

# Chapter 9: Signal (4/5)

김선영

[sunzero@gmail\(dot\)com](mailto:sunzero@gmail.com)

버 전: 2017-05-27

# Signal handler on MTM



# signal handling : MTM

## ❖ Multi-threaded model

- ▶ 전통적인 방법은 사용할 수가 없다.
- ▶ 어떤 스레드가 수행하는 함수에 **EINTR**이 걸릴지 판단 불가
  - ↳ 짧은 시간에 복수의 시그널이 도착한다면?  
= 서로 다른 스레드에 도착하는 최악의 상황도 발생!

# Signal on MT

## ❖ Threading 환경에서의 signal 처리

- ▶ signal handler를 담당할 thread 설정
- ▶ 그 외의 모든 thread는 signal block

```
int pthread_sigmask(int how, const sigset_t *newmask, sigset_t *oldmask);  
int pthread_kill(pthread_t thread, int signo);  
int sigwait(const sigset_t *set, int *sig);
```

# Signal on MT (con't)

## ❖ pthread\_signal.c

```
#define MAX_THREAD 5
void *start_sigthread(void *);      /* signal thread */
void *start_thread(void *);
struct thread_arg {
    pthread_t  pt_id;
    void *(*func)();
} thr_arg[MAX_THREAD] = {
    {0, start_sigthread},          /* 시그널 처리 쓰레드 */
    {0, start_thread},             /* 일반 쓰레드 */
    {0, NULL}
};
void clean_thread(struct thread_arg *t_arg);
```

# Signal on MT (con't)

```
int main() {
    int    i;
    sigset_t  sigset_mask, sigset_oldmask;
    sigfillset(&sigset_mask);
    sigdelset(&sigset_mask, SIGINT);
    pthread_sigmask(SIG_SETMASK, &sigset_mask, &sigset_oldmask);
    printf("[MAIN] Process PID = %d\n", getpid());
    for(i=0; i<MAX_THREAD && thr_arg[i].func != NULL; i++) {
        if (pthread_create(&thr_arg[i].pt_id,
                        NULL, thr_arg[i].func, (void *)&thr_arg[i])) {
            fprintf(stdout, "[MAIN] FAIL: pthread_create()\n");
            return 0;
        }
        printf("[MAIN] Create thread : tid = %lu\n", thr_arg[i].pt_id);
    }
    clean_thread(thr_arg);
    return 0;
}
```

# Signal on MT (con't)

```
void *start_sigthread(void *arg) {
    struct thread_arg *t_arg = (struct thread_arg *)arg;
    sigset_t    sigset_mask;    int    signum;
    sigemptyset(&sigset_mask);
    sigaddset(&sigset_mask, SIGUSR1); sigaddset(&sigset_mask, SIGTERM);
    while (1) {
        if (sigwait(&sigset_mask, &signum))
            fprintf(stdout, "[SIGNAL] FAIL: sigwait()\n");
        switch(signum) {
            case SIGUSR1:
                printf("[SIGNAL] Received SIGUSR1\n"); break;
            case SIGTERM:
                printf("[SIGNAL] Received SIGTERM\n"); exit(EXIT_SUCCESS);
            default:
                fprintf(stdout, "[SIGNAL] unknown (%d)\n", signum); break;
        }
    }
    return t_arg;
}
```

# Signal on MT (con't)

```
void *start_thread(void *arg)
{
    struct thread_arg *t_arg = (struct thread_arg *)arg;
    int i;
    for(i=0;;i++) {
        printf("\t[THREAD:%lu] i = %d\n", t_arg->pt_id, i);    sleep(2);
    }
    return t_arg;
}

void clean_thread(struct thread_arg *t_arg)
{
    int i;
    struct thread_arg *t_arg_ret;
    for (i=0; i<MAX_THREAD; i++, t_arg++) {
        pthread_join(t_arg->pt_id, (void **)&t_arg_ret);
        printf("[Join] Thread (No:%d) (ID:%lu)]\n", t_arg->idx, t_arg->pt_id);
    }
    printf("[Clean] All thread are joined.\n");
}
```

# MT의 signal handling은?

- ❖ 시그널 매스킹과 처리를 주의해야 한다.
  - ▶ 특히 `sigprocmask`를 쓰면... X
    - multi-thread 환경에서는 `sigprocmask`를 사용하면 오류가 발생한다.  
= core dump가 발생할 수 있다.