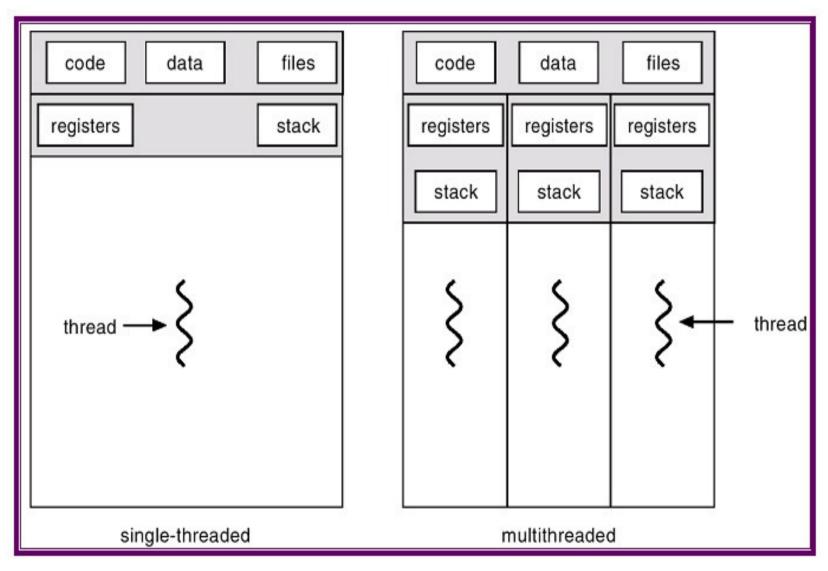
Потоки

- →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

Thread 2

Thread 1

stack

text

data

heap

stack

User Address Space

routine1() var1

var2

var3

routine2()

main()
routinel()

arrayA

arrayB

routine2()

Stack Pointer

Stack Pointer

Prarm. Counter

Registers

Reaisters

Process ID

User ID Group ID

Files Locks

Sockets

Prgrm. Counter

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 — аргу<u>мент, передаваемый потоковой</u>
```

cdecl
https://cdecl.org/

функции.

cdecl

C gibberish ↔ English

char *(*(**foo[][8])())[]

declare foo as array of array 8 of pointer to pointer to function returning pointer to array of pointer to char

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
```