

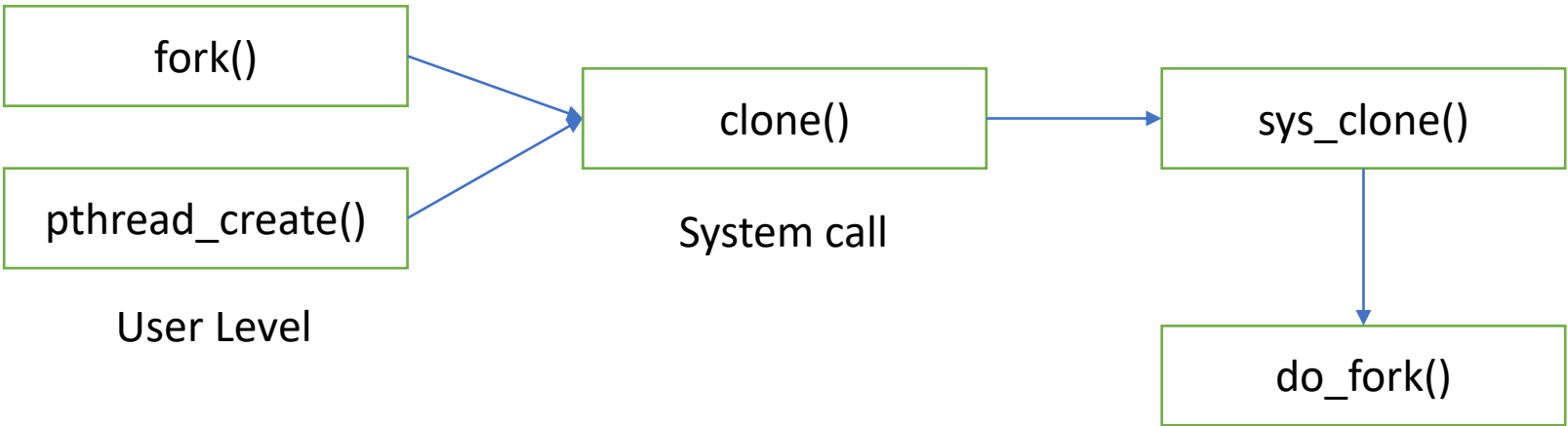
Part 04. 리눅스 커널 입문

Chapter 02. 태스크

진행 순서

Chapter 02_01	프로세스와 태스크
Chapter 02_02	task_struct
Chapter 02_03	상태 전이
Chapter 02_04	스케줄링

Chapter 02_01 프로세스와 태스크

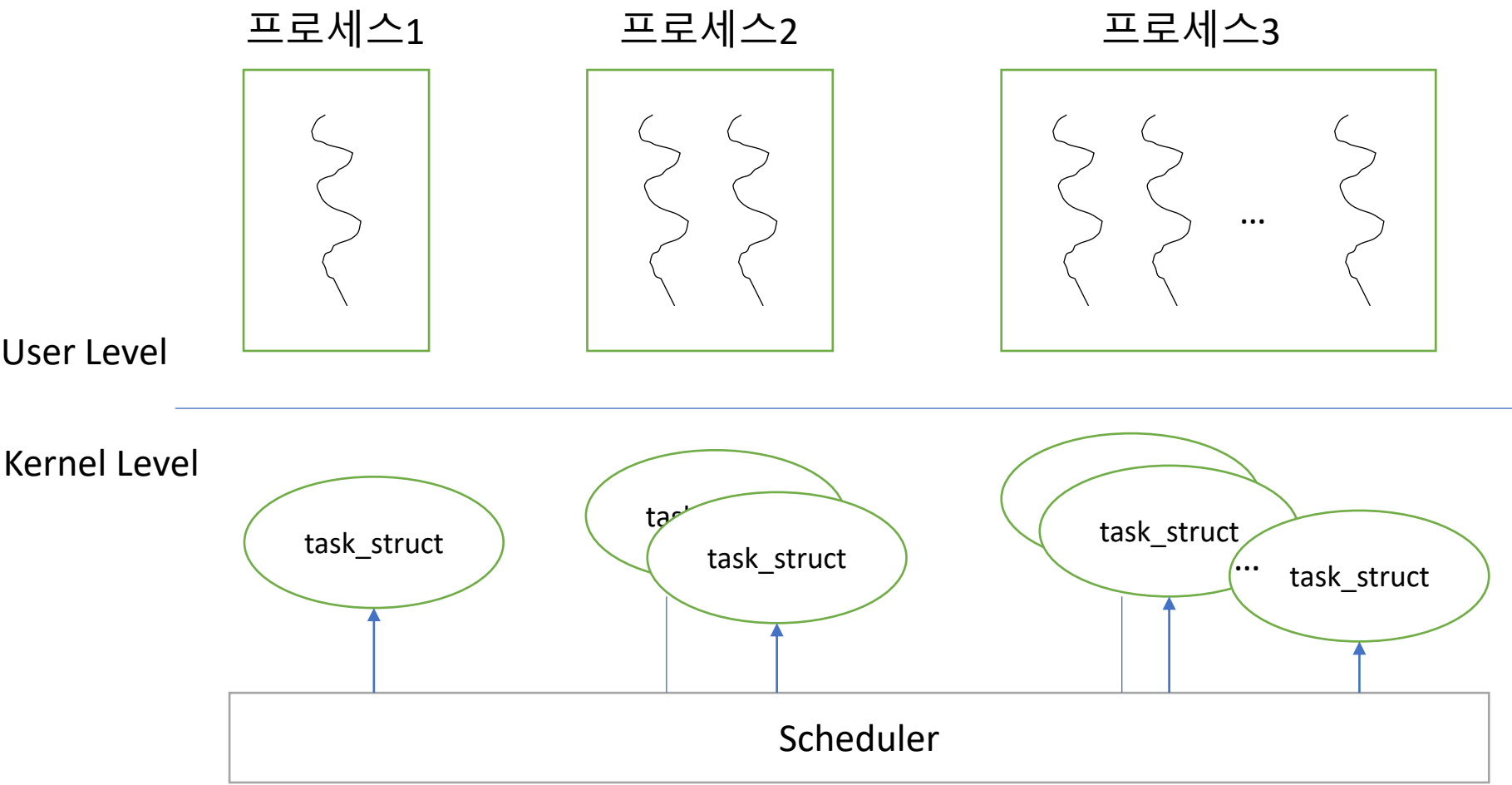


User Level

Kernel Level

task_struct 구조체 할당

Chapter 02_01 프로세스와 태스크



Chapter 02_02 task_struct

task_struct 자료구조 (/include/linux/sched.h - struct task_struct)

- task identification
 - 태스크를 인식하기 위한 변수
 - pid: 태스크 ID
 - tgid: 스레드 그룹 ID
 - pid를 통해 태스크의 task_struct를 찾기 위한 해시 관련 필드
 - 접근 권한 제어 필드: uid(사용자 ID), euid(유효 사용자 ID), suid(저장된 사용자 ID), fsuid(파일시스템 사용자 ID)
 - 사용자 그룹 접근 제어 필드: gid, egid, sgid, fsgid
- state
 - 태스크의 생성부터 소멸까지 상태 관리를 위한 변수
 - TASK_RUNNING(0)
 - TASK_INTERRUPTIBLE(1)
 - TASK_UNINTERRUPTIBLE(2)
 - TASK_STOPPED(4)
 - TASK_TRACED(8)
 - EXIT_DEAD(16)
 - EXIT_ZOMBIE(32)

Chapter 02_02 task_struct

task_struct 자료구조 (/include/linux/sched.h - struct task_struct)

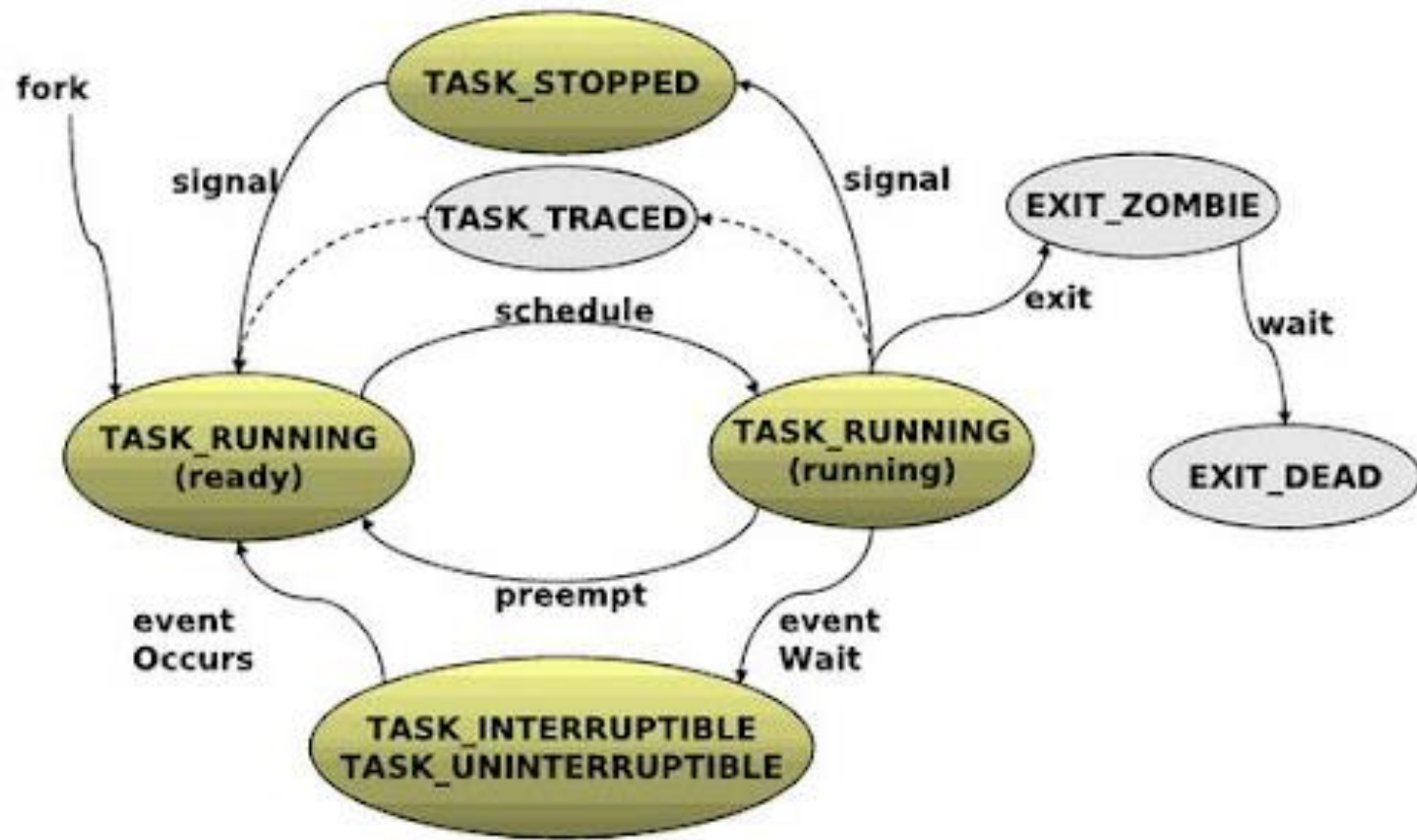
- task relationship
 - 태스크 가족 관계
 - real_parent: 현재 태스크를 생성한 부모 태스크의 task_struct 구조체를 가리킴
 - parent: 현재 부모 태스크의 task_struct 구조체를 가리킴
 - children: 자식 태스크의 리스트
 - sibling: 형제 태스크의 리스트
- scheduling information
 - 스케줄링 관련 변수
 - prio, policy, cpus_allowed, time_slice, rt_priority
- signal information
 - 시그널 관련 변수
 - signal, sighand, blocked, pending
- memory information
 - 명령어와 데이터를 text, data, stack, heap 공간에 저장
 - 이 공간에 대한 위치, 크기, 접근 제어 정보 등을 관리하는 변수
 - 가상주소를 물리주소로 변환하기 위한 디렉토리와 페이지 테이블 등의 주소 변환 정보
 - mm_struct

Chapter 02_02 task_struct

task_struct 자료구조 (/include/linux/sched.h - struct task_struct)

- file information
 - files_struct files: 태스크가 open한 파일 접근
 - fs_struct fs: 루트와 현재 디렉토리의 inode 접근
- thread struct
 - 스레드 구조 TSS(thread_struct)
 - 문맥 교환을 수행할 때 태스크가 현재 어디까지 수행되었는지 기억하는 공간
- time information
 - 태스크의 시간 정보를 위한 변수
 - start_time, real_start_time
 - 태스크가 사용한 CPU 시간 통계
- format
 - 바이너리 실행 포맷 정보(linux, bsd, svr4 포맷 실행 정보)
- resource limits
 - 태스크가 사용할 수 있는 자원의 한계
 - rlim_max: 최대 허용 자원의 수
 - rlim_cur: 현재 설정된 허용 자원의 수

Chapter 02_03 상태 전이



Chapter 02_04 스케줄링

스케줄링: 대기 상태의 태스크 들 중 특정 태스크를 선택하여 CPU 자원에 할당하는 과정

우선순위 140 단계

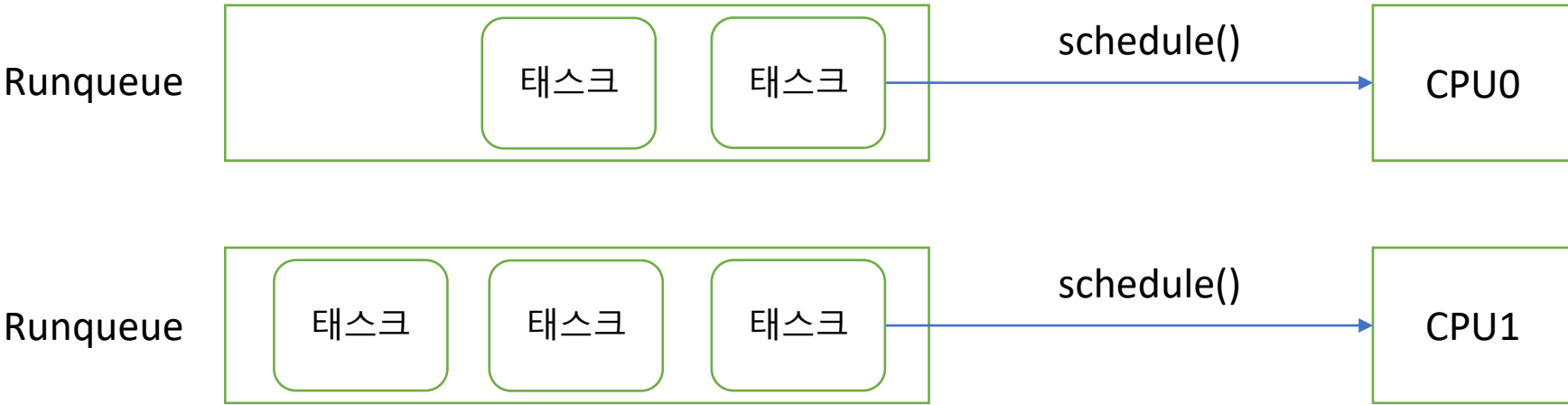
실시간 태스크: 0 ~ 99 단계

일반 태스크: 100 ~ 139 단계

런 큐(Runqueue)

수행 가능한 상태의 태스크를 자료구조로 관리 (kernel/sched/sched.h – struct rq)

CPU 별 하나의 런 큐가 유지됨



Chapter 02_04 스케줄링

실시간 태스크 스케줄링

- SCHED_FIFO: 타임 슬라이스 기반, 동일한 우선순위가 없는 경우의 RR
- SCHED_RR: 타임 슬라이스 기반, 동일한 우선순위 경우 타임 슬라이스 다 쓸 때까지
- SCHED_DEADLINE: deadline이 가장 가까운(가장 급한) 태스크를 스케줄링 대상으로 선정
- 우선순위 task_struct 구조체의 rt_priority 필드 (0 ~ 99)

일반 태스크 스케줄링

- SCHED_NORMAL
- SCHED_IDLE: 중요도 낮은 일을 수행하는 태스크가 CPU 점유하는 것을 막기 위해 가장 낮은 우선순위로 스케줄링
- SCHED_BATCH: 사용자와 상호 작용이 없는 CPU 중심의 일괄 작업(batch job) 태스크를 위한

```
struct rq {
```

```
...
```

```
struct rt_rq rt;
```

```
struct dl_rq dl;
```

```
struct cfs_rq cfs;
```

```
...
```

```
}
```

Runqueue

FIFO & RR

struct rt_rq

DEADLINE

struct dl_rq

CFS

struct cfs_rq

Chapter 02_05 참고

참고 서적

- 리눅스 커널 심층 분석(Linux Kernel Development), 로버트 러브, 에이콘 출판사
- 리눅스 커널 내부구조, 백승제, 최종무, 아티오 출판사