

copyright 김성우

# LINUX

프로세스 / 부트로더

## 프로세스 (Process)

- **프로세스**: HDD에 저장된 실행 코드(프로그램)가 메모리에 로딩되어 활성화 된 것
- **포그라운드 프로세스(Foreground Process)**: 화면에 나타나 사용자와 상호 작용되는 프로세스를 의미하며, 화면에서 실행되는 것이 보이는 프로세스
- **백그라운드 프로세스(Background Process)**: 실행은 되었지만 화면에 나타나지 않고 뒤에서 실행되는 프로세스
- **프로세스 번호(PID)**: 각각의 프로세스에 할당된 고유 번호. 메모리에서 프로세스를 제거할 때 프로세스 번호를 사용함
- **작업 번호**: 현재 실행되는 백그라운드 프로세스의 순차 번호
  
- **부모 프로세스/자식 프로세스**: 모든 프로세스는 독립적이지 않고 부모 프로세스의 하위에 종속되서 실행됨  
부모 프로세스를 종료하면 자식 프로세스도 모두 종료됨
- **데몬 프로세스(Demon process)**: 특정 서비스를 제공하기 위해 존재하는 프로세스로, 리눅스 커널에 의해 실행됨
- **고아 프로세스(Orphan process)**: 자식 프로세스가 종료되기 전에 부모 프로세스가 먼저 종료되는 것을 의미.  
1번 프로세스가 새로운 부모 프로세스가 되어 고아 프로세스의 작업을 처리
- **좀비 프로세스(Zombie process / Defunct process)**: 프로세스가 정지된 상태지만, 그 정보가 삭제되지 않은 것(프로세스 목록에 남은 경우)을 의미.

데몬과 서비스

- 데몬(Daemon): 서버 프로세스를 지칭하는 말로, 리눅스에서의 서비스를 의미
- 서비스(Service): 웹 서버, 네임 서버, DB 서버 등의 '프로세스'를 지칭하는 말로, 평상시에도 늘 가동하는 프로세스
- 소켓(Socket): 필요할 때만 작동하는 서버 프로세스
- 서비스 = 데몬 = 서버 프로세스
- 서비스와 소켓은 서비스 매니저 프로그램인 'systemd'로 작동시키고 관리
- 'systemd'는 리눅스 시스템이 시작될 때, 가장 먼저 실행되어 다른 프로세스를 실행하는 역할을 함

서비스

- 시스템과 독자적으로 구동되어 제공하는 프로세스
- 서비스의 실행 스크립트 파일: /lib/systemd/system/서비스명.service
- enable/disabled 확인: #systemctl list -unit -files

#systemctl [start/stop/restart] [서비스명]

옵션/명령어	의미
status [서비스명]	서비스 상태 확인
[enable/disable] [서비스명]	서비스 사용/사용안함 설정
is-enabled [서비스명]	서비스 활성화 여부 확인
is-active [서비스명]	현재 서비스 동작 여부 확인

상태(STATE)	의미
static	다른 서비스/소켓에 의존된 형태로 enable/disable 설정불가

/ect/init.d와 systemd

- **/etc/init.d**: System V init tools (SysVinit)가 사용하는 스크립트들이 저장된 디렉터리로, 시스템이 시작될 때 PID가 1번인 init프로세스가 해당 디렉터리의 파일들을 실행하게 됨. (run-level의 스크립트들을 순차 실행. 속도가 느림)  
해당 스크립트들은 사용자가 직접 start/stop/restart/reload 가능
- **systemd(system daemon)**: 기존의 init 역할을 대체하는 데몬으로서, 우분투에서는 init 대신 systemd가 작동하게 됨  
따라서 PID 1번을 가지며, 부팅, 서비스, 로그 등을 관리함 (병렬로 실행되기 때문에 부팅 속도가 빠름)
- **/lib/systemd/system/**: systemd 데몬이 실행될 때 실행할 서비스 파일들이 존재하는 디렉터리

데몬명	설명
acpid	(Advanced configuration and power interface) 전원관리 데몬
apmd	(Advanced power Management) 노트북이나 배터리를 필요로 하는 장치에 사용되는 전원관리 데몬
autofs	파일 시스템을 사용하고 있을 때는 자동으로 마운트가 되기 하고, fs를 이용하고 있지 않을때는 자동으로 언마운트를 시켜주는 데몬
crond	사용자가 지정한 프로그램을 특정 시간에 주기적으로 실행할 수 있도록 해주는 표준 Unix 데몬
cups	프린터를 관리하는 데몬 (프린트 시 동작하고 있어야함)
dm	디스플레이 관리자가 작동할 수 있게 해주는 데몬 (X Window 사용 시 필수)
httpd	웹 서버 시작 또는 종료 담당 데몬
iptables	패킷 필터링 방화벽 구축 담당 데몬
messagebus	시스템 이벤트 메시지 담당 데몬

/ect/init.d와 systemd

데몬명	설명
nfs	NFS 서버 담당 데몬
mysqld	MySQL 데이터베이스 담당 데몬
newtwork	설정된 네트워크 인터페이스가 부팅 시 작동할 수 있도록 해주는 네트워크 담당 데몬
netfs	삼바, NFS, NCP등을 마운트하거나 언마운트 할 때 필요한 데몬
kerneld	커널 모듈을 관리하는 데몬
numlock	시스템 시작 시 Num Lock을 활성화 시켜주는 데몬
shorewall	shorewall 방화벽 (iptables와 연동하여 동작)
smb	윈도우와 리눅스간의 자료 공유를 담당하는 삼바 서버 데몬
sound	사운드 시스템 담당 데몬
sshd	원격 셸 접속 프로토콜인 ssh 서버 데몬
syslog	시스템에 발생하는 메시지를 log파일로 저장하는 데몬
xinetd	수퍼 데몬
xorgconfig	그래픽 카드를 자동 검색하여 X Window 해상도를 설정해주는 데몬

## 데몬 동작 유형과 xinetd 수퍼데몬

- **xinetd (extended Internet daemon)**: 리눅스 시스템이 부팅될 때 시작되는 서비스 데몬들을 관장하는 수퍼데몬  
안드로이드, 임베디스 시스템 사용자 이외에는 사용되고 있지 않음
- **1) Standalone 모드**: 수퍼데몬에 의지하지 않고 단독으로 서버 데몬이 항상 백그라운드 모드로 작동되어 요청이 있는 즉시  
반응하도록 하는 방법. 속도는 빠르지만 프로세스가 항상 메모리상에서 작동하기때문에 시스템 자원을 사용함  
*부모 프로세스 작동 -> 요청 -> 자식 프로세스 생성 -> 요청 끝음 -> 자식 프로세스 kill -> 요청 대기*
- **2) xinetd 모드**: xinetd에 의해 서비스가 작동되며, 외부의 접속 요청이 있을 때 xinetd 데몬에 의해 동작함. 속도가  
느리다는 단점이 있지만 시스템 자원을 standalone보다 적게 차지함

### xinetd 설치

```
#apt-get install -y xinetd
```

## 소켓

- 외부에서 특정 서비스를 요청할 경우 실행되는 프로세스. systemd가 구동시킴
- 요청이 끝나면 종료되며, 대표적으로 telnet이 있음
- 소켓 관련 스크립트 파일: /lib/systemd/system/소켓명.socket

# 프로세스

# 프로세스

## 프로세스

### ps

- Process status. 현재 프로세스의 상태를 확인하는 명령어
- 옵션 생략 시 현재 터미널에서 실행된 프로세스 목록을 출력
- 일시 중지된 프로세스 뿐만 아니라 백그라운드로 실행중인 프로세스도 모두 출력

옵션/명령어	의미
-A	모든 프로세스에 대한 정보 출력. -e와 같음
-a	가장 많이 요청되는 프로세스에 대한 정보 출력. 터미널과 관련이 없는 프로세스는 보여주지 않음
-u	특정 사용자에게 대한 모든 프로세스 정보 (user)
-x	시스템에서 실행 중인 모든 프로세스 정보
-e	시스템에 실행중인 모든 프로세스 정보
-f	자세한 정보 (full option)
-p	지정 PID 프로세스 정보
-l	사용자가 실행한 프로세스 목록 확인

항목	의미
UID	UserID. 프로세스 소유자의 ID
PID	Process ID. 프로세스의 ID
PPID	Parent PID. 부모 프로세스의 ID
C	CPU 가동률
STIME	프로세스가 시작된 날짜와 시간
TTY	프로세스와 연결된 터미널
TIME	총 CPU 사용 시간
CMD	프로세스의 실행 명령
PRI	프로세스 실행 우선 순위



# 프로세스

# 프로세스

## 프로세스 상태

```
root@Server:~/바탕화면# ps -axj | head -1; ps -axj | grep -P "(ps|1723)" | tail -5
  PPID      PID      PGID      SID TTY          TPGID STAT   UID    TIME COMMAND
    1        1086    1086     1086 ?             -1  Ss      116    0:00 /usr/sbin/kerneloops
  1713       1723    1723     1723 pts/0        3578  Ss       0    0:00 bash
  1723       3578    3578     1723 pts/0        3578  R+       0    0:00 ps -axj
  1723       3579    3578     1723 pts/0        3578  S+       0    0:00 grep --color=auto -P (ps|1723)
  1723       3580    3578     1723 pts/0        3578  S+       0    0:00 tail -5
```

## ps의 필드 정보

- PPID: 부모 프로세스 PID
- PID: 프로세스의 ID
- PGID: 프로세스 그룹ID
- SID: 세션 ID
- TTY: 사용 터미널. (로컬 로그인시 'tty+터미널수', 원격 로그인시 'pts/번호'로 표시, '?'는 터미널과 무관함을 의미)
- TPGID: tty에 속한 프로세스 그룹ID
- UID: UserID
- COMMAND: 실행 명령어
- STAT:

코드분류	의미
D	IO 같이 중지(interrupt)시킬 수 없는 잠자고 있는(휴지) 프로세스 상태
R	현재 동작중이거나 동작할 수 있는 상태
S	잠자고(sleep) 있지만, 중지시킬 수 있는 상태
T	작업 제어 시그널로 정지되었거나 추적 중에 있는 프로세스 상태
X	완전히 죽어 있는 프로세스
Z	죽어 있는 좀비 프로세스

코드분류	의미
<	프로세스의 우선 순위가 높은 상태
N	프로세스의 우선 순위가 낮은 상태
L	실시간이나 기존 IO를 위해 메모리 안에 잠겨진 페이지를 가짐
s	세션 리더 프로세스(주도 프로세스)
l	멀티 스레드
+	포어그라운드(fg) 상태로 동작하는 프로세스



프로세스 죽이기 (kill)

- 프로세스 제어 시 kill 명령어를 사용하여 프로세스에 시그널(signal)을 보내어 시그널에 의해 프로세스 제어 가능
- 프로세스는 시그널을 받을 시 '시그널에 해당되는 동작 실행', '시그널 무시', '사용자 정의 함수 실행' 중 동작
- 시그널은 '비신뢰성', '대기하지 않음' 성질을 가짐

프로세스 종료: #kill [시그널번호 or 시그널] PID

시그널 종류 확인: #kill -l

root@Server:~/바탕화면# kill -l				
1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL	5) SIGTRAP
6) SIGABRT	7) SIGBUS	8) SIGFPE	9) SIGKILL	10) SIGUSR1
11) SIGSEGV	12) SIGUSR2	13) SIGPIPE	14) SIGALRM	15) SIGTERM
16) SIGSTKFLT	17) SIGCHLD	18) SIGCONT	19) SIGSTOP	20) SIGTSTP
21) SIGTTIN	22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
26) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO	30) SIGPWR
31) SIGSYS	34) SIGRTMIN	35) SIGRTMIN+1	36) SIGRTMIN+2	37) SIGRTMIN+3
38) SIGRTMIN+4	39) SIGRTMIN+5	40) SIGRTMIN+6	41) SIGRTMIN+7	42) SIGRTMIN+8
43) SIGRTMIN+9	44) SIGRTMIN+10	45) SIGRTMIN+11	46) SIGRTMIN+12	47) SIGRTMIN+13
48) SIGRTMIN+14	49) SIGRTMIN+15	50) SIGRTMAX-14	51) SIGRTMAX-13	52) SIGRTMAX-12
53) SIGRTMAX-11	54) SIGRTMAX-10	55) SIGRTMAX-9	56) SIGRTMAX-8	57) SIGRTMAX-7
58) SIGRTMAX-6	59) SIGRTMAX-5	60) SIGRTMAX-4	61) SIGRTMAX-3	62) SIGRTMAX-2
63) SIGRTMAX-1	64) SIGRTMAX			

시그널명	번호	의미
SIGHUP	1	프로세스를 다시 띄우기 (재시작)
SIGINT	2	프로세스 interrupt(중지) (Ctrl + c)
SIGQUIT	3	프로세스 실행 중지 (Ctrl + W)
SIGILL	4	illegal instruction. 잘못된 명령을 실행시 발생
SIGBUS	7	메모리 접근 에러
SIGKILL	9	프로세스를 강제로 종료 (무시 불가, 제어 불가)
SIGTERM	15	Termiante. 프로세스 정상 종료. kill의 기본 시그널
SIGCHLD	17	자식 프로세스가 stop/종료 되었을 때 부모에게 전달
SIGCONT	18	Continue. STOP에 의해 정지된 프로세스를 재실행
SIGSTOP	19	Stop. 프로세스 정지. SIGCONT로 재실행 가능
SIGTSTP	20	실행 정지 후 다시 실행을 계속하기 위해 대기시키는 시그널. 일시정지로, SIGCONT로 재실행 가능 (Ctrl + z)
SIGIO	29	비동기 입출력이 발생했을 경우 발생

## 백그라운드(Background) 작업과 포그라운드(Foreground) 작업

- 셸이나 콘솔창에서 명령을 입력하면 포그라운드로 실행됨
- 백그라운드에서 작업이 실행될 때는 그 작업이 실행되는지 확인할 수 없음
- **&(앰퍼센트):** 백그라운드로 작업을 실행하는 키워드  
ex) less /etc/passwd &
- **jobs:** 백그라운드에서 실행되는 작업을 확인하는 명령어 **#jobs**  
[+] : 현재 작업이 실행중임을 나타냄. fg나 bg명령어에서 우선순위를 가짐  
[-] : 실행될 작업을 나타냄. 현재 작업이 끝나고 실행될 프로세스를 의미
- **fg:** 백그라운드의 작업을 포그라운드로 전환하는 명령어. **#fg %PID**
- **bg:** 포그라운드 작업을 백그라운드로 전환하는 명령어 **#bg %PID**
- **pstree:** 실행중인 프로세스 상태를 트리 구조로 보여주는 명령어 **#pstree [유저명]**
- **nohup:** 부모 프로세스가 죽거나 종료되었더라도 자식 프로세스가 계속 작동할 수 있도록 bg에서 프로세스가 실행되도록 하는 명령어 **#nohup [명령어] &**
- **nice:** 프로세스의 실행 우선 순위를 결정하는 명령어 (-20~19) **#nice -[순위숫자] [명령어]**
- **renice:** 프로세스 실행 우선 순위를 변경하는 명령어 **#renice [nice값] -p [PID]**

프로세스의 점유율과 우선순위

- top: 프로세스들을 CPU점유율 순으로 표시하는 명령어  
#top [옵션]

```
top - 22:22:29 up 2:14, 1 user, load average: 0.04, 0.03, 0.00
Tasks: 285 total, 1 running, 284 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.5 us, 0.5 sy, 0.0 ni, 98.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3889.3 total, 560.9 free, 1062.0 used, 2266.5 buff/cache
MiB Swap: 3905.0 total, 3905.0 free, 0.0 used. 2566.4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+ COMMAND
 5514 root        20   0   22092   4064   3284 R   3.0   0.1   0:00.15 top
   517 root        20   0  175644   7228   6256 S   1.0   0.2   1:02.77 vmttoolsd
 5011 root        20   0      0      0      0 I   1.0   0.0   0:00.11 kworker/u256:0-events_fre+
     1 root        20   0  101988  11276   8272 S   0.0   0.3   0:01.87 systemd
```

옵션	의미
d	스크린을 갱신하는 시간 설정
p	주어진 pid에 대한 프로세스만 모니터링
c	명령 라인 모두 출력

단축키	설명
c	COMMAND필드에 명령어 전부 보이기
f	정보 필드 수정 (d로 선택)
i	유휴(idle) 프로세스 표시/감추기
k	지정한 PID 프로세스 KILL
l	시스템 정보 첫째행 보이기/감추기
m	메모리와 스왑 정보 보이기/감추기
M	메모리 사용량 기준으로 정렬
N	PID 번호순으로 오름차순 정렬
P	CPU 사용량순으로 정렬
r	프로세스 실행 우선 순위 변경 (renice)
T	프로세스 실행시간 순으로 정렬
u	해당 사용자의 프로세스 표시
z	컬러/흑백으로 보기
q	종료

# 프로세스

# 프로세스

```
top - 22:22:29 up 2:14, 1 user, load average: 0.04, 0.03, 0.00
Tasks: 285 total, 1 running, 284 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.5 us, 0.5 sy, 0.0 ni, 98.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3889.3 total, 560.9 free, 1062.0 used, 2266.5 buff/cache
MiB Swap: 3905.0 total, 3905.0 free, 0.0 used. 2566.4 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5514	root	20	0	22092	4064	3284	R	3.0	0.1	0:00.15	top
517	root	20	0	175644	7228	6256	S	1.0	0.2	1:02.77	vmtoolsd
5011	root	20	0	0	0	0	I	1.0	0.0	0:00.11	kworker/u256:0-events_fre+
1	root	20	0	101988	11276	8272	S	0.0	0.3	0:01.87	systemd

	행	항목	설명	행	항목	설명
시스템정보	첫번째 행 top	22:22:29	top을 실행한 현재 시간	네번째 행 MEM	3889.3 total	총 메모리
		up 2:14	시스템 가동 시간		560.9 free	남아 있는 메모리
		1 users	로그인 사용자 수		1062.0 used	사용된 메모리
		load average	시스템 평균 부하율 (1, 5, 15분)		2266.5 buff/cache	버퍼 메모리
	두번째 행 Task	285 total	동작중인 총 프로세스	다섯번째 행 SWAP	3905.0 total	총 스왑 메모리
		1 running	현재 실행중인 프로세스 수		3905.0 free	남아 있는 스왑 메모리
		284 sleeping	유휴 프로세스 수		0.0 used	사용된 스왑 메모리
		0 stopped	종료된 프로세스 수		2566.4 avail Mem	가용 메모리
		0 zombie	좀비 프로세스 수			
	세번째 행 CPU	1.5 us	사용자의 실행 프로세스 CPU 사용률			
		0.5 sy	시스템의 CPU 사용률			
		0.0 ni	nice에 의해 사용되는 CPU 사용률			
		98.0 id	사용되지 않은 CPU 사용율			
		0.0 wa	입출력 대기 상태의 CPU 사용율			
		0.0 hi	인터럽트에 의해서 사용된 CPU 사용율			
		0.0 si	인터럽트에 의해서 사용된 CPU 사용율			
		0.0 st	프로세스가 비자발적으로 대기하는데 사용되는 시간			

시스템 예약 (cron)

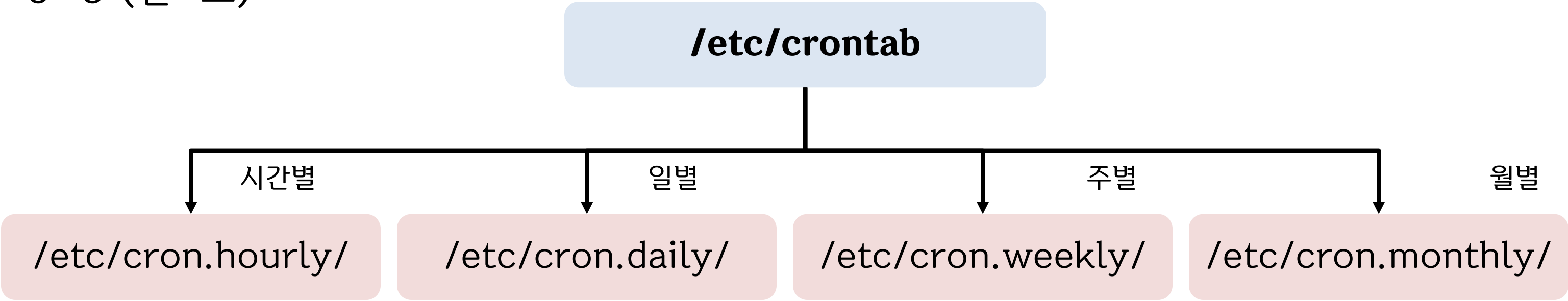
- cron: 주기적으로 반복되는 시스템 작업을 예약해놓는 것
- cron과 관련된 데몬(서비스)은 crond, 관련 파일은 /etc/crontab

```
#crontab [-u 유저명] [파일명]  
#crontab [-u 유저명] [-e | -l | -r ]
```

분 시 일 월 요일 사용자 실행명령

- 분: 0~59
- 시: 0~23
- 일: 1~31
- 월: 1~12
- 요일: 0~6 (일~토)

옵션	의미
-e	새로운 예약 작업을 작성함
-l	현재 예약되어 있는 작업 리스트 출력
-r	예약되어있는 작업 명령을 취소하거나 제거함





시스템 예약 (at)

- at: 일회성 시스템 작업을 예약해두는 것
  - 일회용이기 때문에 한번 실행되고 소멸됨
- #apt -y install rdate at
- #at [옵션] [시간] [날짜] [+증가시간]

at 사용

- 예약: #at [시간]  
명령어 입력 후 Ctrl + D
- 확인: #at -l
- 취소: #atrm [작업번호]

옵션	의미
-l	현재 예약되어 있는 작업 리스트 출력
-r	예약되어있는 작업 명령을 취소하거나 제거함

분류	형식	설명	예
시간	hh:mm	hh(시간):mm(분)으로 표현	16:18
	am, pm	am, pm을 이용하여 12시간 단위로 표현	4:18pm
	midnight noon teatime now	서술형 시간으로 표현 서술형 시간 다음에는 주로 증가 시간을 설정해서 사용함	midnight(00:00) noon(12:00) teatime(16:00) now(현재)
날짜	yyyy-mm-dd	일반적인 날짜 구성으로 표현	2012-4-13
	month num	4월 13일은 “April 13”이나 약자로 “Apr 13”으로 표현	April 13
	today, tomorrow	오늘이나 내일로 표현	

부트로더(BootLoader)

- HDD의 첫 번째 섹터에는 MBR(Master Boot Record)라는 부트 섹터(Boot sector)가 존재. 약 512byte
- 부트 섹터에는 설치되어 있는 운영체제로 부팅되도록 하는 부트 정보가 저장되어 있음
- 부트로더: 부트 정보를 담고 있는 부트 프로그램
- 리눅스의 부트로더: LILO (Linux LOader), GRUB, GRUB2, BURG(Brand-new Universal loader from GRUB)
- <https://www.gnu.org/software/grub/manual/grub/grub.html>

경로 및 파일	내부 파일	설명
/boot/grub	grub.cfg	/etc/grub.d 스크립트, /etc/default/grub를 이용하여 생성되는 설정 파일
	fonts	글꼴이 위치한 경로
/etc/grub.d	00_header	/etc/default/grub에 있는 설정 불러옴
	10_linux	설치된 리눅스 운영체제에 대한 메뉴 entry를 불러옴
	30_os-prober	타운영체제를 탐색하여 부트 메뉴에 추가해 줌
	40_custom	사용자 정의로 부트 메뉴에 추가해 주는 템플릿 역할
	90_persistent	사용자가 grub.cfg 파일 일부를 직접 수정할 수 있도록 함
/etc/default/grub	/etc/grub.d에 있는 스크립트들에서 사용할 옵션 설정을 포함하는 파일	
/usr/bin	grub-menulst2cfg	구버전의 menu
/usr/sbin	grub-install	grub을 HDD에 설치
/usr/sbin	grub-mkconfig	/boot/grub/grub.cfg 설정 파일을 생성해주는 도구



## GRUB Boot loader (Grand Unified Bootloader)

- GRUB 부트로더: 우분투를 부팅할 때 처음 나오는 선택화면으로, Windows의 BIOS화면과 같음
- 부트로더: 운영체제가 시동되기 이전에 미리 실행되면서 커널이 올바르게 시동되기 위해 필요한 모든 관련 작업을 마무리하고 최종적으로 운영 체제를 시동시키기 위한 목적을 가진 프로그램
- GRUB2의 설정파일: /boot/grub/grub.cfg  
읽기 전용이며, root도 편집하면 안됨.
- 설정법: /etc/default/grub (파일), /etc/grub.d/ (디렉터리) 을 수정한 후 #update-grub 명령 실행

## GRUB의 특징

- 부트 정보를 사용자가 임의로 변경해서 부팅 가능.
- 다른 여러 가지 운영체제와 멀티부팅 가능
- 대화형 설정을 제공하므로 커널 경로와 파일 이름만 알면 부팅이 가능

## GRUB2의 특징

- 셸 스크립트 문법을 사용하기 때문에 사용이 복잡함
- 조건식과 함수 등 셸 스크립트 문법을 사용가능
- 동적 모듈 로드 가능 (/boot/grub/i386-pd/파일명.mod)
- 그래픽 부트 메뉴를 지원
- ISO image를 사용해서 부팅 가능

GRUB Boot loader

/etc/default/grub

행번호	명령	설명
1	GRUB_DEFAULT=0	GRUB 목록 중 0번째 (첫번째)가 기본으로 선택되도록 함
2	GRUB_TIMEOUT_STYLE=hidden	hide일 경우 3행의 시간 동안 화면에 GRUB의 목록이 보이지 않도록 설정함
3	GRUB_TIMEOUT=0	처음 화면이 나오고 자동으로 부팅되는 시간을 초 단위로 설정 -1 로하면 자동으로 넘어가지 않고 사용자가 직접 선택할때까지 기다림
4	GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null    echo Debian`	초기 부팅 화면의 각 엔트리 앞에 붙을 배포판 이름을 추출 lsb_release -i -s는 "Ubuntu"글자를 출력함
5	GRUB_CMDLINE_LINUX_DEFAULT="maybe-ubiquity"	부팅 시 커널에 전달할 파라미터를 지정
6	GRUB_CMDLINE_LINUX=""	
7	GRUB_GFX_MODE	부팅 화면의 해상도를 설정. 기본값은 auto이며 '가로x세로x색상수'로 설정

<https://www.gnu.org/software/grub/manual/grub/grub.html>

## root 비밀번호 분실시 초기화

1. 부팅 시 **ESC**를 연타...
2. Ubuntu에 둔 상태로 키보드의 '**E**'키 누름 (Edit)

3. 아래 사진의 위치에 (/boot/vmlinuz-~~~~~줄의 맨 뒤)  
**init=/bin/bash** 입력 후 '**Ctrl+X**' 또는 '**F10**'으로 부팅

GNU GRUB version 2.04

```
*Ubuntu
Ubuntu용 고급 설정
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
```

Use the ↑ and ↓ keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the commands  
before booting or 'c' for a command-line.

GNU GRUB version 2.04

```
insmod part_msdos
insmod ext2
set root='hd0,msdos2'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2\
--hint-efi=hd0,msdos2 --hint-baremetal=ahci0,msdos2 82b219e3-bc5b-4f65\
-aea2-cd98b85e994a
else
    search --no-floppy --fs-uuid --set=root 82b219e3-bc5b-4f65-aea\
2-cd98b85e994a
fi
linux /boot/vmlinuz-5.13.0-40-generic root=UUID=82b219e3-\
bc5b-4f65-aea2-cd98b85e994a ro quiet splash $vt_handoff init=/bin/bash_
initrd /boot/initrd.img-5.13.0-40-generic
```

Minimum Emacs-like screen editing is supported. TAB lists  
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a  
command-line or ESC to discard edits and return to the GRUB  
menu.

## root 비밀번호 분실시 초기화

4. root 프롬프트가 부팅되고,  
**#mount -o remount,rw /**  
로 '/' 파티션을 rw 모드로 remount 시킴

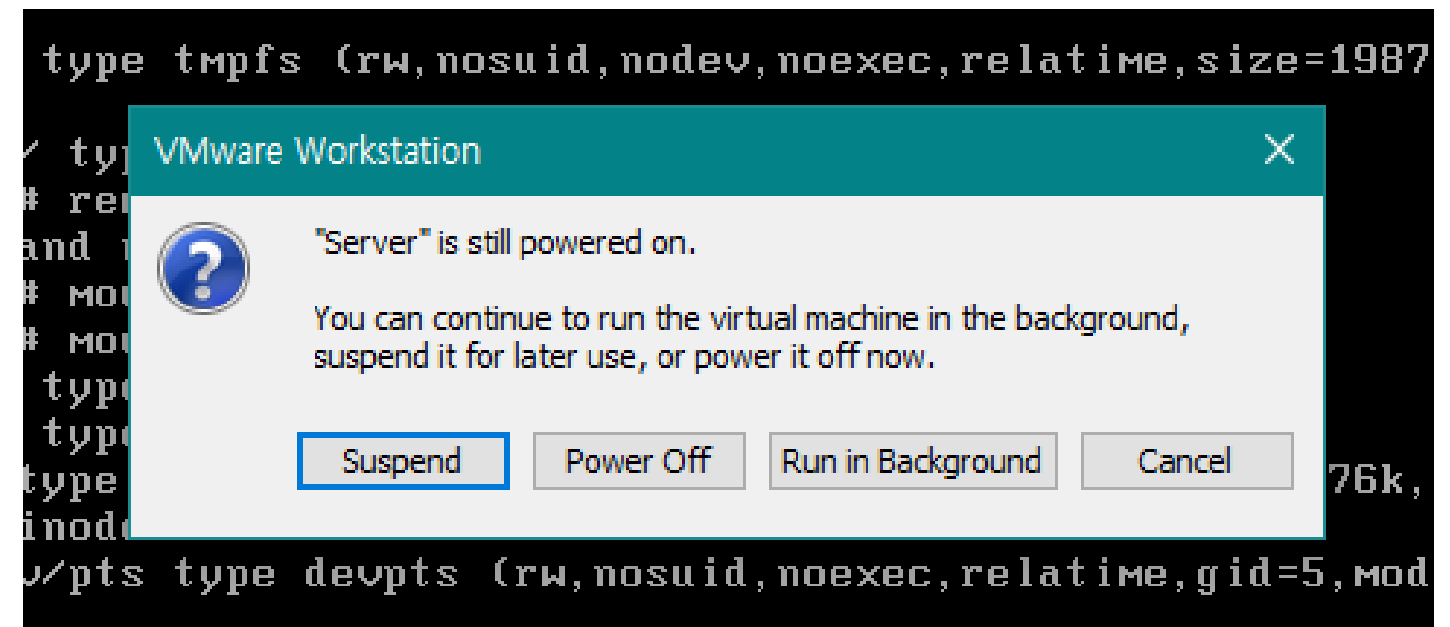
```
root@(none):/# mount -o remount,rw /  
root@(none):/# mount _
```

그냥 passwd 명령어를 입력하면 '/' 폴더가 read-only라 불가능..

```
bash: cannot set terminal process group (-1): Inappropriate ioctl for device  
bash: no job control in this shell  
root@(none):/# passwd  
New password:  
Retype new password:  
passwd: Authentication token manipulation error  
passwd: password unchanged  
root@(none):/# _
```

5. **#mount** 명령어 입력 후 Enter.  
**#passwd** 명령어를 통해 비밀번호 재 설정 후 VMware로 갔다 김

```
root@(none):/# passwd  
New password:  
Retype new password:  
passwd: password updated successfully  
root@(none):/#
```



## 부트로더 실행 변경

### /etc/default/grub 파일 수정

```
GNU nano 4.8 /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

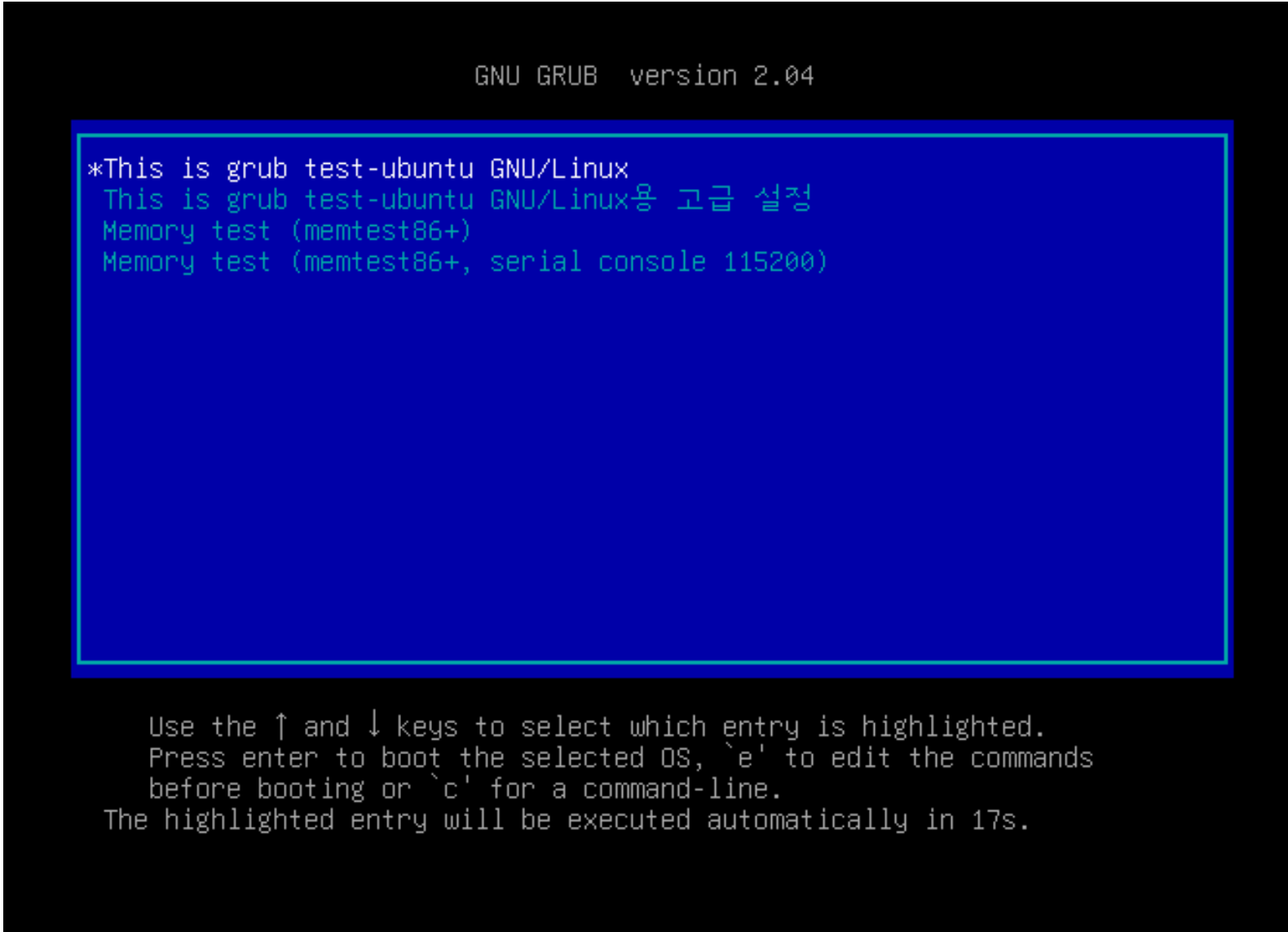
GRUB_DEFAULT=0
#GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=20
GRUB_DISTRIBUTOR="This is grub test-ubuntu"
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

### grub 업데이트 후 재부팅

```
root@Server:~/바탕화면# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
grub 설정 파일을 형성합니다 ...
리눅스 이미지를 찾았습니다: /boot/vmlinuz-5.13.0-40-generic
initrd 이미지를 찾았습니다: /boot/initrd.img-5.13.0-40-generic
리눅스 이미지를 찾았습니다: /boot/vmlinuz-5.13.0-30-generic
initrd 이미지를 찾았습니다: /boot/initrd.img-5.13.0-30-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
완료되었습니다
root@Server:~/바탕화면# reboot
```

부트로더에 비밀번호 설정

동작 확인



/etc/grub.d/00\_header

가장 마지막 행에 4줄 추가

grubuser: 새로운 GRUB 사용자명 (리눅스 사용자와 관계x)

```
GNU nano 4.8 /etc/grub.d/00_header
if [ "x${GRUB_INIT_TUNE}" != "x" ] ; then
    echo "play ${GRUB_INIT_TUNE}"
fi

if [ "x${GRUB_BADRAM}" != "x" ] ; then
    echo "badram ${GRUB_BADRAM}"
fi

cat << EOF
set superusers="grubuser"
password grubuser 1234
EOF
```

부트로더에 비밀번호 설정

동작 확인

```
root@Server:~# update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
grub 설정 파일을 형성합니다 ...
리눅스 이미지를 찾았습니다: /boot/vmlinuz-5.13.0-40-generic
initrd 이미지를 찾았습니다: /boot/initrd.img-5.13.0-40-generic
리눅스 이미지를 찾았습니다: /boot/vmlinuz-5.13.0-30-generic
initrd 이미지를 찾았습니다: /boot/initrd.img-5.13.0-30-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
완료되었습니다
root@Server:~# reboot
```

GRUB 설정 창에서 패스워드 확인

```
Enter username:
grubuser
Enter password:
_
```



01. 실행 중인 프로세스들의 CPU 사용률을 실시간으로 확인할 때 사용하는 명령으로 알맞은 것은?

- ① nice                      ② pstree
- ③ renice                    ④ top

02. 다음 중 ( 괄호 ) 안에 들어갈 내용으로 알맞은 것은?

- ① exec                      ② fork
- ③ inetd                    ④ standalone

( 괄호 ) 방식은 관련 프로세스가 메모리에 항상 상주하는 것이 아니라, 클라이언트의 서비스 요청이 들어왔을 때 관련 프로세스를 실행시키고 요청이 끝나면 자동으로 프로세스를 종료시키는 방식이다.

03. 다음 중 백그라운드 프로세스를 확인하는 명령으로 알맞은 것은?

- ① bg                        ② fg
- ③ jobs                    ④ nohup

04. 다음은 프로세스 아이디(PID)가 1222번인 프로세스의 우선순위 값을 -10으로 지정하는 과정이다. ( 괄호 ) 안에 들어갈 내용으로 알맞은 것은?

- ① ㉠ nice ㉡ -10          ② ㉠ nice ㉡ --10
- ③ ㉠ renice ㉡ -10       ④ ㉠ renice ㉡ --10

# ( ㉠ ) ( ㉡ ) 1222

05. 다음 설명과 관련 있는 명령으로 알맞은 것은?

- ① nice                      ② renice
- ③ nohup                    ④ bg

오랜 시간이 소요되는 백업 명령을 실행하려고 한다. 작업 중인 터미널이 닫혀도 계속 실행될 수 있게 명령을 내리려고 한다.

06. 다음 중 프로세스에 전송되는 시그널명과 시그널 번호를 확인할 때 사용하는 명령으로 알맞은 것은?

- ① ps
- ② kill
- ③ stat
- ④ signals

07. 다음 중 SIGTERM의 시그널 번호로 알맞은 것은?

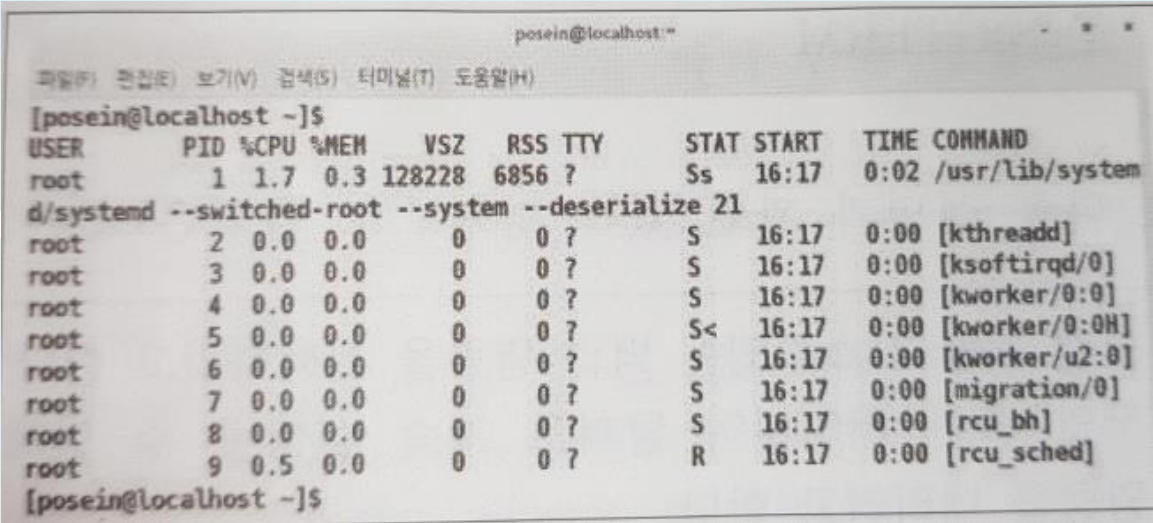
- ① 1
- ② 9
- ③ 15
- ④ 20

08. 실행중인 모든 프로세서를 확인하기 위해 사용하는 ps 명령 옵션으로 알맞은 것은?

- ① ef
- ② -a
- ③ aux
- ④ -f

09. 다음 결과에 해당하는 명령으로 알맞은 것은?

- ① ps
- ② top
- ③ pstree
- ④ pgrep



10. 프로세스 아이디(Process Identity)가 1222번인 프로세스를 강제 종료하려고 한다. 다음 ( 괄호 ) 안에 들어갈 내용으로 알맞은 것은?

- ① -1
- ② -9
- ③ -15
- ④ -20

# kill ( 괄호 ) 1222

11. 다음 설명으로 가장 알맞은 것은?

- ① init                      ② systemd
- ③ daemon                ④ xinetd

주기적이고 지속적인 서비스 요청을 처리하기 위해 메모리에 계속 실행되어 있는 프로세스로 백그라운드 프로세스의 일종이다.

12. 다음 중 백그라운드 프로세스와 가장 관련이 깊은 기호로 알맞은 것은?

- ① >                        ② &
- ③ %                        ④ ^

13. 다음은 ihduser가 cron 설정을 하는 과정이다. ( 괄호 ) 안에 들어갈 명령어의 옵션으로 알맞은 것은?

- ① -n                        ② -e
- ③ -i                        ④ -u

[ihduser@www ~]\$ crontab ( 괄호 )

14. 다음 설명과 같이 cron을 설정할 때의 날짜 형식으로 알맞은 것은?

- ① 10 11 \* \* 1-3 /etc/check.sh
- ② 11 10 \* \* 1-3 /etc/check.sh
- ③ 10 11 \* \* 1,3 /etc/check.sh
- ④ 11 10 \* 1,3 /etc/check.sh

매주 월요일과 수요일 오전 11시 10분에 점검 스크립트인 /etc/check.sh를 실행한다.

15. 다음 중 cron을 이용해서 매주 월요일부터 금요일까지 오후 12시에 백업 스크립트를 실행하려고 할 때 ( 괄호 ) 안에 들어갈 내용으로 알맞은 것은?

- ① 12 0 \* \* 1-5            ② 0 12 \* \* 1-5
- ③ 12 0 \* 1-5 \*            ④ 0 12 \* 1-5 \*

(    괄호    ) /etc/backup.sh

수고하셨습니다