

디몬.

(=서비스).

• 상주하는 프로세스, 백그라운드 위치.

• 디몬의 임플리 x → 구별 할당 x.

• 독자 세션 가짐.

Thread.

• 프로세스는 너무 "무겁다" ... 오버헤드가 큼.

• ^(작은) 구별 할당 단위는 스레드가 됨.

• Context Switching: Lightweight하게 실행될 수 있도록.

• 스레드 vs 프로세스.

프로세스의 단점

• 프로세스 생성 오버헤드 ↑

• 컨텍스트 스위칭 오버헤드 ↑

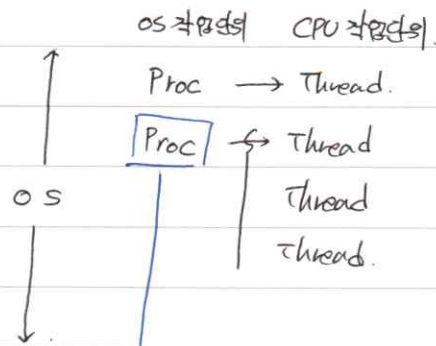
• 프로세스 간 통신 복잡 ↑.

• 하나의 작업을 여러 단위로 쪼개서 실행하면... 어렵다.

↳ 여러 플러시어 { 화면 ... 프로세스 기반 멀티태스킹?
 소리. ↳ 컨텍스트 스위칭: 부담.
 데이터베이스
 텍스트
 자막
 ...
 생활 복잡 ↑.

⇒ 스레드 등장 ... 프로세스의 Dispatch 단위: 스레드.
 관리 단위: 프로세스.

스레드



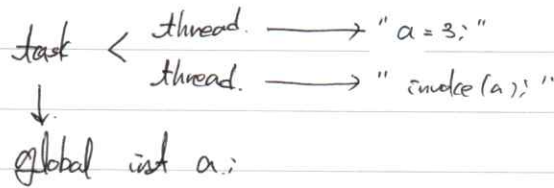
→ 가장 기본적인 프로세스 단위 ... 스레드 사이에 메모리 공유

→ 코드.데이터.첩. (But 스택, 힙은 각 스레드마다)

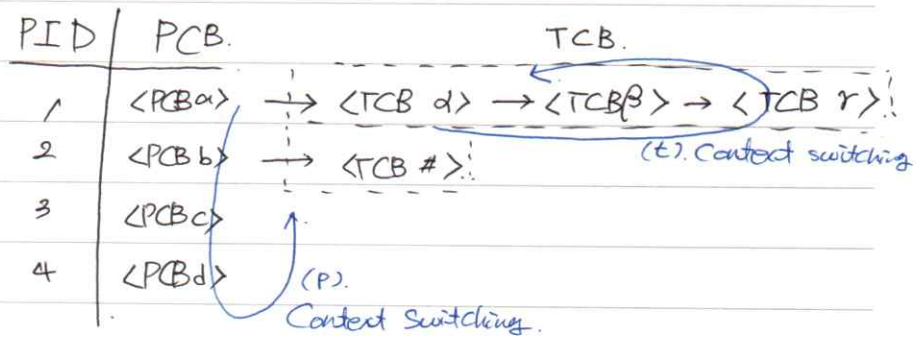
• 스레드 ... 함수로 생각해도 된다.

→ 모든 프로그램은 진입점 함수가 있다, 함수란 커 이상의 코드 블록.

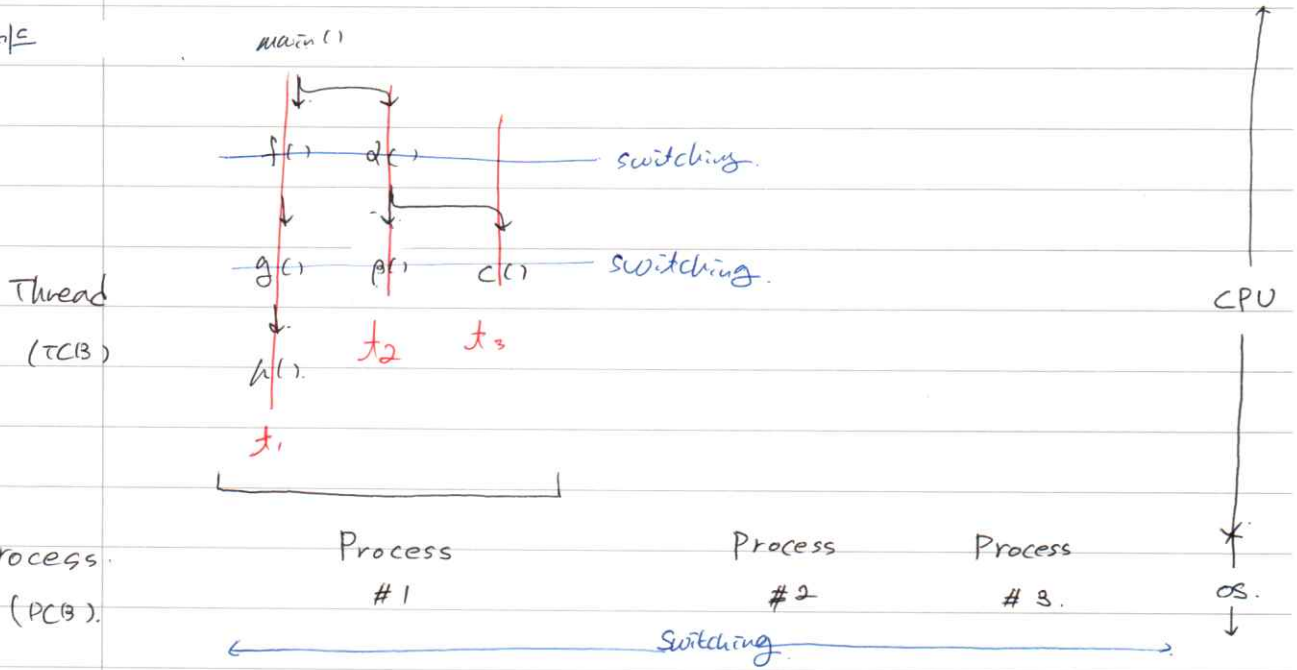
• 스레드 간 통신 개념도.



• TCB.



스레드



Life Cycle

· 프로세스 종료 \Rightarrow 스레드 종료.

· 어떤 스레드에서 오류 발생 \Rightarrow 다른 스레드 종료.
(프로세스 종료 \rightarrow 다른 스레드 종료)

구현

· `pthread_t` (struct).

· `--thread` TLS에 저장됨.

· `pthread_create` (function).

· `pthread_join` (function).

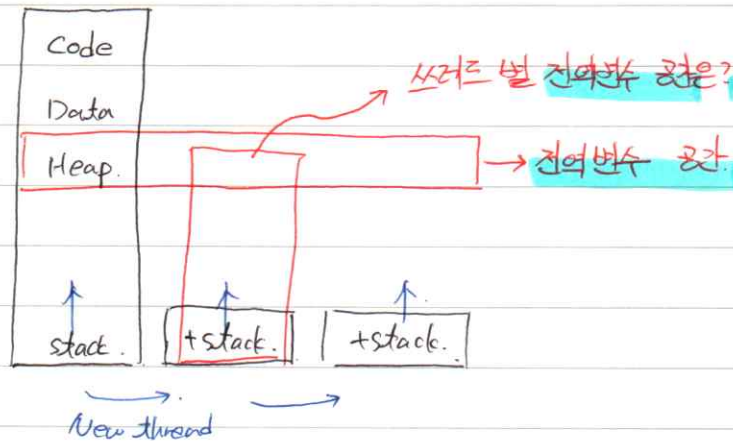
Note

· 스레드 실행 시 알 수 없음. "동시성"

· 공유 자원 동시에 사용 가능할 때, O 안됨.

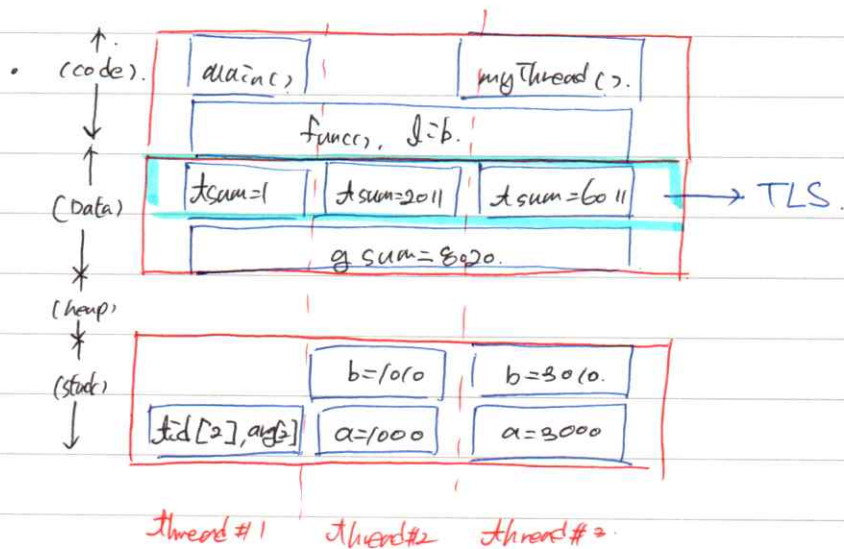
· 하나의 자원 여러개 사용? \Rightarrow 운영체제 문제.

쓰레드 구현.



• TLS ... Thread Local Storage.

쓰레드별 전역변수 공간.

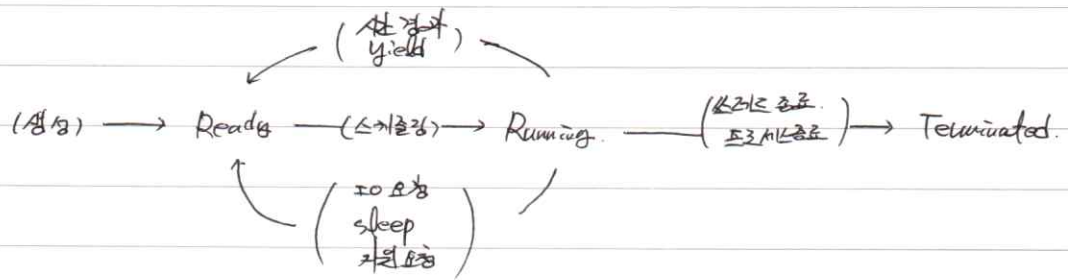


• int --thread tsum=1

↳ TLS 변수 선언.

↳ 이륙이 크리티컬 바운더리 할당 대상을 이미 여러 쓰레드 컨텍스트 상의 TLS 변수만
접근 되어, 수정 불가.

LifeCycle.



Operations.

- 생성 ... `syscall`, 라이브러리 기능으로 생성.
... 프로세스 생성시 `main` 스레드 생성.

- 종료 ≠ 프로세스 종료.

... 프로세스 종료 : `exit()`

= 메인 스레드 종료

= 모든 스레드 종료.

} 종료.

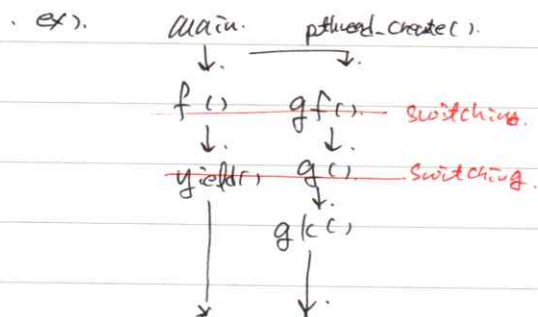
... 스레드 종료 : `pthread_exit()`. → 메모리를 청산... 메인 종료.

- 종료 "스레드 합류 ... 합류 위해 (종료시키기) 대기"

... `pthread_join()`.

- 합류 ... 자식을 만들 때 스레드 대기 우선.

... `yield()`.



Thread Context Switching.

(Page 93. Material 05.)

• \longleftrightarrow 프로세스 간 스위칭.

TLS 스택 위주 \longleftrightarrow 전체.

Overhead in context switching.

• High Cost Operation. ... 본래 할 일 못함.

• 동일 프로세스 내 스위칭

① 컨텍스트 저장 복제.

② TCB 리소스 조작.

③ 캐시 플러시, 재충전.

캐시 리드 ↑

• 프로세스 간.

① 메모리 바이트.

② 캐시 바이트. ... 캐시 복원 작업.

Type of thread.

• Kernel-level thread.

• syscall에 의해 생성

• Pure kernel-level thread

... 운영체제 관리 위함.

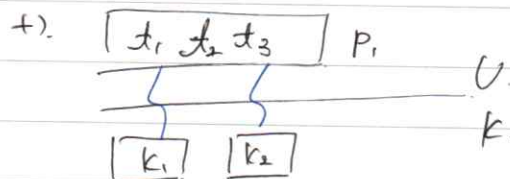
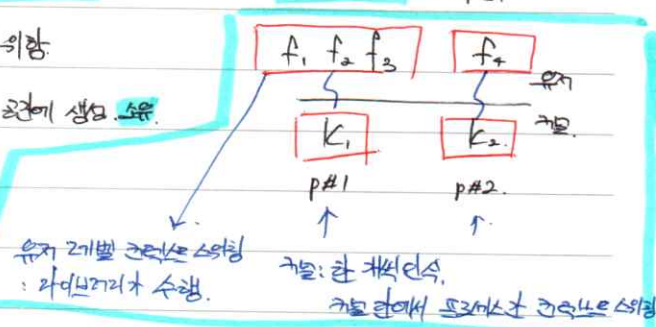
정확한, 조종에 용이.

• 각 프로세스 TCB를 커널 공간에 생성. 소유.

• User-level thread.

• 커널, User-level thread 인식 X.

단일한 프로세스 내.



↳ 커널에서 프로세스 객체 인식.

* Page 91, Material 03.

+) Simultaneous Multithreading. (SMT).

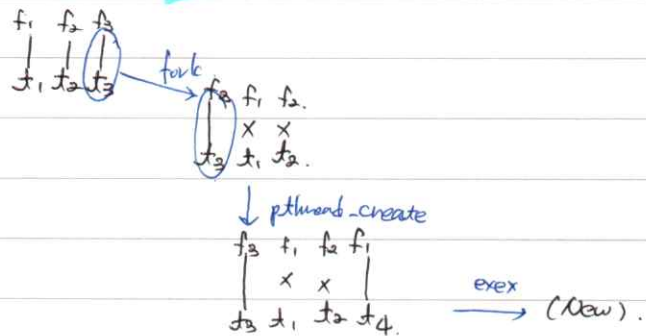
a.k.a. Hyper-threading.

Page 32, Material 05.

추가 이슈들.

· 멀티 프로세서에서 fork 콜 시?

· (구현마다 다름). 일반적으로, 개별 프로세서 복사. (컨텍스트에 대해).



· fork가 전체를 복제한다면?

몇몇은 그럴.

일반적인 사나리움... fork \rightarrow exec 로출라니 의미 없음 + 복잡함.· 자원 공유 문제 (\propto Thread-safe).

· Thread-safe. ... 여러 프로세서가 동시에 접근하는 이상 없음.

↓ 예.

· Re-entrancy. ... 다른 프로세서가 어떤 프로세서 실행 중인 함수 호출 시 결과가 추가 복잡함.

· Thread-local storage ... 프로세서의 동시 접근 차단.

· Mutual exclusion. ... 공유자원 사용 시 자원 접근, 락으로 통제.

· Atomic operations. ... "원자 연산", "원자적 구현"

#장점.

· 연속한 음절성 \longrightarrow I.e.) UI 단

· 자의 사용 효율성.

· 프로그램 구조 단순화.

· 작업 시간. 효율적 증가.