Параллельные потоки

Windows Threads

Потоки MS Windows

- Операционная система содержит готовые системные вызовы для работы с потоками
- Компилятор поддерживает вызов функций работы с потоками
- Организация взаимодействия потоков – ответственность программиста

Создание потока

- Win32
- CreateThread

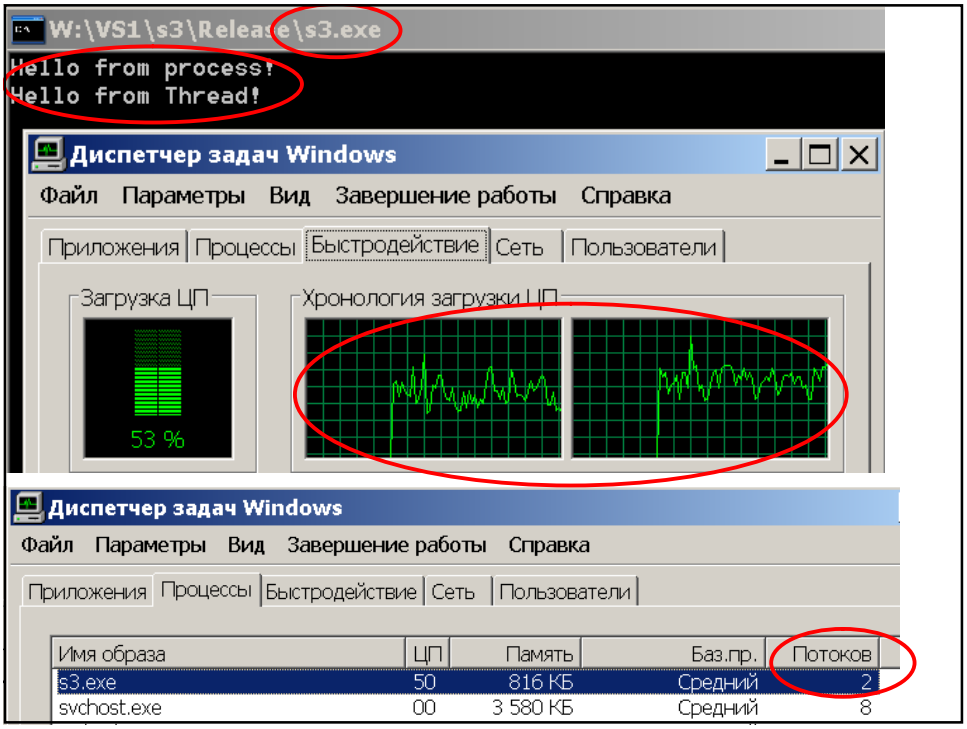
```
HANDLE Thread;
Thread = CreateThread(NULL, 0,
  ThreadProc,
  (LPVOID)&GlobalVar, 0, NULL);
```

CreateThread

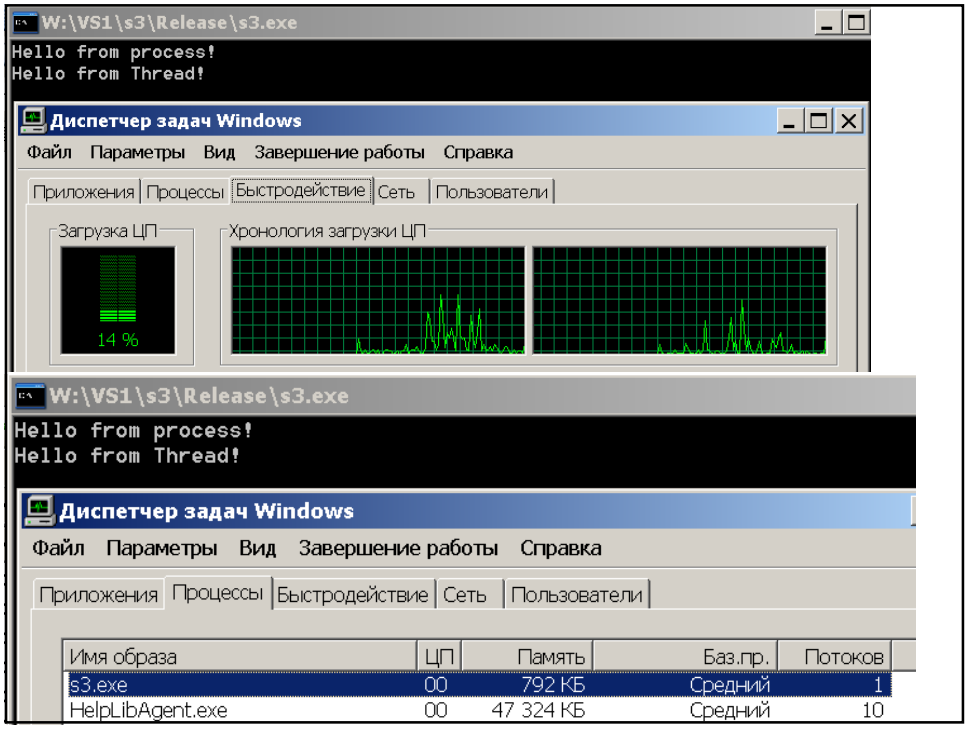
Создает поток в виртуальном адресном пространстве вызывающего процесса

```
HANDLE WINAPI CreateThread(
  __in_opt   LPSECURITY_ATTRIBUTES
  lpThreadAttributes,
  __in       SIZE_T dwStackSize,
  __in       LPTHREAD_START_ROUTINE lpStartAddress,
  __in_opt   LPVOID lpParameter,
  __in       DWORD dwCreationFlags,
  __out_opt  LPDWORD lpThreadId
);
```

```
#include <windows.h>
#include <stdio.h>

void WINAPI ThreadProc()
{
printf("Hello from Thread!\n");
while(1);
}

void main()
{
  HANDLE Thread;
  Thread = CreateThread(0, 0,
  (LPTHREAD_START_ROUTINE) ThreadProc, 0, 0, 0);
  printf("Hello from process!\n");
  getchar();
}
```



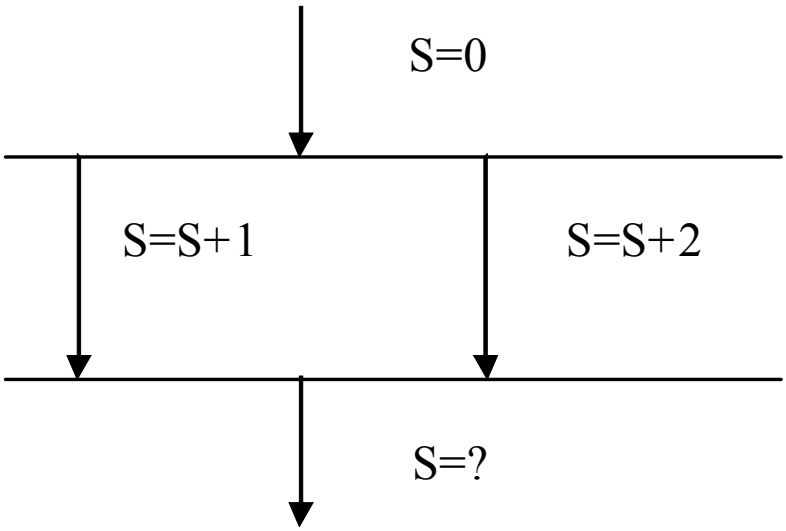
```
#include <windows.h>
#include <stdio.h>

void WINAPI ThreadProc()
{
printf("Hello from Thread!\n");
//while(1);
}

void main()
{
  HANDLE Thread;
  Thread = CreateThread(0, 0,
  (LPTHREAD_START_ROUTINE) ThreadProc, 0, 0, 0);
  printf("Hello from process!\n");
  getchar();
}
```



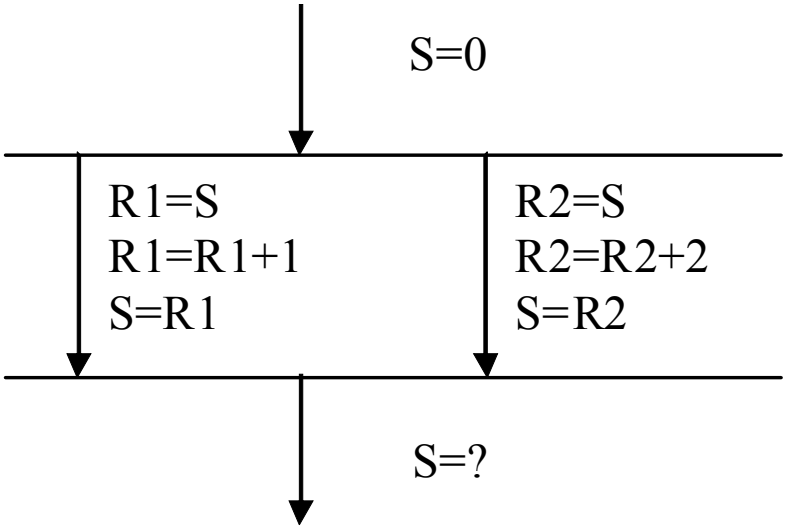
Data Race

- Гонки, соревнование по доступу к данным
- Параллельный доступ вычислительных потоков к глобальным переменным
- Нарушение «целостности данных»
- Появление случайной ошибки в расчетах

Параллельное суммирование



Чтение-запись регистры-ОЗУ

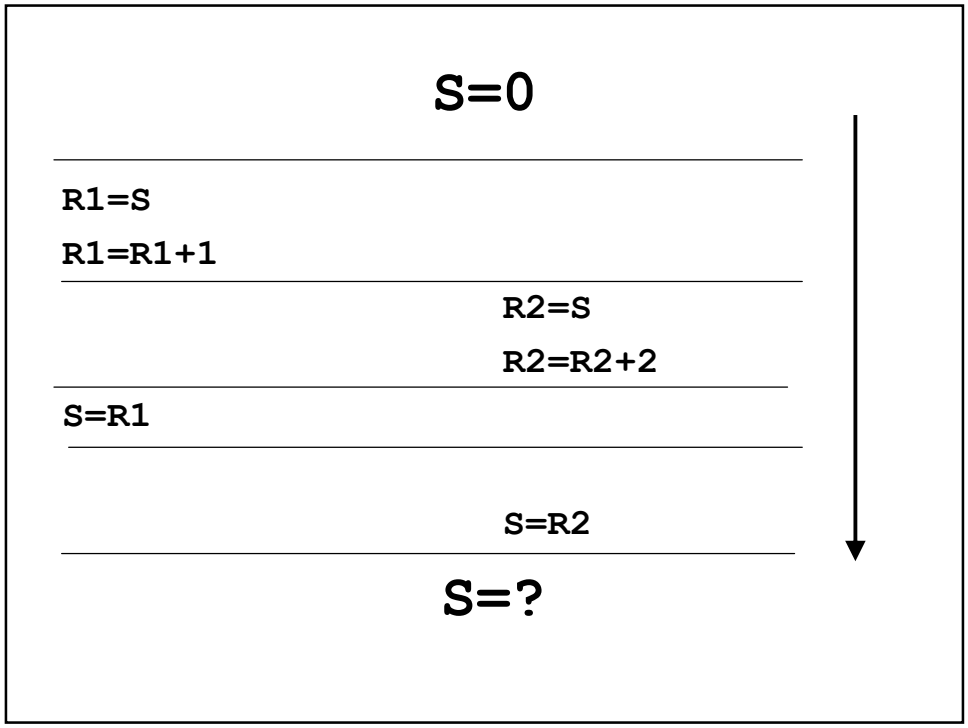


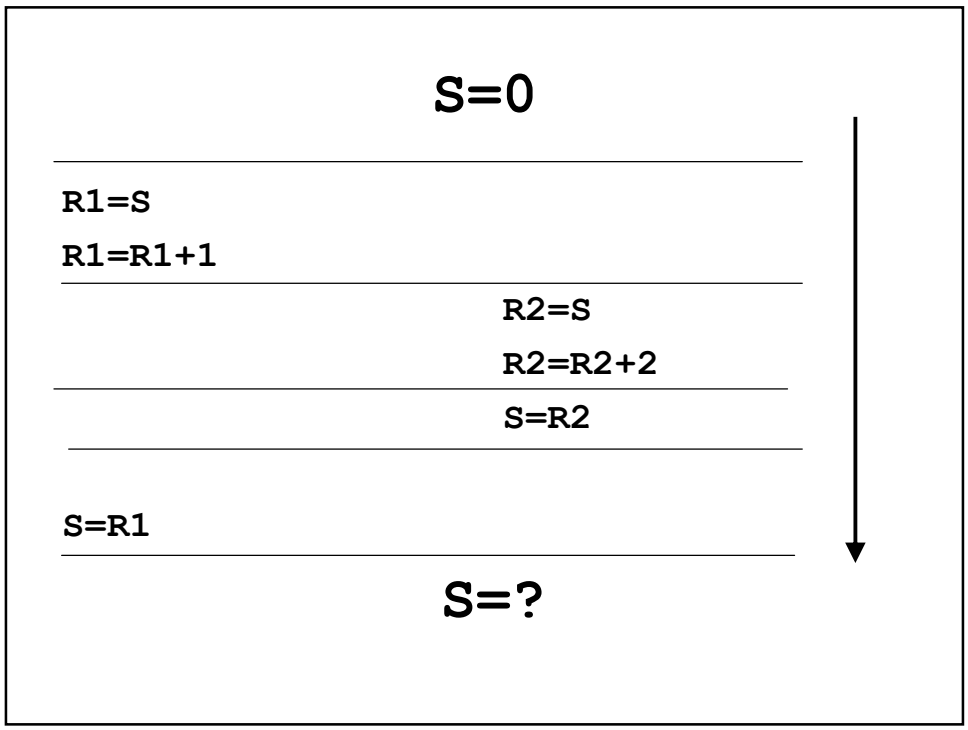
S=0

```
R1=S
R1=R1+1
S=R1
```

```
R2=S
R2=R2+2
S=R2
```

S=?





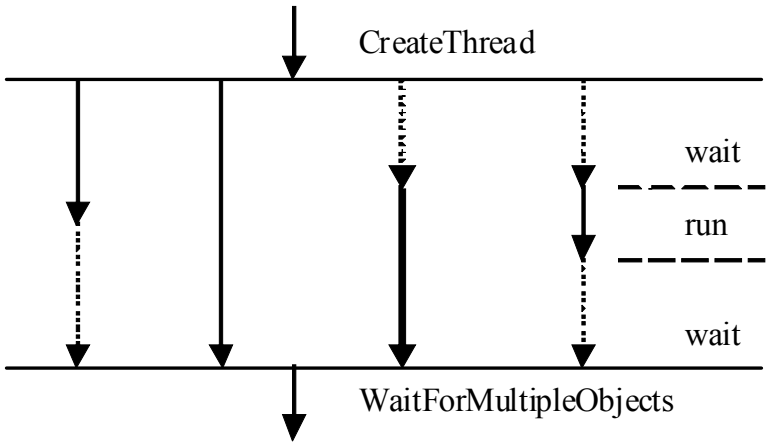
```
#include <windows.h>
#include <stdio.h>
#define NTh 200
#define Ns 10000000
int S = 0;
void WINAPI ThreadProc()  {
for (int i=0; i<Ns; i++)
S = S + 1;  }

void main()  {
HANDLE Thread [NTh];
for (int i=0; i < NTh; i++)
Thread [i] = CreateThread(0, 0,
  (LPTHREAD_START_ROUTINE) ThreadProc, 0, 0, 0);
WaitForMultipleObjects(NTh,Thread,TRUE,INFINITE);
printf("Ns*NTh = %d\n     S = %d\nError =
  %d\n",Ns*NTh,S,Ns*NTh-S);
getchar();  }
```

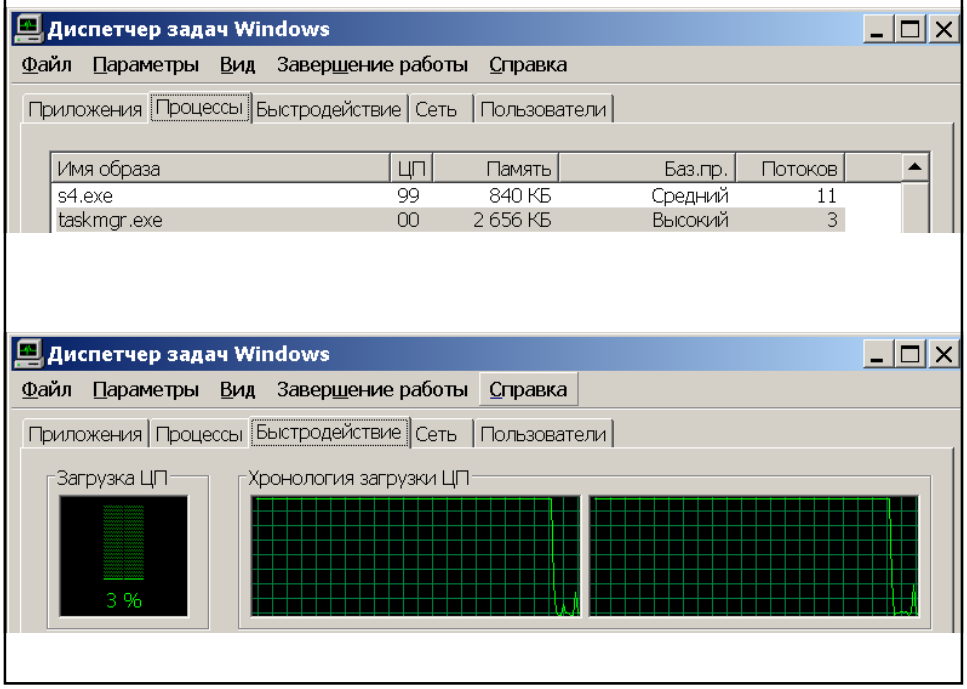
WaitForMultipleObjects

Ожидание завершения работы всех потоков

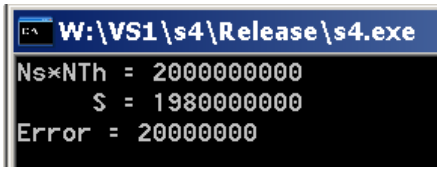
```
WaitForMultipleObjects(NTh,
 Thread,TRUE,INFINITE)
```

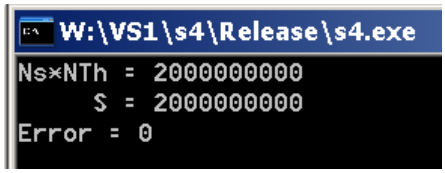
Барьерная синхронизация

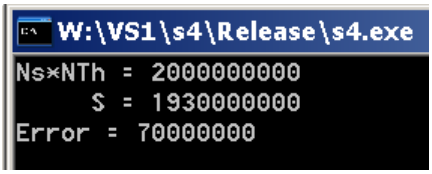


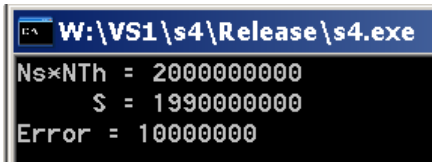


```
#define NTh 200
#define Ns 10000000
int S = 0;
…
printf("Ns*NTh = %d\n      S = %d\nError
  = %d\n",Ns*NTh,S,Ns*NTh-S);
```





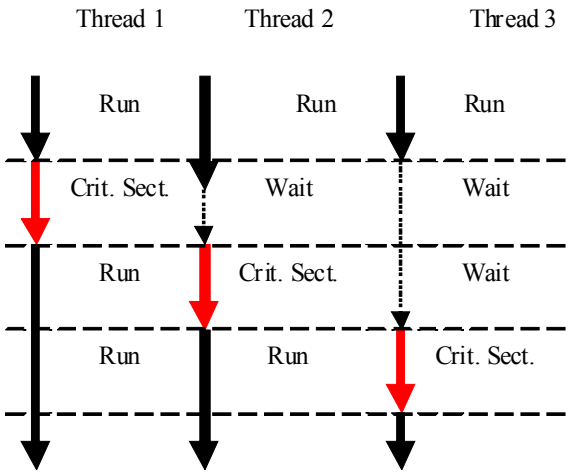




Критическая секция

- Часть программного кода параллельной программы, выполняемая последовательно
- Когда один поток начинает выполнять критическую секцию, остальные потоки могут выполнять другие операции, но не могут параллельно войти в свою критическую секцию

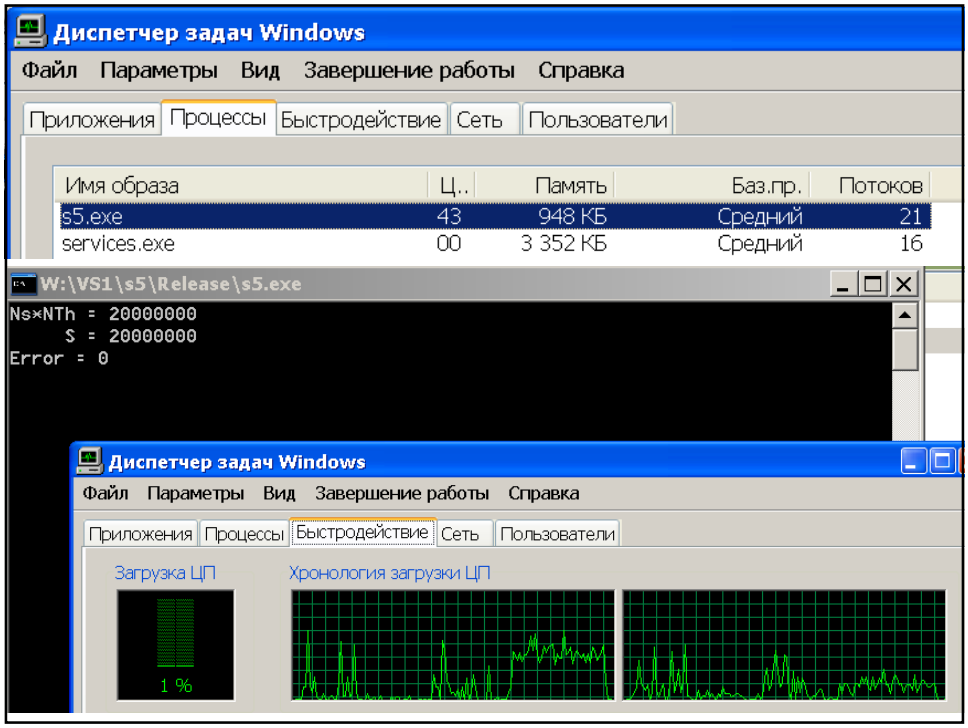
Критическая секция




```
// Объявляем глобальную переменную CS
CRITICAL_SECTION CS;
void main()
{
    // Инициализируем критическую секцию
    InitializeCriticalSection(&CS);
...
    // Удаляем критическую секцию
    DeleteCriticalSection(&CS)
}
DWORD WINAPI ThreadProc( LPVOID lpParameter )
{
    // Входим в критическую секцию
    EnterCriticalSection(&CS);
...
    // Выходим из критической секции
    LeaveCriticalSection(&CS);
}
```

```
#include <windows.h>
#include <stdio.h>
#define NTh 20
#define Ns 1000000
int S = 0;
CRITICAL_SECTION CS;
void WINAPI ThreadProc()
{
for (int i=0; i<Ns; i++)
  {
     EnterCriticalSection(&CS);
     S = S + 1;
     LeaveCriticalSection(&CS);
  }
}
```

```
void main()
{
HANDLE Thread [NTh];
InitializeCriticalSection(&CS);
for (int i=0; i < NTh; i++)
Thread [i] = CreateThread(0, 0,
  (LPTHREAD_START_ROUTINE) ThreadProc, 0, 0, 0);
WaitForMultipleObjects(NTh,Thread,TRUE,INFINITE);
DeleteCriticalSection(&CS);
printf("Ns*NTh = %d\n     S = %d\nError =
  %d\n",Ns*NTh,S,Ns*NTh-S);
getchar();
}
```


Диспетчер задач Windows
Файл Параметры Вид Завершение работы Справка
Приложения Процессы Быстродействие Сеть Пользователи
Имя образа Ц.. Память Баз.пр. Потоков
s5.exe 43 948 КБ Средний 21
services.exe 00 3 352 КБ Средний 16
W:\VS1\s5\Release\s5.exe
Ns*NTh = 20000000
$ = 20000000
Error = 0
Диспетчер задач Windows
Файл Параметры Вид Завершение работы Справка
Приложения Процессы Быстродействие Сеть Пользователи
Загрузка ЦП
1 %
Хронология загрузки ЦП