

## Chapter 3. Process

getpid : return process id.

getppid : return parent process id.

getgid : return group id.

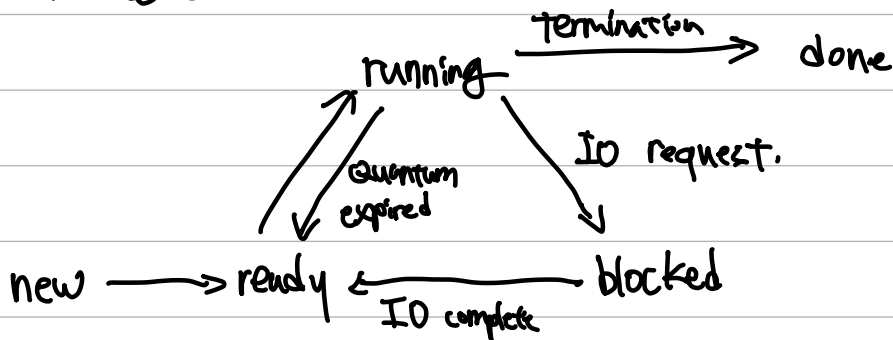
getegid : return effective group id. 실행권속지: real

getuid : return user id.

권한유지: effective

geteuid: return effective user id.

### ★ Process state



### fork

현재 프로세스를 복사하여 자식프로세스 생성

- return 0 to 자식
- return child pid to 부모
- 실패시 return -1 to 부모

자식과 부모 차이점.

- pid, ppid 가 다름.
- fork 반환값이 다름.
- CPU usage
- Locks and Alarm

wait : 자식 중 하나라도 종료되면 반환

waitpid ( pid, start\_loc, options ) : 특정 프로세스 지정 가능, 옵션 가능.

⋮

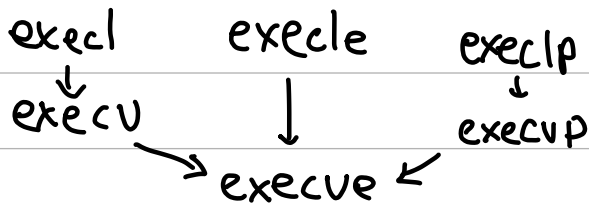
WNOHANG : 즉시 반환

모든 자식 기다려옴...

```
while ( r = wait ( NULL ) > 0 );
```

## exec

변환하여 조금씩은 execve 가 표준임.



이전과 같은 프로세스는 env를 상속받는다 (execl, execlp 제외)

exec ( path, arg0, arg1, ..., argn, NULL )

execlp ( path-or-file, arg0, arg1, ..., argn, NULL )

execl ( path, arg0, arg1, ..., argn, NULL, env[1] )

execv ( path, argv[1] )

execvp ( path-or-file, argv[1] )

execve ( path, argv[1], env[1] )

execcmd ls -l \*.c 를 치면 와일드카드를 해석해서,  
argv는 exccmd ls -l a.c b.c 가 된다.

## exit

exit (status)

atexit ( )

↑  
void function( void ) 에 작성!

최대 32개 까지 stack 으로 관리됨.

## Interrupt, Background

Ctrl-C ; 백그라운드에서 동작하고 있는 프로세스에는 정말 X

커맨드 맨 뒤에 & 붙이면 백그라운드 프로세스로 실행됨.

Daemon : 백그라운드 프로세스인데 중단안하고 계속 실행할것. ex) OS