

운영체제론 실습 5주차

정보보호연구실 @ 한양대학교

운영체제론 실습 5주차

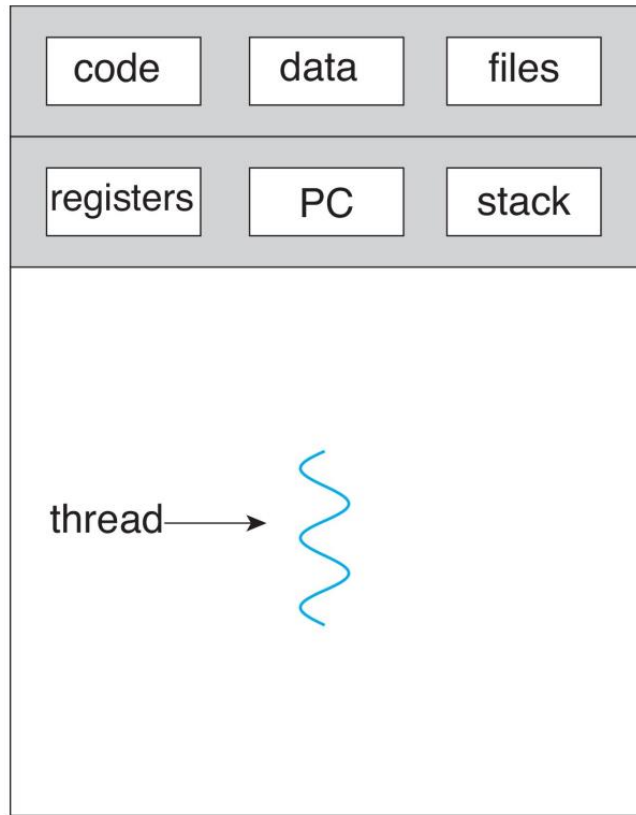
1. 스레드 (Thread)
2. OpenMP
3. 소수 탐색 프로그램 (count_primes)

스레드

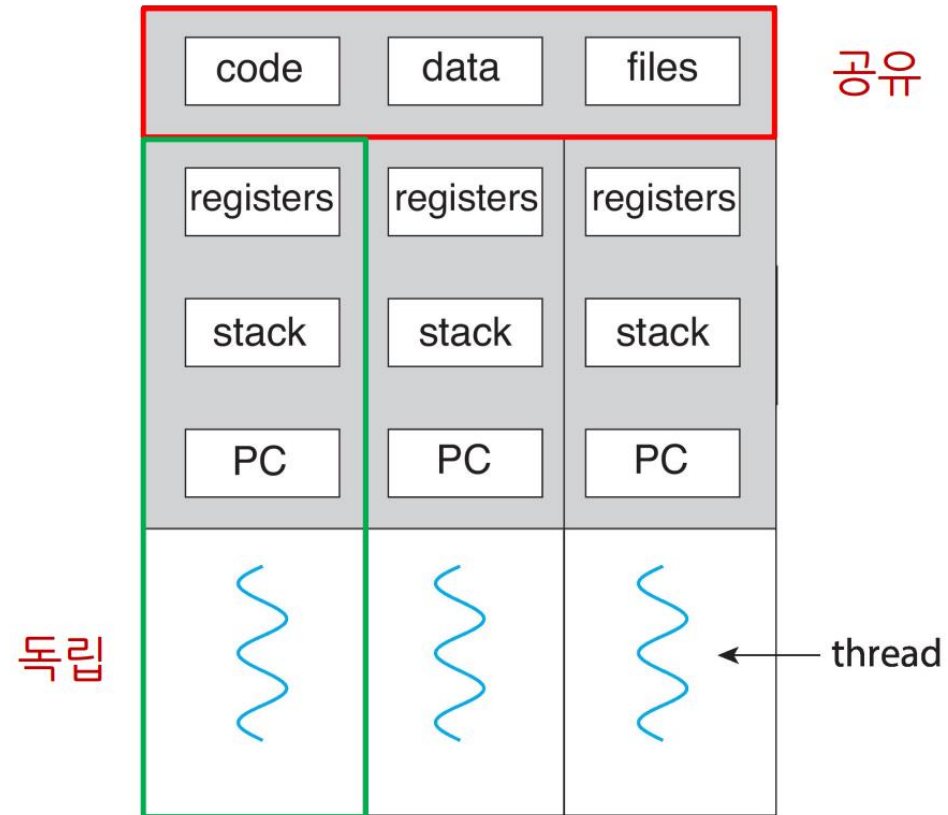
프로세스 안에서 실질적으로 작업을 실행하는 주체

- 프로세스마다 반드시 한 개 이상의 스레드가 존재
- 두 개 이상의 스레드일 경우는 multi-thread
- 프로세스의 생성은 heavy-weight, 스레드의 생성은 light-weight
- 프로세스의 자원을 공유
- Context switching에서 더 적은 overhead를 가짐
- 멀티코어 구조를 이용할 수 있음

스레드



single-threaded process



multithreaded process

#include <pthread.h>

- pthread_create
 - 스레드의 생성
 - 스레드 식별자, 스레드 특성, 실행할 함수, 실행할 함수의 매개변수
 - 스레드 특성은 주로 NULL
 - 실행 시, main 스레드 외의 작업 스레드가 생성
- pthread_join
 - 스레드의 종료를 대기, 종료하면 포인터를 리턴
 - 스레드 식별자, 리턴 값(보통 NULL)

#include <pthread.h>

- pthread_exit
 - 스레드의 종료 및 cleanup, destructor 루틴을 위해 사용
 - value
 - 주로 0

OpenMP

- Open Multi-Processing
 - 다중코어 환경에서 병렬프로그래밍을 지원하는 API
 - MacOS의 경우 사전 설치 필요 (LMS 참고)
 - Linux: \$ gcc -fopenmp *{filename}.c*
 - MacOS: \$ gcc -Xpreprocessor -fopenmp *{filename}.c* -lomp

코어의 수만큼 스레드를 생성하고
N개의 iteration을 스레드에 분할 매핑

```
#pragma omp parallel for  
for (i = 0; i < N; i++) {  
    c[i] = a[i] + b[i];  
}
```

실습: 소수 탐색 프로그램

TODO:

- 주어진 수가 소수인지 판단하는 `isPrime()`
- `isPrime`을 이용하여, 어떤 범위 내 존재하는 소수의 수 세기
- GROUP개의 스레드를 생성하고, 코어에 분산하여 할당
- OpenMP를 사용하여 코어의 개수만큼 스레드 생성 및 분산, 할당

실습 결과화면

- Count의 개수는 모두 동일해야 함
- 병렬 계산을 통해 작업 시간이 크게 단축됨을 확인

```
kevin@kevin-3090ti:~/Downloads$ ./a.out
1000000000부터 1040000000까지 소수를 찾는다.
순차 계산...
소수 개수: 1928678개
실행 시간: 124.0026초
----
병렬 계산 (다중 스레드)...
소수 개수 1928678개
실행 시간: 9.1293초
----
병렬 계산 (OpenMP)...
소수 개수 1928678개
실행 시간: 9.1209초
----
kevin@kevin-3090ti:~/Downloads$ -□
```