9주차 1차시 시그널의 개념

【학습목표】

1. 시그널의 기본 개념을 설명할 수 있다.
2. 시그널 발생 및 처리 방법을 설명할 수 있다.

학습내용1 : 시그널 개념

1. 시그널 이란?

① 개념
소프트웨어 인터럽트 프로세스에 뭔가 발생했음을 알리는 간단한 메시지를 비동기적으로 보내는 것

② 발생사유
0으로 나누기처럼 프로그램에서 예외적인 상황이 일어나는 경우
kill 함수처럼 시그널을 보낼 수 있는 함수를 사용해서 다른 프로세스에 시그널을 보내는 경우
사용자가 Ctrl+C와 같이 인터럽트 키를 입력한 경우

③ 시그널 처리방법
각 시그널에 지정된 기본 동작 수행. 대부분의 기본 동작은 프로세스 종료
시그널을 무시
시그널 처리를 위한 함수(시그널 핸들러)를 지정해놓고 시그널을 받으면 해당 함수 호출
시그널이 발생하지 않도록 블록처리

* 시그널의 종류

시그널	번호	기본 처리	발생 요건
SIGHUP	1	종료	행업으로 터미널과 연결이 끊어졌을 때 발생
SIGINT	2	종료	인터럽트로 사용자가 Ctrl+C를 입력하면 발생
SIGQUIT	3	코어 덤프	종료 신호로 사용자가 Ctrl+\를 입력하면 발생
SIGILL	4	코어 덤프	잘못된 명령 사용
SIGTRAP	5	코어 덤프	추적(trace)이나 중단점(breakpoint)에서 트랩 발생
SIGABRT	6	코어 덤프	abort 함수에 의해 발생
SIGEMT	7	코어 덤프	에뮬레이터 트랩으로 하드웨어에 문제가 있을 경우 발생
SIGFPE	8	코어 덤프	산술 연산 오류로 발생
SIGKILL	9	종료	강제 종료로 발생
SIGBUS	10	코어 덤프	버스 오류로 발생
SIGSEGV	11	코어 덤프	세그먼테이션 폴트로 발생
SIGSYS	12	코어 덤프	잘못된 시스템 호출로 발생
SIGPIPE	13	종료	잘못된 파이프 처리로 발생
SIGALRM	14	종료	알람에 의해 발생
SIGTERM	15	종료	소프트웨어적 종료로 발생
SIGUSR1	16	종료	사용자 정의 시그널 1

이외에도 시그널의 종류는 다양함 (표 7-7 참조)

학습내용2 : 시그널 발생 및 처리

1. 시그널 보내기 함수

프로그램에서 시그널 보낼때 사용하는 함수
프로세스 종료시 kill 함수 사용

기능	함수원형

시그널 보내기	int kill(pid_t pid, int sig);
	int raise(int sig);
	void abort(void);
	int sigsend(idtype_t idtype, id_t id, itn sig);

2. 시그널 핸들러 지정 함수

시그널 핸들러란?
시그널을 받았을 때 이를 처리하기 위해 지정된 함수
프로세스를 종료하기 전에 처리할 것이 있거나, 특정 시그널에 대해 종료하고 싶지 않을 경우 지정

기능	함수원형

시그널 핸들러 지정	void (*signal (int sig, void (*disp)(int)))(int);
	void (*sigset(int sig, void (*disp)(int)))(int);
	int sigignore(int sig);

3. 시그널 집합 관련 함수

시그널 집합(signal set) : 여러 개의 시그널 처리(POSIX 표준)

기능	함수원형

시그널 집합	int sigemptyset(sigset_t *set);
	int sigfillset(sigset_t *set);
	int sigaddset(sigset_t *set, int signo);
	int sigdelset(sigset_t *set, int signo);
	int sigsmember(sigset_t *set, int signo);

4. 시그널 처리 제어 함수

* sigaction() 함수
시그널을 받아 처리할 시그널 지정 및 플래그 설정해 시그널 처리 과정 제어 가능
시그널 핸들러가 수행되는 동안 다른 시그널 블록할 수 있음

기능	함수원형
시그널 제어	int sigaction(int sig, const struct sigacction *restrict act, struct sigaction *restrict oact);

5. 알람 시그널 관련 함수

기능	함수원형
알람 시그널	unsigned int alarm(unsigned int sec);
인터벌 타이머	int gettimer(int which, struct itimerval *value); int settimer(int whicch, const struct itimerval *value, struct itimerval *ovalue);

6. 기타 시그널 관련 함수

일정한 시간이 지난 후에 자동으로 시그널 발생.
한번 혹은 일정 시간 간격의 주기적 발생

기능	함수원형
시그널 정보 출력	void psignal(int sig, const char *s) char *strsignal(int sig);

시그널 블록과 해제	int sighold(int sig);
	int sigrelse(int sig);
	int sigprecmask(int how, const sigset_t *restrict set, sigset_t *restrict oset);

시그널 대기	int sigpause(int sig);
	int sigsuspend(const sigset_t *set)
	int pause(void);

【학습정리】

1. 시그널 : 소프트웨어 인터럽트 프로세스에 뭔가 발생했음을 알리는 간단한 메시지를 비동기적으로 보내는 것

2.

기능	함수원형

시그널 보내기	int kill(pid_t pid, int sig);
	int raise(int sig);
	void abort(void);
	int sigsend(idtype_t idtype, id_t id, itn sig);

기능	함수원형

시그널 핸들러 지정	void (*signal (int sig, void (*disp)(int)))(int);
	void (*sigset(int sig, void (*disp)(int)))(int);
	int sigignore(int sig);

기능	함수원형

시그널 집합	int sigemptyset(sigset_t *set);
	int sigfillset(sigset_t *set);
	int sigaddset(sigset_t *set, int signo);
	int sigdelset(sigset_t *set, int signo);
	int sigsmember(sigset_t *set, int signo);

3.

기능	함수원형
시그널 제어	int sigaction(int sig, const struct sigacction *restrict act, struct sigaction *restrict oact);

기능	함수원형
알람 시그널	unsigned int alarm(unsigned int sec);
인터벌 타이머	int gettimer(int which, struct itimerval *value); int settimer(int whicch, const struct itimerval *value, struct itimerval *ovalue);

기능	함수원형
시그널 정보 출력	void psignal(int sig, const char *s) char *strsignal(int sig);

시그널 블록과 해제	int sighold(int sig);
	int sigrelse(int sig);
	int sigprecmask(int how, const sigset_t *restrict set, sigset_t *restrict oset);

시그널 대기	int sigpause(int sig);
	int sigsuspend(const sigset_t *set)
	int pause(void);