Signal 처리

Signal

- <signal.h>에 정의되어 있음
  - SIGINT, SIGABRT, SIGALRM, …
  - 반드시 찾아 확인할 것
- 정상 종료와 비정상 종료
  - 대부분 시그널이 수신되면 정상 종료됨
  - SIGABRT, SIGBUS, SIGSEGV, SIGQUIT 등은 비정상 종료됨 –core 파일을 만들고 종료됨

Signal 처리

- 디폴트 행동
  - 정상 종료 : SIGINT, …
  - 비정상 종료 : SIGQUIT, SIGABRT, …
  - 무시 : SIGUSR1, SIGUSR2
- 무시하고 수행을 계속
- 사용자가 정의한 행동

Signal 집합

- 사용법

```
#include <signal.h>
int sigemptyset(sigset_t *set);
int sigfillset(sigset_t *set);
int sigaddset(sigset_t *set, int signo);
int sigdelset(sigset_t *set, int signo);
```

- 사용 예

```
sigset_t mask1, mask2;

sigemptyset(&mask1);

sigaddset(&mask1, SIGINT);
sigaddset(&mask1, SIGQUIT);

sigfillset(&mask2);

sigdelset(&mask2, SIGCHILD);
```

Signal 행동 지정

- 사용법

```
#include <signal.h>
int sigaction(int signo, const struct sigaction *act, struct sigaction *oact);
```

- 구조체

```
struct sigaction {
        void (*sa_handler)(int);                    /* 취해질 행동 */
        sigset_t sa_mask;                           /* 시그널 처리할 동안 추가 시그널 봉쇄 */
        int sa_flags;                               /* 시그널 형태에 영향을 미칠 플래그들 */
        void (*sa_sigaction)(int, siginfo_t *, void *);
 };
```

  - sa_handler
    - SIG_DFL : 시스템의 디폴트 행동
    - SIG_IGN : 시그널 무시 (SIGSTOP, SIGKILL에는 쓸 수 없음)
    - 함수 주소
  - sa_mask
    - 여기에 지정된 시그널들은 지정된 함수가 수행되는 동안 봉쇄됨
    - 무시되는 것은 아니며, 함수가 수행된 후 해당 시그널 처리(단 중복된 시그널을 구별할 수는 없음)
  - sa_flag
    - SA_RESETHAND : 복귀 후 SIG_DFL로 재설정
    - SA_SIGINFO : 핸들러에 추가 정보 전달

Sample Programs

- sig_sample1.c
  - SIGINT 포착하여 사용자 함수 수행
- sig_sample2.c
  - SIGINT 무시
  - SIGINT 원래 행동으로 복귀
- sig_sample3.c
  - 부모 자식간 SIGUSR1 보내기

Signal 봉쇄

- 사용법

```
#include <signal.h>
int sigprocmask(int how, const sigset_t *set, sigset_t *oset);
```

- 예

```
sigset_t set1;


...
sigfillset(&set1);
sigprocmask(SIG_SETMASK, &set1, NULL);
/* 중요한 코드 수행 */
sigprocmask(SIG_UNBLOCK, &set1, NULL);


...
```