

# 6장. 시그널과 시그널 처리

# 6.1 서론 (1)

- 프로세스
- 프로세스간 통신 (IPC : inter-process communication)
  - Signal
  - Pipe
  - FIFO
  - 고급 IPC 기법
    - 메시지 전달
    - 세마포어
    - 공유 메모리
  - 시그널
    - Ctrl+c

## 6.1 서론 (2)

- 6.1.1 시그널 이름
- 6.1.2 정상적 종료와 비정상적 종료
  - <sys/wait.h> 매크로들

## 6.2 시그널 처리 (1)

- 시그널 처리 방법
  - default 행동
  - 무시
  - 사용자가 정의한 행동
- 6.2.1 시그널 집합
  - sigset\_t과 관련 함수들
- 6.2.2 시그널 행동 지정 : sigaction
  - SIG\_DFL
  - SIG\_IGN
  - 함수 주소

## 6.2 시그널 처리 (2)

- 6.2.2 시그널 행동 지정 : sigaction (계속)
  - 예 1: SIGINT 포착
  - 예 2: SIGINT 무시
  - 예 3: 이전 행동 복원
  - 예 4: 우아한 퇴장
- 6.2.3 시그널과 시스템 호출
  - 시스템 호출 도중 시그널 도착시 행동
- 6.2.4 sigsetjmp와 siglongjmp
  - 예:

## 6.3 시그널 봉쇄(blocking)

- sigprocmask
  - SIG\_SETMASK
  - SIG\_UNBLOCK
  - SIG\_BLOCK

## 6.4 시그널 보내기

- 6.4.1 다른 프로세스에게 보내기 : kill
  - kill의 pid 인수
- 6.4.2 자신에게 시그널 보내기 : raise와 alarm
  - raise
  - alarm
    - 설정
    - 끄기
  - alarm 예
- 6.4.3 pause 시스템 호출
  - 임의의 시그널이 올 때까지 수행 중지
  - tml 예

# 실습과제

- 본문 예제 프로그램 실행 후 결과 확인(필요한 경우 main() 작성)
  - 6.2 예 1 [50점]
  - 6.2 예 2 [50점]
  - 6.2 예 3 [50점]
  - 6.2.4 예 [50점]
  - 6.3 예 [50점]
  - 6.4 예 [50점]
  - 6.4.2 예 [50점]
  - tml 예 [100점]
- 프로세스 간 시그널 처리 [500점]
  - 2개의 프로세스 생성
  - A 프로세스는 SIG\_USR1만을 받아 처리하고 나머지 시그널 무시
  - A 프로세스는 처음에 자신의 프로세스 번호 출력하고, 60초 후부터 화면에 # 반복 출력하다가 SIG\_USR1이 들어오면 화면에 시그널 도착 메시지 출력하고 종료
  - B 프로세스는 임의(random)의 시간 후 A 프로세스에게 SIG\_USR1 보내기
  - B 프로세스는 A 프로세스에게 SIG\_USR1은 반드시 보내고 종료