

PID

PID(Process ID)는 운영체제에서 프로세스를 식별하기 위해 프로세스에 부여하는 번호를 의미한다.

PPID

PPID(Parent Process ID)는 부모 프로세스의 PID를 의미한다.

만일 부모 프로세스가 자식 프로세스보다 일찍 종료되는 경우 자식 프로세스는 고아 프로세스가 되어 PPID로 `init` process의 PID(1)를 가지게 된다.

PGID

프로세스 그룹(Process Group)은 1개 이상의 프로세스의 묶음을 의미한다.

PGID(Process Group ID)는 이러한 프로세스 그룹을 식별하기 위해 부여되는 번호이다.

SID

세션(Session)은 1개 이상의 프로세스 그룹의 묶음을 의미한다.

SID(Session ID)는 이러한 세션을 식별하기 위해 부여되는 번호이다.

리눅스에서의 PID, PPID, PGID, SID

리눅스 운영체제에서 PID, PPID, PGID, SID를 확인하기 위해서는 `ps` 명령어를 사용하면 된다.

실행 중인 모든 프로세스의 PID, PPID, PGID, SID, COMMAND를 확인하기 위해 다음 명령어를 실행한다.

```
ps -A -o pid,ppid,pgid,sid,command
```

PLAINTEXT

위 명령어를 실행한 결과는 아래와 같다.(Docker, Ubuntu 16.04)

```
root@4a83f3110780:/# ps -A -o pid,ppid,pgid,sid,command
PID  PPID  PGID  SID  COMMAND
  1    0    1    1   /bin/bash
 29    1   29    1   ps -A -o pid,ppid,pgid,sid,command
```

PLAINTEXT

부모 프로세스와 자식 프로세스의 관계 이해하기

① 1 ~ 100 까지 순회하며 “Hello World”를 출력하는 `child.sh` 와 ② 이를 실행하는 `parent.sh` 스크립트가 존재한다.

child.sh

```
#!/bin/bash

for x in `seq 1 100`;
do
    sleep 1
    echo "Hello World"
done
```

parent.sh

```
#!/bin/bash

./child.sh
```

다음 명령어를 통해 `parent.sh`를 백그라운드로 실행하고, `ps -A -o pid,ppid,pgid,sid,command` 명령어를 통해 프로세스 목록을 확인해본다.

```
root@4a83f3110780:/# ps -A -o pid,ppid,pgid,sid,command
PID  PPID  PGID  SID  COMMAND
502   1    502   1    /bin/bash ./parent.sh 2
503   502   502   1    /bin/bash /child.sh
```

`child.sh`의 PPID가 자신을 실행한 `parent.sh`의 PID인 것을 확인할 수 있다.

또한 `parent.sh`의 PGID와 `child.sh`의 PGID가 502로 동일한 Process Group에 속해있는 것을 알 수 있다.

PGID의 존재 이유

위의 상태에서 `kill 502` 명령어를 이용하여 `parent.sh` 프로세스를 종료한 뒤, `ps -A -o pid,ppid,pgid,sid,command` 명령어를 통해 프로세스 목록을 확인해본다.

```
root@4a83f3110780:/# ps -A -o pid,ppid,pgid,sid,command
PID  PPID  PGID  SID  COMMAND
503   1     502   1    /bin/bash /child.sh
```

PLAINTEXT

`parent.sh` 가 실행한 `child.sh` 는 아직도 실행되고 있는 것을 확인할 수 있다.

`child.sh` 가 계속 실행되고 있는 이유는 `kill` 명령어의 Usage를 확인해보면 알 수 있다.

```
root@4a83f3110780:/# kill
kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]
```

PLAINTEXT

`kill` 명령어는 PID를 기준으로 실행되며, 특정 Process에 Signal을 보낸다.

여기서 주의해야 할 점은 **Signal**을 수신한 **Process**는 자기 자신이 종료될 뿐 자식 프로세스에까지 **Signal**을 전달하지 않는다는 것이다.

이러한 문제점을 해결하기 위해 POSIX 기반 운영체제에서는 여러 프로세스를 묶어서 프로세스 그룹을 만들고, 프로세스 그룹에 PGID를 부여하여 한번에 프로세스를 종료할 수 있도록 한다.

PGID로 프로세스 종료하기

다시 부모 프로세스를 실행하고 `ps -A -o pid,ppid,pgid,sid,command` 명령어를 실행하여 프로세스 목록을 확인한다.

PLAINTEXT

```
root@4a83f3110780:/# ps -A -o pid,ppid,pgid,sid,command
PID  PPID  PGID  SID  COMMAND
827   1    827   1    /bin/bash ./parent.sh 2
828   827   827   1    /bin/bash /child.sh
833   828   827   1    sleep 1
834   1    834   1    ps -A -o pid,ppid,pgid,sid,command
```

`kill -- -$PGID` 명령어를 통해 Process Group에 속한 모든 Process들을 종료할 수 있다.

PLAINTEXT

```
root@4a83f3110780:/# kill -- -827
```

위 명령어를 수행한 뒤 실행 중인 프로세스를 확인하면 `parent.sh` 와 `child.sh` 모두 종료된 것을 확인할 수 있다.

PLAINTEXT

```
root@4a83f3110780:/# ps -A -o pid,ppid,pgid,sid,command
PID  PPID  PGID  SID  COMMAND
928   1    928   1    ps -A -o pid,ppid,pgid,sid,command
```