Inspiratron.org - Natural language processing, machine learning and cybersecurity

Koriscenje niti i sinhronizacija u jeziku C

by Nikola Miloševi? - Thursday, February 18, 2010

https://inspiratron.org/blog/2010/02/18/koriscenje-niti-i-sinhronizacija-u-jeziku-c/

Kada nekoliko razlicitih aktivnosti treba obaviti istovremeno u okviru jednog procesa, najpriruodnije je koriscenje niti. Niti dele adresni prostor, tako da ne zauzimaju prostor, medjutim podela adresnog prostora dovodi do problema sinhronizacije. Kako da se sinhronizuje da jedna nit ne promeni nesto sto je trebala nepromenjeno da iskoristi druga nit. Tu na snagu stupaju mutex objekti i kriticne sekcije koje obezbedjuju da se odredjeni segment koda izvrsi atomicno. Sledeci primeri pokazuju nekoliko nacina implementacije niti i kriticnih sekcija na jezuku C. U prvom primeru se koriste kriticne sekcije, koje obezbedjuju slicno ponasanje mutex objektu, ali mogu biti iskorisceni samo od niti jednog procesa. Dok mutex objekat moze biti koriscen iz niti proizvoljnih procesa. Samo jedna nit u datom trenutku moze da poseduje mutex i da radi s njim nesto. To je demonstrirano u drugom primeru.

Primer 1

```
#include
 #include
 #include CRITICAL_SECTION cs;
 int a[ 5 ];

void Thread( void* pParams )
 {
 int i, num = 0;

while ( TRUE )
 {
 EnterCriticalSection( &cs );
 for ( i = 0; i  LeaveCriticalSection( &cs );
 num++;
 }
 }

int main( void)  {
 InitializeCriticalSection( &cs );
 _beginthread( Thread, 0, NULL );

while( TRUE )
 {
 EnterCriticalSection( &cs );
 printf( “%d %d %d %d %dn”,
a[ 0 ], a[ 1 ], a[ 2 ],
 a[ 3 ], a[ 4 ] );
 LeaveCriticalSection( &cs );
 }
 return 0;
 }
```

Primer 2

```
#include
 #include
 #include HANDLE hMutex;
 int a[ 5 ];

void Thread( void* pParams )
 {
 int i, num = 0;

while ( TRUE )
 {
 WaitForSingleObject( hMutex, INFINITE );
 for ( i = 0; i  ReleaseMutex( hMutex );
 num++;
 }
 }

int main( void )
 {
 hMutex = CreateMutex( NULL, FALSE, NULL );
 _beginthread( Thread, 0, NULL );

while( TRUE )     {
 WaitForSingleObject( hMutex, INFINITE );
 printf( “%d %d %d %d %dn”,
a[ 0 ], a[ 1 ], a[ 2 ],
 a[ 3 ], a[ 4 ] );
 ReleaseMutex( hMutex );
 }
 return 0;
 }
```

Nikola Miloševi?