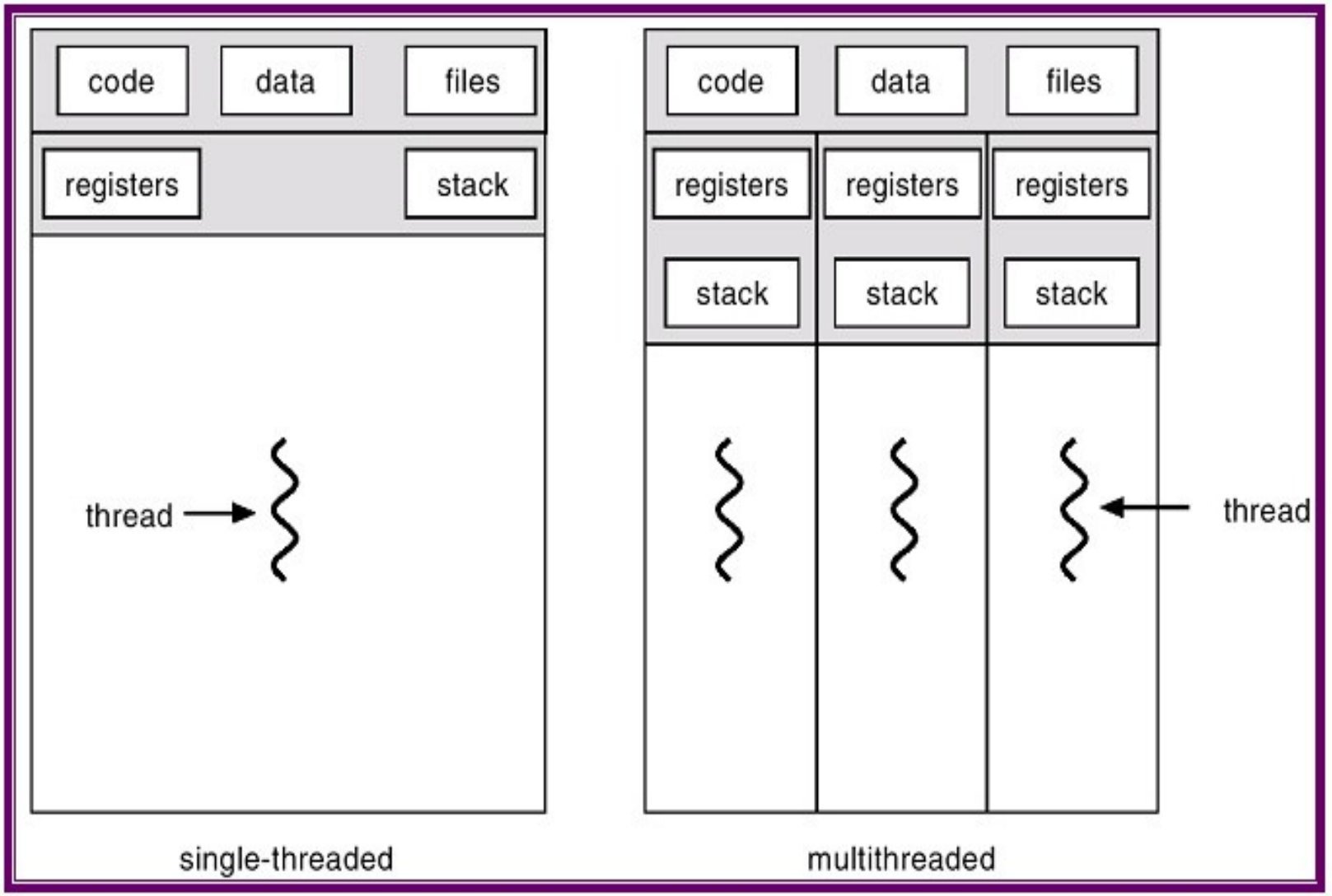
Потоки

- Создание нового потока
- Завершение потока
- Ожидание потока
- Информация о потоке

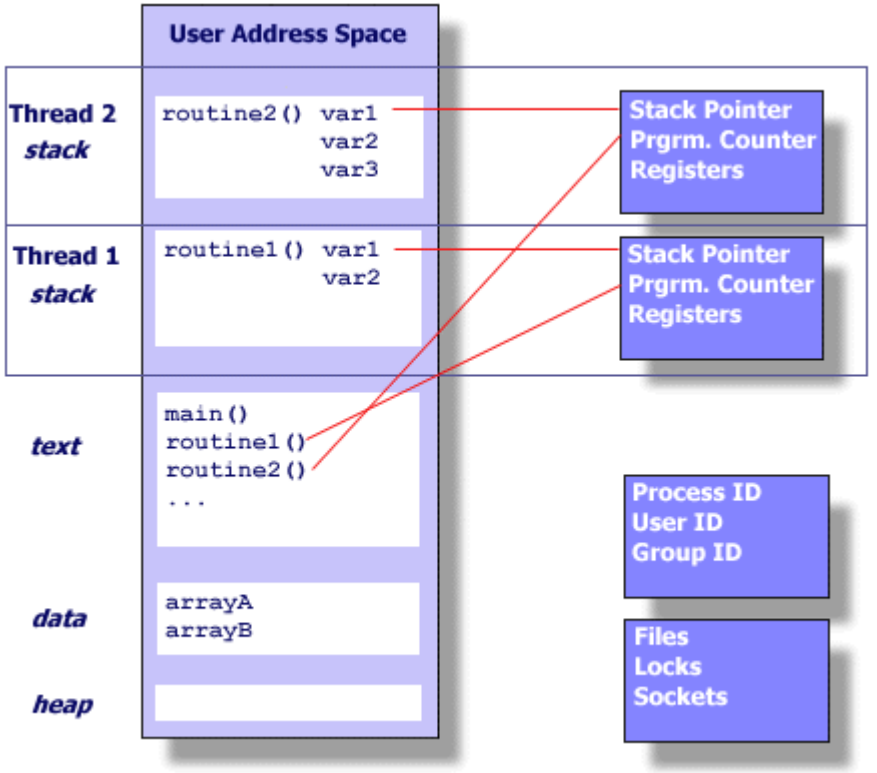
Многопоточные приложения



POSIX Threads Programming:
https://computing.llnl.gov/tutorials/pthreads/

POSIX Threads (pthreads)

execution model that exists independently from a language



1. #include <pthread.h>

2. link option: -lpthread

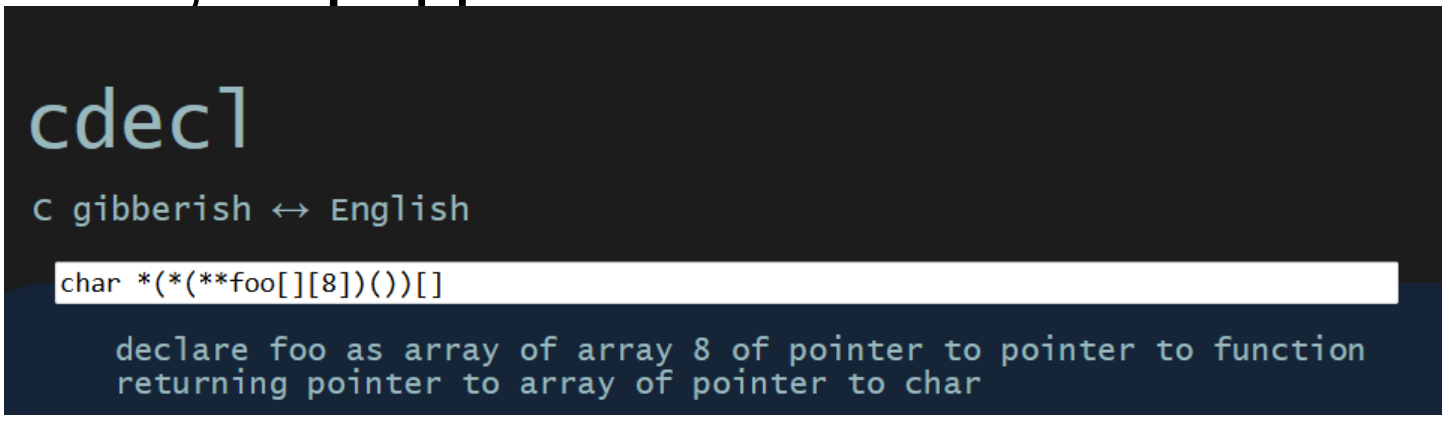
gcc -o Task main.c func.c -lm -lpthread

Создание потока

```
int pthread_create(pthread_t * THREAD_ID,
                   void * ATTR,
                   void *(*THREAD_FUNC)(void*),
                   void * ARG);
```

THREAD_ID - идентификатор нового потока
ATTR - атрибуты потока (пока NULL)
PTHREAD_FUNC - указатель на потоковую функцию
ARG — аргумент, передаваемый потоковой функции.

cdecl
https://cdecl.org/



Завершение потока

0. Вызов функции exit, _exit или _Exit из любого потока завершает весь процесс (со всеми Потоками)
1. Завершить работу потока без завершения всего процесса — завершение потоковой функции (**return**)
2. Поток можно принудительно завершить другим потоком того же самого процесса
3. Поток может вызвать функцию:

```
#include <pthread.h>
void pthread_exit(void * RESULT);
```

Ожидание завершения потока

```
#include <pthread.h>

int pthread_join(pthread_t THREAD_ID,
                 void ** DATA);
```

0 — успешный вызов; != 0 - ошибка

Блокирует вызывающий поток, пока не завершится поток с идентификатором THREAD_ID.
По адресу DATA помещаются данные, возвращаемые потоком функцией pthread_exit() или инструкцией return потоковой функции.

Получение информации о потоке

```
#include <pthread.h>

pthread_t pthread_self(void);
int pthread_equal(pthread_t THREAD1,
                  pthread_t THREAD2);
```

result = 0 — THREAD1 и THREAD2 являются идентификаторами разных потоков
result = 1 — идентификаторы THREAD1 и THREAD2 относятся к одному потоку