebno za funkciju exit()
#include <pthread.h>    // potrebno za rad sa nitima

void *printMessage (void *thread_id)
{
      int thread_id_2;
      thread_id_2 = (int)thread_id; // cast je potreban zato sto je
                                    // thread_id tipa (void *)

      printf("Javlja se nit broj %d!\n", thread_id_2);
      pthread_exit(NULL);
}

int main()
{
      int thread_id = 0;      // identifikator niti
      int noThreads = 3;      // koliko niti zelimo kreirati
      int result = 0;         // rezultat pthread_create funkcije

      // koristi se za identifikaciju niti
      // (pthread_t je apstraktni tip podataka koji se koristi kao handle ili
      // referenca na nit)
      pthread_t threads[noThreads];

      for (thread_id = 0; thread_id < noThreads; thread_id++)
      {
            printf("main() funkcija: kreiram nit broj %d\n", thread_id);
            result = pthread_create(&threads[thread_id],
                                    NULL,
                                    printMessage,
                                    (void *)thread_id);

            if (result != 0)  // ako nit nije uspjesno kreirana
            {
                  printf("Doslo je do greske prilikom kreacije niti. Sifra greske
                          je %d.\n", result);
                  exit(-1);
            }
      }

      pthread_exit(NULL);

      return 0;
}
```

Ako gornji program sačuvamo pod nazivom lab_vj_1.c, onda se on iz Unix komandne linije kompajlira i poziva na slijedeći način:

```
gcc -Wall -pthread -o lab_vj_1 lab_vj_1.c      // kompajliraj program
./lab_vj_1                                     // izvrši program
```

Gornje su naredbe detaljnije objašnjene u *Tablici 2*.

Tablica 2. Unix naredbe za kompajliranje programa

Dio naredbe	**Objašnjenje**
gcc	(engl. *GNU compiler collection*) naredba za pozivanje kompajlera
-Wall	naredba za ispisivanje eventualnih upozorenja (engl. *warnings*) do kojih može doći tijekom kompajliranja programa
-pthread ili -lpthread	opcija potrebna za kompajliranje programa koji koriste pthread biblioteku funkcija
-o lab_vj_1	nazovi .exe datoteku lab_vj_1 umjesto a.out (a.out je *default*-na vrijednost)
lab_vj_1.c	specificiraj ime .c datoteke u kojoj se nalazi programski kod
./lab_vj_1	izvrši program

Ispis programa lab_vj_1:

```
main() funkcija: kreiram nit broj 0
main() funkcija: kreiram nit broj 1
Javlja se nit broj 0!
main() funkcija: kreiram nit broj 2
Javlja se nit broj 1!
Javlja se nit broj 2!
```

Obratite pozornost na to da ispis programa neće uvijek biti isti, jer procesor neće uvijek niti izvršavati istim redoslijedom (osim ako se ne koristi poseban mehanizam za definiranje rasporeda njihovog izvršavanja)!

Literatura:

[1] Blaise Barney, "POSIX Threads Programming", Lawrence Livermore National Laboratory, UCRL-MI-133316.
Dostupno na: https://computing.llnl.gov/tutorials/pthreads/