

5주차 2차시 프로세스 식별

【학습목표】

1. PID의 개념을 설명할 수 있다.
2. PID를 함수를 활용하여 검색하고 관리할 수 있다.

학습내용1 : PID(Process ID) 개념

1. 프로세스란?

실행중인 프로그램을 프로세스(process)라고 부른다.
각 프로세스는 유일한 프로세스 번호 PID를 갖는다.
각 프로세스는 부모 프로세스에 의해 생성된다.

2. 프로세스 식별

① PID 검색: getpid(2)

이 함수를 호출한 프로세스의 PID를 리턴

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

② PPID 검색 : getppid(2)

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

* 부모 프로세스의 PID를 리턴

```
# ps -ef | more
UID  PID  PPID  C   STIME  TTY  TIME  CMD
root   0     0    0   1월 30일 ?    175:28 sched
root   1     0    0   1월 30일 ?     0:02 /sbin/init
root   2     0    0   1월 30일 ?     0:00 pageout
.....
```

부모 프로세스ID

* getpid, getppid 함수 사용하기

```
01 #include <unistd.h>
02 #include <stdio.h>
03
04 int main(void) {
05     printf("PID : %d\n", (int)getpid());
06     printf("PPID : %d\n", (int)getppid());
07
08     return 0;
09 }
```

678 프로세스는
콘솔

```
# ex5_1.out
PID : 2205
PPID : 678
```

```
# ps
PID TTY          TIME CMD
678 pts/3        0:00 ksh
2206 pts/3        0:00 ps
```

3. 프로세스 그룹

① 프로세스 그룹이란

관련 있는 프로세스를 묶은 것으로 프로세스 그룹 ID(PGID)가 부여됨

작업제어 기능을 제공하는 C셸이나 콘셸은 명령을 파이프로 연결하여 프로세스 그룹 생성 가능

② 프로세스 그룹 리더

프로세스 그룹을 구성하는 프로세스 중 하나가 그룹 리더가 됨

프로세스 그룹 리더의 PID가 PGID

프로세스 그룹 리더는 변경 가능

4. PGID 검색 : getpgrp(2), getpgid(2)

```
#include <unistd.h>

pid_t getpgrp(void);
pid_t getpgid(pid_t pid);
```

pid : PGID를 구하려는 프로세스의 ID

5. PGID 변경: setpgid(2)

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

int setpgid(pid_t pid, pid_t pgid);
```

pid : 프로세스 그룹에 속한 프로세서의 ID

pgid : 새로 지정할 PGID

① getpgrp, getpgid 함수 사용하기

```
01 #include <unistd.h>
02 #include <stdio.h>
03
04 int main(void) {
05     printf("PID : %d\n", (int)getpid());
06     printf("PGRP : %d\n", (int)getpgrp());
07     printf("PGID(0) : %d\n", (int)getpgid(0));
08     printf("PGID(2287) : %d\n", (int)getpgid(2287));
09
10     return 0;
11 }
```

```
# ex5_2.out
PID : 2297
PGRP : 2297
PGID(0) : 2297
PGID(2287) : 2285
```

실행방법
2287은 sleep의 PID

```
$ ps -ef | more | sleep 300 &
$ ps
PID TTY          TIME CMD
2278 pts/6        0:00 ksh
2301 pts/6        0:00 ps
2287 pts/6        0:00 sleep
```

학습내용2 : PID 관련 함수

1. 프로세스 목록 보기

① 현재 실행중인 프로세스 목록을 보려면 ps 명령 사용

```
# ps
  PID TTY          TIME CMD
  678 pts/3        0:00 ksh
 1766 pts/3        0:00 ps
```

② ps aux

항목	의미
UID	프로세스를 실행시킨 사용자 ID
PID	프로세스 번호
PPID	부모 프로세스 번호
C	프로세스의 우선순의
STIME	프로세스의 시작 시간
TTY	명령어가 시작된 터미널
TIME	프로세스에 사용된 CPU 시간
CMD	실행되고 있는 명령어(프로그램) 이름

③ 전체 프로세스를 보려면 -ef 옵션 사용

```
# ps -ef | more
  UID  PID  PPID  C   STIME  TTY  TIME  CMD
  root   0    0    0   1월 30일 ?    175:28 sched
  root   1    0    0   1월 30일 ?     0:02 /sbin/init
  root   2    0    0   1월 30일 ?     0:00 pageout
  .....

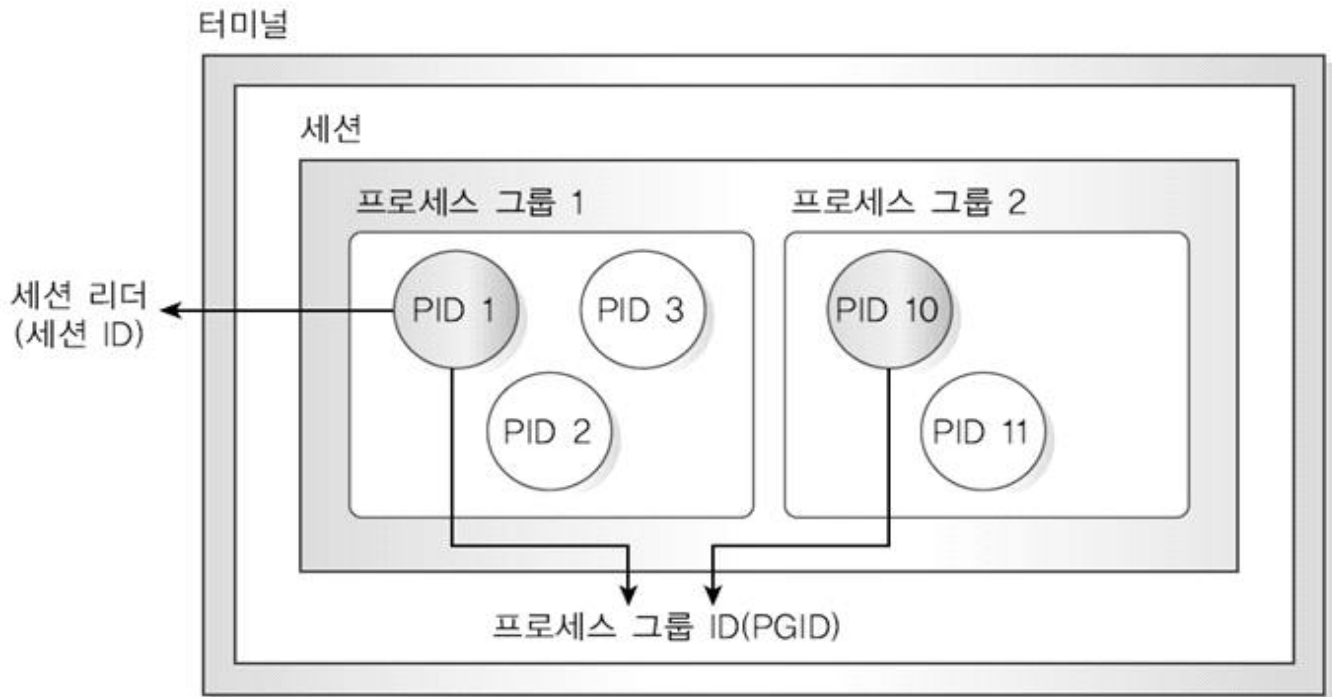
```

2. 세션

① 세션이란?

POSIX 표준에서 제안한 개념

사용자가 로그인해 작업하고 있는 터미널 단위로 프로세스 그룹을 묶은 것



② 세션 검색: getsid(2)

세션ID는 SVR4에서 정의한 개념

새로운 세션을 생성하면 해당 프로세스는 세션 리더가 되면 세션 리더의 PID가 세션ID

```
#include <unistd.h>

pid_t getsid(pid_t pid);
```

pid : 자신이 속한 세션의 ID 구하려는 프로세스의 ID

③ 세션 생성: setsid(2)

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

pid_t setsid(void);
```

getsid 함수 사용하기

```

01 #include <unistd.h>
02 #include <stdio.h>
03
04 int main(void) {
05     printf("PID : %d\n", (int)getpid());
06     printf("PGID : %d\n", (int)getpgrp());
07     printf("SID : %d\n", (int)getsid(0));
08
09     return 0;
10 }

```

```

# ex5_3.out
PID : 2584
PGID : 2584
SID : 678
# ps
PID TTY          TIME CMD
678 pts/3        0:01 ksh
2585 pts/3        0:00 ps

```

【학습정리】

1. 프로세스 식별 함수

기능	함수원형
PID 검색	pid_t getpid(void);
부로 PID 검색	pid_t getppid(void);
프로세스 그룹 ID 검색	pid_t getpgrp(void); pid_t getpgid(pid_t pid);
프로세스 그룹 ID 변경	int setpgid(pid_t pid, pid_t pgid);
세션 리더 ID 검색	pid_t getsid(pid_t pid);
세션 생성	pid_t setsid(void);