

Multiprocesorski sistemi (SI4MPS, IR4MPS, MS1MPS)

Laboratorijska vežba 1 - Pthreads

Cilj laboratorijske vežbe je upoznavanje studenata sa korišćenjem Pthreads niti na platformama Windows i Linux i izrada, prevođenje i debugovanje jednostavnog programa koji koristi Pthreads niti.

Podešavanja okruženja Visual Studio

Za rad sa Pthreads nitima na Windows platformi može se koristiti pthreads-win32 implementacija koja se može preuzeti na adresi <http://sourceware.org/pthreads-win32/>. Ova implementacija Pthreads niti se mogu koristiti bilo kroz statičko vezivanje odgovarajuće biblioteke sa napisanim programom, bilo korišćenjem odgovarajućeg DLL-a.

Podešavanja se u svakom trenutku mogu preuzeti u arhivi koja se nalazi na sajtu predmeta u odgovarajućoj sekciji (<http://mups.etf.rs/vezbe/pthreads/pthreads-win32.zip>). Prema uputstvima u priloženom readme.txt fajlu podesiti okruženje za razvoj i kontrolisano izvršavanje (engl. debugging) Pthreads programa na lokalnom računaru.

Alternativno, na http://mups.etf.rs/vezbe/pthreads/code/pthreads-win32_pokazni_primeri.zip se može preuzeti arhiva koja sadrži skup projekata koji prikazuju kako se mogu koristiti Pthreads niti sa statičkim vezivanjem odgovarajuće biblioteke. Pomenuta arhiva sadrži izvorni kod pthreads-win32 implementacije, a u priloženom ReadMe.txt je detaljno objašnjen način podešavanja projekta kod statičkog povezivanja.

Tehničke napomene

Pre početka rada, treba podesiti Visual Studio tako da radi sa VC++ podešavanjima. U dijalogu Tools->Import and Export Settings... izabрати "Reset All Settings", na sledećem izabрати "No" i na poslednjem izabрати "Visual C++ Development Settings".

Podešavanje projekta za uspešno prevođenje

Najpre je potrebno napraviti novi Visual Studio projekat, tipa "Win32 Console Application" (File->New Project->Visual C++ Project->General->Empty Project). Zatim dodati jedan novi izvorni fajl. U okviru direktorijuma koji sadrži kreirani projekat potrebno je prekopirati pthreads_win32 direktorijum koji sadrži neophodne biblioteke.

C/C++ podešavanja:

1. Dodati " pthreads_win32/include" u General->Additional Include Directories.
2. Podesiti Advanced->Compile As na "Compile as C Code (/TC)" ili "Compile as C++ Code (/TP)", zavisno od jezika u kome je izvorni programski kod. Ovo nije neophodno ako izvorni fajlovi imaju ekstenziju .c, odnosno .cpp, pošto za fajlove sa tim ekstenzijama VS podrazumeva da sadrže programski kod na jeziku C, odnosno C++.
3. Po želji, dodati _CRT_SECURE_NO_DEPRECATED u Preprocessor->Preprocessor Definitions radi izbegavanja nepotrebnih upozorenja.
4. Postaviti Properties->C/C++->General->Detect 64 Bit Portability Issues na "false" radi izbegavanja upozorenja o 64-bit kompatibilnosti programa.

Podešavanje linkera:

1. Dodati "pthread_win32/lib" u General->Additional Library Directories.
2. Dodati "pthreadVC2.lib" u Input->Additional Dependencies. Alternativno, može se koristiti neka druga verzija pthreads-win32 biblioteke, kao što će biti objašnjeno kasnije.

Za ispravno izvršavanje programa potrebno je da odgovarajući DLL (pthreadVC2.dll) bude dostupan u direktorijumu gde se nalazi izvršni fajl ili u nekom od sistemskih direktorijuma (na primer, C:\WINDOWS\System32).

Umesto para biblioteka pthreadVC2.lib i pthreadVC2.dll moguće je koristiti i neke druge odgovarajuće parove biblioteka iz pthreads_win32 direktorijuma. Biblioteke imaju sledeće značenje:

```
pthread[VG]{SE,CE,C}c.dll  
pthread[VG]{SE,CE,C}c.lib
```

gde [VG] označava prevodilac korišćen za prevođenje pthreads-win32 biblioteke:

```
V    Microsoft Visual C prevodilac  
G    GNU C prevodilac
```

Oznake {SE,CE,C} označavaju način obrade izuzetaka:

```
SE    Strukturirana obrada izuzetaka (Microsoft specifična obrada izuzetaka za Windows)  
CE    Standardni C++ način obrade izuzetaka  
C     Ne koriste se izuzeci – koriste se setjmp/longjmp funkcije
```

Broj "c" na kraju označava verziju DLL koja zavisi od verzije izvornog koda same biblioteke i koristi se za proveru kompatibilnosti. Biblioteke koje sadrže slovo "d" nakon broja verzije označavaju varijante koje sadrže dodatne debugging informacije. Više informacija se može naći u dokumentaciji pthreads-win32.

Debugging multithreaded aplikacija koristeći Visual Studio

Proces debugovanja se vrši kroz prozor Threads do koga se može doći klikom na Debug->Windows->Threads kada je aktivna debug sesija. U datom prozoru je moguće izabrati određenu nit za izvršavanje (Switch To Thread opcija), moguće je zamrznuti izvršavanje ostalih niti (opcija Freeze). Takođe, moguće je svakoj niti dati ime i označiti je odgovarajućom bojom ili zastavicom, što olakšava praćenje izvršavanja.

Više detalja o celom procesu za debugging na Visual Studio se može pronaći u dokumentu na adresi <http://msdn.microsoft.com/en-us/library/ms164746%28v=VS.90%29.aspx>.

Korišćenje Pthreads na Linux

Pthreads biblioteka je na Linux operativnom sistemou dostupna na rtidev4 serveru. Prevođenje i povezivanje programa se može izvršiti pomoću sledeće komandne linije:

- Programski jezik C: gcc -pthread -lm -o dz1z1.exe dz1z1.c
- Programski jezik C++: g++ -pthread -lm -o dz1z1.exe dz1z1.cpp

Zadatak 1

Sastaviti program koji ispisuje „Hello World!“. Glavni program treba najpre da učiata broj niti koji treba da kreira. Svaka nit treba da ispiše pozdravnu poruku i jedinstveni identifikacioni broj koji joj je prosledila glavna nit. [1,N]

Zadatak 2

Sastaviti program koji ispisuje ukupan broj parnih i neparnih brojeva u nizu celih brojeva. Glavni program treba najpre da učiata broj niti koji treba da kreira, a zatim i niz celih brojeva. Ukoliko dimenzija niza nije umnožak broja niti, prekinuti program. Glavna nit treba da sakupi sve rezultate i ispiše konačan rezultat. Zadatak rešiti korišćenjem pthread_join funkcije. Rezultat rada nizi proslediti bilo putem zadate funkcije bilo putem korišćenja globalnih promenljivih.

Zadatak 3

Sastaviti program koji računa sumu elemenata niza celih brojeva. Glavni program treba da učiata niz, a zatim i broj niti koji treba da kreira. Potrebno je nitima što približnije raspodeliti delove niza na obradu. Zadatak rešiti korišćenjem brava (pthread_mutex_t). [2,N]

Mali Pthreads podsetnik

```
int pthread_create(pthread_t *handle, const pthread_attr_t *attr,
                  void *(*start_routine)(void*), void *arg);
void pthread_exit(void *value_ptr);
int pthread_join(pthread_t thread, void **value_ptr);

int pthread_attr_init(pthread_attr_t *attr);
int pthread_attr_destroy(pthread_attr_t *attr);

int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);
int pthread_mutex_destroy(pthread_mutex_t *mutex);
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```