

# Создание потока

```
int pthread_create
```

```
(pthread_t *, const pthread_attr_t *, void *(*func)(void *), void *)
```



(OUT) Идентификатор потока



- Атрибуты потока
- NULL



Указатель на потоковую функцию



- Указатель на аргумент функции
- NULL

# Основа программы

```
#include "pthread.h"
```

```
void *func(void *param);
```

```
int main(int argc, char *argv[]) {  
    pthread_t tid;  
    pthread_attr_t attr;  
    pthread_attr_init(&attr);  
    pthread_create(&tid, &attr, potok, argv[1]);  
    pthread_join(tid, NULL);  
}
```

```
void *potok(void *param) {  
    ...  
    pthread_exit(0);  
}
```

# Завершение потока

```
void pthread_exit (void *retval)
```

- Поток завершает выполнение задачи когда:
  - *поток*овая функция выполняет **return** и возвращает результат произведенных вычислений;
  - в результате вызова завершения исполнения потока **pthread\_exit()**;
  - в результате вызова отмены потока **pthread\_cancel()**;
  - одна из нитей совершает вызов **exit()**
  - основная нить в функции **main()** выполняет **return**, и в таком случае все нити процесса сворачиваются.
- Если в последнем варианте **старшая** нить (из функции **main()**) выполнит **pthread\_exit()** вместо просто **exit()** или **return**, то остальные нити продолжат исполняться.

# Досрочное завершение потока

```
int pthread_cancel (pthread_t thread_id)
```

- Возвращается сразу
- Не является средством принудительного завершения потоков
- Следует рассматривать как запрос на выполнение досрочного завершения потока.
- Если важно, чтобы поток был удален, нужно дождаться его завершения через `pthread_join()` или `pthread_testcancel()`

# Ожидание потока

```
int pthread_join (pthread_t thread_id, void ** data)
```



- (OUT) Данные потока
  - NULL
- 
- Несколько потоков не могут ждать завершения одного.
  - Если они пытаются выполнить это, один поток завершается успешно, а все остальные — с ошибкой ESRCH.
  - После завершения pthread\_join() пространство стека, связанное с потоком, может быть использовано приложением.

# Отсоединение потока

```
int pthread_detach (pthread_t thread_id)
```

- Отсоединенный поток — приговор
  - *Отсоединенный поток нельзя перехватить с помощью `pthread_join()`, чтобы получить статус завершения и прочее*
  - *Нельзя отменить его отсоединенное состояние*
- Что будет, если завершение потока не перехватить вызовом `pthread_join()` и чем это отлично от сценария, при котором завершился отсоединенный поток?

## Компиляция программы

```
gcc superhot.c -o hot -lpthread (-lrt -pthread)
```

## Запуск программы

```
./hot
```

# Общие процедуры PThread. Задача 1

- Составить программу «Hello, world!»
- Каждый поток выводит свой номер