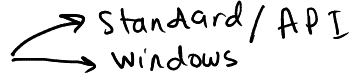


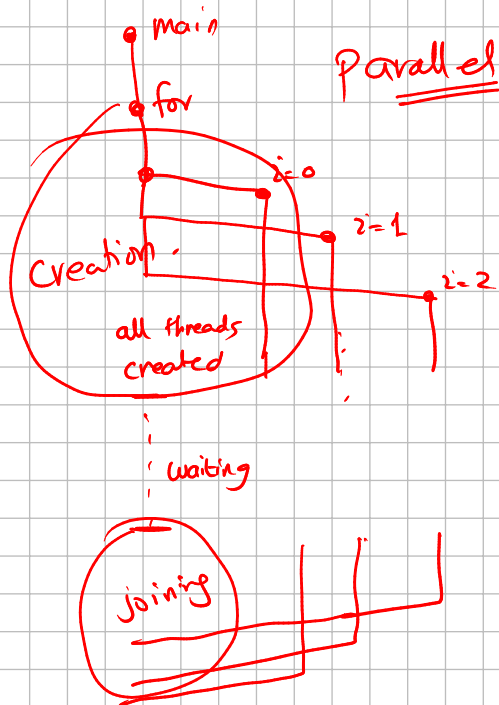
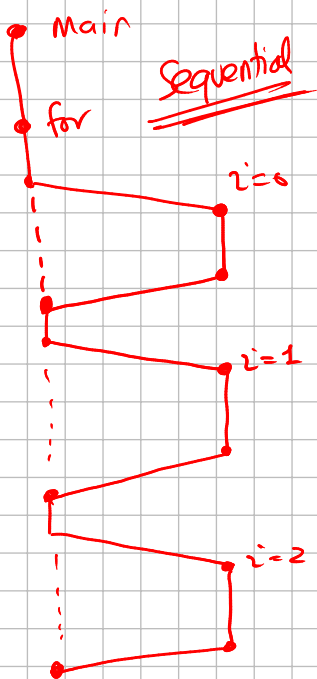
Posix Threads **pthread**s

- Before there was OpenMP, common approach to support parallel programming was(is) **pthread**s
- **P**ortable **O**perating **S**ystem **I**nterface for UNIX 
- Originally for UNIX and Linux, but meant for all operating systems that are POSIX standard compliant (Windows did not fall down this way)

```
#include <pthread.h>
#define NUM_THREADS 5

void *PrintHello(void *threadid) {
    printf("\n%d: Hello World!\n", threadid);
    pthread_exit(NULL);
}

int main() {
    pthread_t threads[NUM_THREADS];
    int rc, t;
    for(t=0; t < NUM_THREADS; t++) {
        printf("Creating thread %d\n", t);
        rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
        if (rc) printf("ERROR; return code from pthread_create() is %d\n", rc);
    }
    pthread_exit(NULL);
}
```



Process

↳ PID, UID, GID
Memory, Stack,
Heap.....

Thread

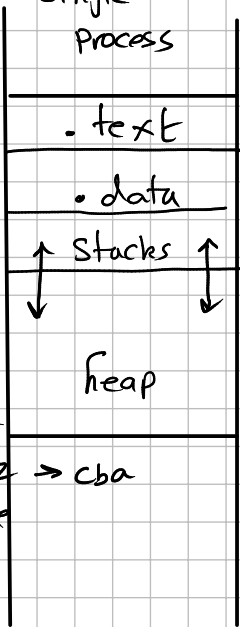
• Main() — abc()

↳ thread 1 create

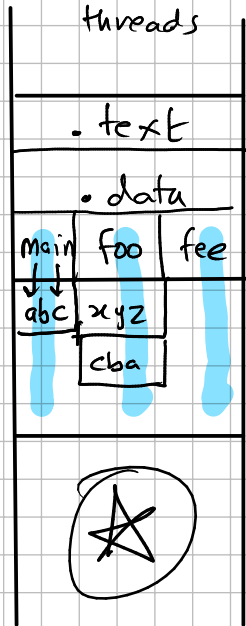
(foo) → xyz → cba

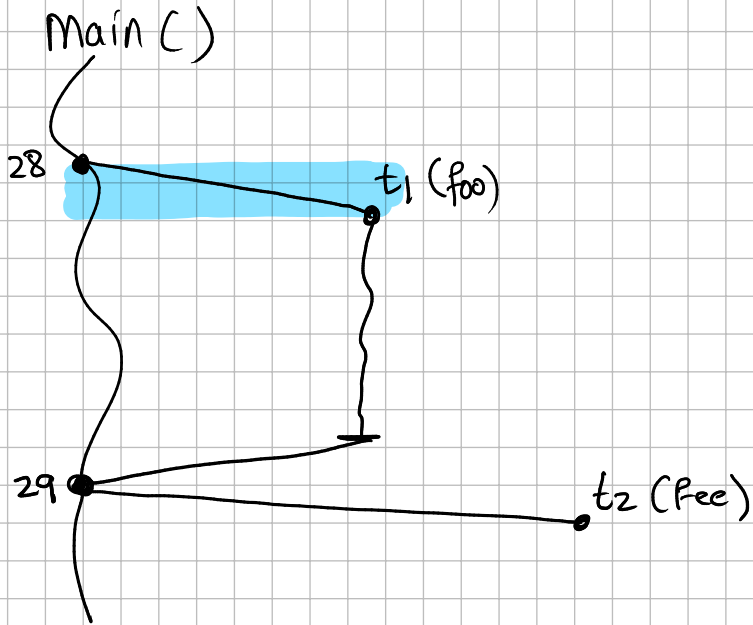
↳ thread 2 create
fee

Single Process



Multiple threads



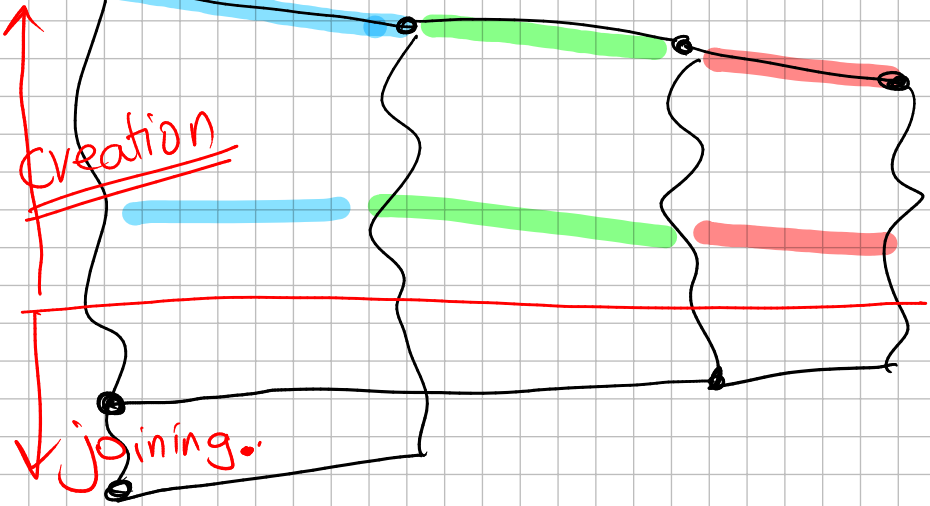


Main
main()
create t1

t1
foo
Create t2

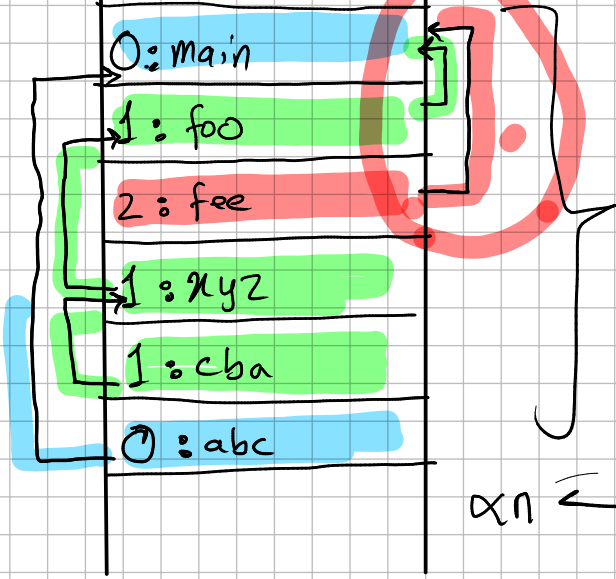
t2
foo
Create t3

t3



↓ Numbering scheme → tid

~~LIFO~~
per thread basis



single
stack

$\propto n$

- stack pointer
- registers
- scheduler (thread)
- signal

Posix Threads **pthread** (cont.)

- Compiled as

```
gcc filename.c -lpthread
```

- **Joining a Thread:** Making one thread wait for another (e.g., calling thread waiting for called thread)

```
void *foo() {
    printf("Hello Thread\n");
}

int main() {
    pthread_t tid;
    7 → pthread_create(&tid, NULL, foo, NULL);
    8 → pthread_join(tid, NULL); → wait
    9 → printf("Hello Process\n");
    exit(0);
}
```



- Many programmers find posix to be hard, cumbersome
 - Function pointers
 - Cryptic functions calls such as:


```
pthread_create(), pthread_exit(), pthread_join()
```
- Low chances that a compiler may optimize automatically for the above code
- Code is dependent on Posix compatible platforms (operating systems) only.
- Not designed for data-parallelism (scientific computing)

