

UNIX 시스템 프로그래밍

» 5장. 프로세스 정보

Process

- ▶ process := 프로그램 코드, 변수 값, 스택 등에 저장된 값, PCB 내용 등을 포함;
 - Process vs. Program
- ▶ process의 계층 구조: parent process → child processes;
 - UNIX system의 모든 process는 init의 descendent process;

프로세스 목록 보기

- ▶ shell 상에서 프로세스 목록 확인
 - \$ps
- ▶ 실행 중인 프로세스 종료 시키기
 - \$kill -9 프로세스 번호

Process Identifier

- ▶ process identifier : 음이 아닌 정수
 - 0 : swapper
 - 1 : init
- ▶ getpid(), getppid() 시스템 호출

```
#include <unistd.h>  
pid_t getpid(void);  
pid_t getppid(void);
```

Process Group

▶ process group :

- 프로세스들을 묶어서 하나의 group으로 만든다.
- 같은 group에 속한 process들에게 동시에 signal을 보낼 수 있다.
- 초기에 fork나 exec에 의해 group id 계승

▶ group leader :

- 자신의 pid가 group id 이면, group의 leader 이다.

Process Group (2)

▶ group id 검색 시스템 호출 :

◦ 사용법 :

```
#include <sys/types.h>
```

```
#include <unistd.h>
```

```
pid_t getpgrp(void);
```

```
pid_t getpgid(pid_t pid);
```

➔ getpgid의 인자가 0이면 호출 프로세스 자신의 group id 검색

Process Group (3)

- ▶ process group의 변경 :
 - 사용법 :

```
#include <sys/types.h>
#include <unistd.h>
int setpgid(pid_t pid, pid_t pgid);
```
 - pid 인 프로세스의 group id를 pgid로 설정

Session

▶ session :

- 한 session은 한 단말기를 사용하는 foreground process group과 background process group의 집합체이다.
- 각 process group은 하나의 session에 속한다.

▶ getsid 시스템 호출 : session id를 획득

```
#include <sys/types.h>
#include <unistd.h>
pid_t getsid(pid_t pid);
```


Session (2)

▶ setsid() 시스템 호출 :

```
#include <sys/types.h>
#include <unistd.h>
pid_t setsid(void);
```

- 제어 단말기를 갖지 않는 새로운 session과 group 생성;
- 호출 프로세스의 id가 session과 group의 id가 된다.
- 만약, 호출 process가 현재 group의 leader 이면 -1을 return;

main() 함수 인자 사용

```
int main(int argc, char **argv){  
    int i;  
    printf("%d\n", argc);  
    i=0;  
    while(argv[i]!=NULL){  
        printf("%s\n", argv[i]);  
        i++;  
    }  
  
    return 0;  
}
```