

목표 : 기본 스케줄러 변경(CFS → FIFO)

```
int sched_fork(unsigned long clone_flags, struct task_struct *p)
{
    unsigned long flags;

    __sched_fork(clone_flags, p);
    /*
     * We mark the process as NEW here. This guarantees that
     * nobody will actually run it, and a signal or other external
     * event cannot wake it up and insert it on the runqueue either.
     */
    p->state = TASK_NEW;
```

/usr/src/linux/linux-5.11.22/kernel/sched/core.c 파일의 sched_fork() 함수 내부를 수정

기본 스케줄러를 CFS가 아닌 FIFO를 사용하도록 수정

목표 : 기본 스케줄러 변경(CFS → FIFO)

```
static int __sched_setscheduler(struct task_struct *p,
                                const struct sched_attr *attr,
                                bool user, bool pi)
{
    int newprio = dl_policy(attr->sched_policy) ? MAX_DL_PRIO - 1 :
                MAX_RT_PRIO - 1 - attr->sched_priority;
    int retval, oldprio, oldpolicy = -1, queued, running;
    int new_effective_prio, policy = attr->sched_policy;
    const struct sched_class *prev_class;
    struct callback_head *head;
    struct rq_flags rf;
    int reset_on_fork;
    int queue_flags = DEQUEUE_SAVE | DEQUEUE_MOVE | DEQUEUE_NOCLOCK;
    struct rq *rq;
```

프로세스의 스케줄링 정책이나 우선순위를 변경하는 /usr/src/linux/linux-5.11.22/kernel/sched/core.c 파일의 __sched_setscheduler() 함수를 수정

스케줄링 정책, 우선순위 값 등의 정보를 담은 sched_attr 구조체를 이용하여 프로세서에 적용되는 스케줄러를 FIFO로 변경하도록 수정