

리눅스 초기화 과정

Contents

- 리눅스 디렉토리 구조
 - 부팅 과정과 시스템 초기화
 - 프로세스 및 데몬 관리
 - 소프트웨어 관리
-

리눅스 디렉토리 구조

□ 리눅스 디렉토리 구조

- 리눅스 디렉토리 구조
- 디렉토리 구성과 용도

□ 부팅 과정과 시스템 초기화

□ 프로세스 및 데몬 관리

□ 소프트웨어 관리

리눅스 디렉토리 구조

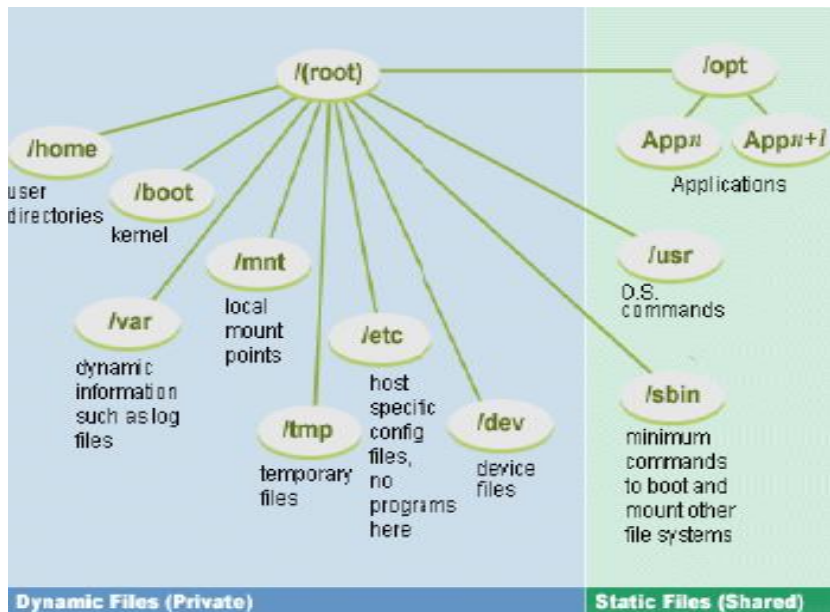
□ 리눅스 디렉토리 구조

■ 고정적인 파일(Static Files)

- 공유 가능한(Shared) 카테고리
- /usr, /sbin, /opt 등

■ 유동적인 파일(Dynamic Files)

- 환경 파일, 디바이스 파일, 커널 파일, 로그 파일, 임시 파일, 사용자의 홈 디렉토리 등 개인적인(Private) 카테고리
- /boot, /home, /tmp, /dev, /etc, /var 등



```
jyn@jyn-sslab: /
File Edit View Search Terminal Help
jyn@jyn-sslab:~$ cd /
jyn@jyn-sslab:/$ ls
bin    home    media   sbin    usr
boot   initrd.img  mnt     selinux var
cdrom  initrd.img.old  opt     srv     vmlinuz
dev    lib      proc    sys     vmlinuz.old
etc    lost+found  root    tmp
```

리눅스의 디렉토리 구조 (cont`d)

□ 디렉토리 구성과 용도

구분		설명
/		마운트 되는 리눅스 파일 시스템이 있는 최상위 디렉토리
/boot	BOOT	부팅에 필요한 커널과 부팅 할 때 필요한 핵심 파일이 위치하는 디렉토리
/bin	BIN aries	리눅스의 기본 명령어(binary)들이 위치하는 디렉토리
/dev	DEV ice	장치 파일이 위치하는 디렉토리 (문자장치(character device)와 블록 장치(block device))
/etc	ETC etera	애플리케이션과 서버 프로그램의 환경 설정에 필요한 설정 파일이 존재하는 디렉토리
/home	HOME	사용자 홈 디렉토리
/lib	LIB raries	시스템 운영 및 프로그램 구동 할 때 필요한 공유 라이브러리(shared library)와 부팅 할 때 사용되는 커널 모듈(kernel module)이 포함되어 있다.

리눅스의 디렉토리 구조 (cont`d)

□ 디렉토리 구성과 용도 (cont`d)

구분		설명
/lost+found	LOST + FOUND	시스템을 부팅할 때 파일시스템의 이상 유무를 진단하여 문제가 있으면 구동되는 fsck(File System Check) 명령어가 사용되는 디렉토리로 개별 파일시스템의 최상위 디렉토리마다 존재한다. - 만약 손상된 데이터를 복구하지 못했다면 문제가 발생된 파일 시스템에 존재하는 lost+found 디렉토리에 복구하지 못한 데이터 파일이 복사된다.
/mnt	MouNT	일시적으로 파일 시스템을 마운트 하는 장소
/media	MEDIA	이동식 미디어들을 마운트 하는 장소
/opt	OPeraTion	애드온(Add-on) 패키지가 설치되는 디렉토리 (표준 레드햇 리눅스 배포판에 존재하지 않는 패키지가 설치될 경우 이 디렉토리에 설치되기도 한다. 거의 사용하지 않는다.

리눅스의 디렉토리 구조 (cont`d)

□ 디렉토리 구성과 용도 (cont`d)

구분		설명
/proc	PRO Cess	프로세스와 시스템 정보를 제공하기 위한 목적으로 설계된 가상 파일 시스템을 사용하는 디렉토리
/root	ROOT	시스템 관리자인 root 사용자의 홈 디렉토리
/sbin	S ystem BI Naries	관리자가 사용하는 시스템 운영에 필요한 명령어가 존재하는 디렉토리
/tmp	Te MPorary	프로세스 생성 과정에서 발생하는 임시 파일을 저장하는 디렉토리(시스템 재 구동을 할 경우 이 디렉토리에 존재하는 파일은 모두 삭제되므로 파일을 저장하는 용도로 사용할 수 없다.
/usr	US eR	프로그램을 설치할 때 패키지에 포함된 대부분의 파일이 위치한 곳 - /usr/bin : 압축 파일과 네트워크 실행 파일, 자료 전송 파일 등 실행 파일들이 존재 - /usr/sbin : 주로 네트워크 관련된 실행 명령어와 실행 데몬들 - /usr/X11R6 : X-Window 시스템에 관련된 파일들이 존재 - /usr/include : 기본 C 라이브러리

리눅스의 디렉토리 구조 (cont`d)

□ 디렉토리 구성과 용도 (cont`d)

구분		설명
/var	VAR iable data	<p>내용이 자주 변경되는 가변(variable) 자료가 저장되는 디렉토리</p> <ul style="list-style-type: none">- 시스템 운영할 때 발생하는 로그(log)- 메일 송/수신할 때 임시로 저장되는 스푼(spool)- 프린터로 전송하기 위해 임시로 저장하는 프린트 스푼(spool) 데이터- 레드햇 배포판에서 패키지로 제공하는 아파치 웹 서버 관련 자료 (/var/www)- MySQL의 데이터 베이스- FTP의 최상위 디렉토리

부팅 과정과 시스템 초기화

□ 리눅스 디렉토리 구조

□ 부팅 과정과 시스템 초기화

- 부팅 초기화 과정
- 부트 로더
- init 프로세스

□ 프로세스 및 데몬 관리

□ 소프트웨어 관리

부팅 초기화 과정

□ 부팅 과정

■ AT&T사의 System V(System 5) 초기화 과정

- BSD 초기화 과정에 비해 더 복잡하고 다양한 파일의 조합을 사용
- Run-level 차이에 따른 서비스 구현이 가능
- 레드햇, 맨드레이크, 수세, 칼데라 등

부팅 초기화 과정 (cont`d)

□ 부팅 과정

■ 1단계 : **POST** 과정

- **P**ower **O**n **S**elf **T**est

- ROM BIOS(Basic Input Output System)

- 메모리의 용량과 연결된 하드디스크 개수, 키보드 및 마우스 연결 여부 등을 자동으로 점검

- 어느 장치로 부팅을 할 것인지 확인하여 설정된 장치(일반적으로 하드디스크)의 부팅에 필요한 블록 정보를 읽어온다.

■ 2단계 : **Boot Loader**

- BIOS 과정 이후 BIOS에서 설정된 부팅할 때 사용할 장치의 0번 트랙, 0번 실린더, 0번 헤드에 위치하는 부트 블록으로 제어권이 넘어간다.

- 리눅스에서는 부트 로더로 GRUB와 LILO를 제공한다.

부팅 초기화 과정 (cont`d)

□ 부팅 과정 (cont`d)

■ 3단계 : **Kernel**

- 부트 로더(LILO 및 GRUB)에 의해 메모리에 적재
- 리눅스 커널은 압축 이미지 파일
- 커널은 부트 로더의 정보를 참조하여 파일시스템의 위치를 참조한 후 / 파일시스템 마운트를 시도(파일시스템 종류는 자동 감지)
 - / 파일 시스템 마운트 실패 → 커널 패닉(panic) 혹은 시스템 종료(halt)
 - / 파일 시스템 읽기 전용(read-only)으로 마운트
- 스와퍼(swapper) 프로세스 구동
 - 프로세스 번호 0번
 - 메모리 관리, 디스크 관리, 프로세스 관리 등을 수행
 - 프로세스 번호 1 번인 init 라는 프로세스를 실행

■ 4단계 : **init 프로세스**

□ GRUB

■ GRand Unified Bootloader

- GNU 커널인 Hurd를 위해 개발
- 레드햇 7.x 부터 기본 부트 로더로 제공
- Ubuntu 10.10 은 Grub2를 사용함

■ 설정 파일 : **/etc/default/grub**

- /etc/default/grub 파일을 수정 후 update-grub 을 실행하면
자동으로 /etc/grub.d 의 파일들이 참조되어
/boot/grub/grub.cfg 가 생성된다

부트 로더 (cont`d)

□ GRUB (cont`d)

■ /boot/grub/grub.cfg 파일

□ kernel 영역

- 시스템의 kernel 파일의 위치 지정

□ initrd 영역

- 시스템의 initrd 이미지 파일의 위치 지정

□ timeout

- 부트 로더 대기 시간을 지정

□ title

- 멀티 부팅 시 GRUB 메뉴에서 해당 운영체제를 나타내는 문자로 출력

프로세스 및 데몬 관리

- 리눅스 디렉토리 구조
 - 파일 시스템
 - 부팅 과정과 시스템 초기화
 - 프로세스 및 데몬 관리
 - 프로세스 관리
 - ps, pstree, top
 - kill, killall
 - background, jobs, foreground
 - 데몬 관리
 - 소프트웨어 관리
-

프로세스 관리

□ 프로세스

■ “실행 중인 프로그램 상태”

■ 언제든지 실행 가능한 프로그램

■ 프로그램 자체인 실행 코드와 자료 그리고 실행 제어 정보가 들어 있는 **PCB**가 결합된 단위

□ 프로세스 제어 블록(**PCB** : **P**rocess **C**ontrol **B**lock)

- 운영 체제가 프로그램 실행 제어를 위하여 필요한 정보를 포함하고 있는 자료 구조

■ 프로세스 관련 용어

□ 포그라운드 프로세스(foreground process)

□ 백그라운드 프로세스(background process)

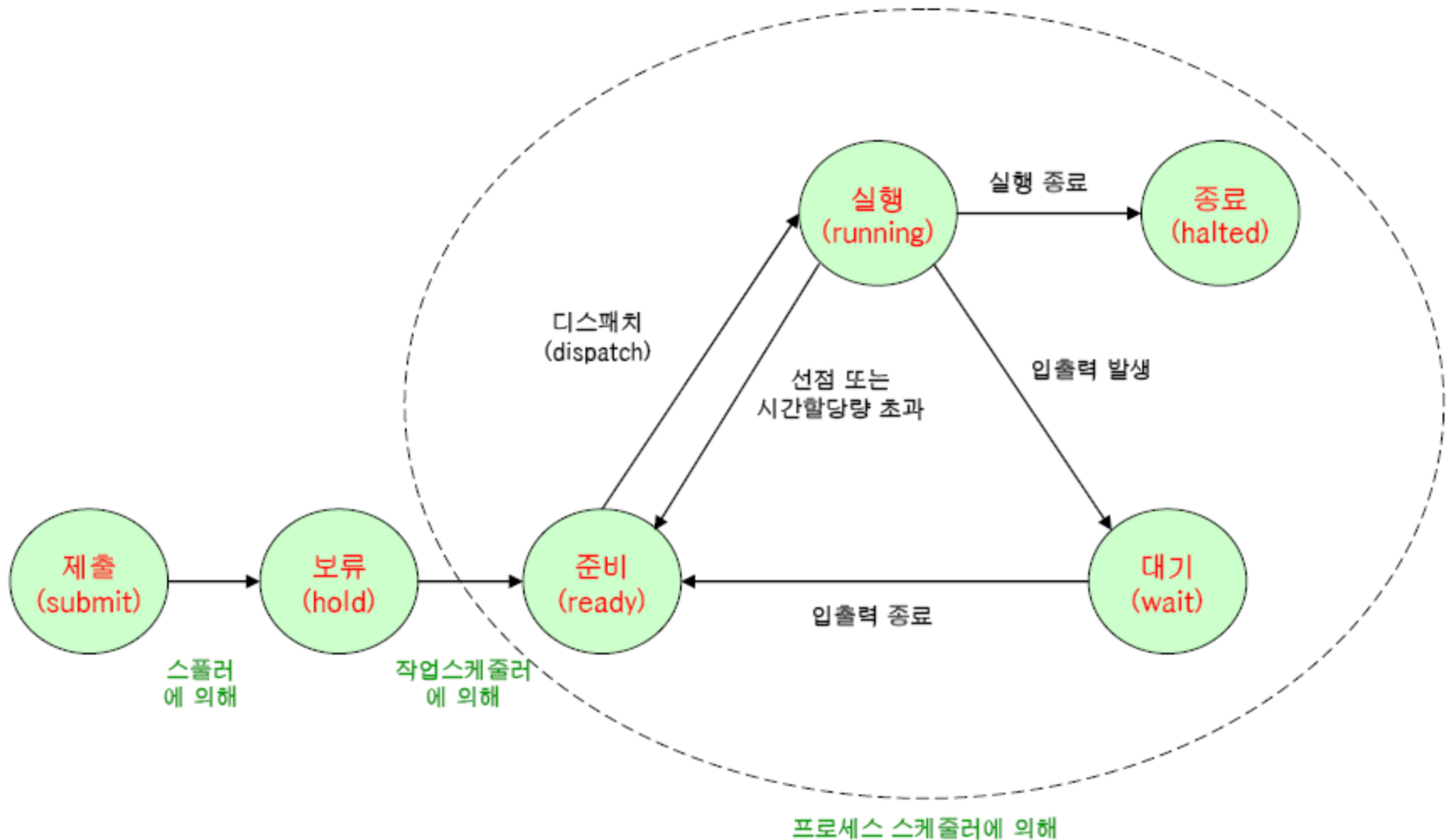
□ 프로세스 번호

□ 작업 번호

□ 부모 프로세스와 자식 프로세스

프로세스 관리 (cont`d)

□ 프로세스의 상태 구분



프로세스 관리 (cont`d)

□ ps

- 현재 진행중인 프로세스를 출력한다.

■ Syntax

ps [option]

-e	현재 사용자가 실행시킨 프로세스뿐만 아니라 제어 터미널을 갖고 있는 모든 프로세스를 보여준다.
-r	수행중인 프로세스만 보여준다
-x	제어 터미널이 없는 프로세스를 보여준다.
-u	프로세스의 소유자를 보여준다.
-f	프로세스 간에 부모와 자식 관계를 보여준다.
-l	자세한 형식으로 보여준다.
-p	지정된 프로세스에 대한 정보를 출력한다.
-t	지정된 터미널에 연관된 프로세스에 대한 정보를 출력한다.

프로세스 관리 (cont`d)

□ ps (cont`d)

```
jyn@jyn-sslabs: /  
File Edit View Search Terminal Help  
jyn@jyn-sslabs:/$ ps -ef | more  
UID      PID  PPID  C  STIME TTY      TIME CMD  
root      1    0    0  19:36 ?        00:00:02 /sbin/init  
root      2    0    0  19:36 ?        00:00:00 [kthreadd]  
root      3    2    0  19:36 ?        00:00:00 [ksoftirqd/0]  
root      4    2    0  19:36 ?        00:00:00 [migration/0]  
root      5    2    0  19:36 ?        00:00:00 [watchdog/0]  
root      6    2    0  19:36 ?        00:00:00 [events/0]  
root      7    2    0  19:36 ?        00:00:00 [cpuset]  
root      8    2    0  19:36 ?        00:00:00 [khelper]  
root      9    2    0  19:36 ?        00:00:00 [netns]
```

UID	해당 프로세스를 실행시키고 있는 프로세스 소유자의 username
PID	프로세스의 식별번호(Process ID)
PPID	해당 프로세스를 발생시킨 부모 프로세스 (Parent Process ID)
STIME	프로세스의 시작된 시간 혹은 날짜 (SYSV)
TTY	프로세스가 실행되고 있는 터미널을 의미
TIME	프로세스가 생겨난 이후 사용한 총 CPU 사용 시간
CMD	프로세스의 이름 또는 명령 실행 시 입력한 옵션 등을 나타낸다.

프로세스 관리 (cont`d)

□ ps (cont`d)

```
root@jyn-sslalab:/# ps -u
Warning: bad ps syntax, perhaps a bogus '- '? See http://procps.sf.net/faq.html
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      825  0.0  0.1  1856   556 tty4      Ss+  19:36   0:00 /sbin/getty -8 38
root      842  0.0  0.1  1856   556 tty5      Ss+  19:36   0:00 /sbin/getty -8 38
root      852  0.0  0.1  1856   556 tty2      Ss+  19:36   0:00 /sbin/getty -8 38
root      855  0.0  0.1  1856   560 tty3      Ss+  19:36   0:00 /sbin/getty -8 38
root      860  0.0  0.1  1856   560 tty6      Ss+  19:36   0:00 /sbin/getty -8 38
root     1020  0.7  3.8 31628 19680 tty7      Ss+  19:36   0:23 /usr/bin/X :0 -nr
root     1118  0.0  0.1  1856   560 tty1      Ss+  19:36   0:00 /sbin/getty -8 38
root     2765  7.3  0.2  5148  1468 pts/0      S    20:32   0:00 su
root     2774  4.5  0.3  5204  1780 pts/0      S    20:32   0:00 bash
root     2784  0.0  0.2  4588  1096 pts/0      R+   20:33   0:00 ps -u
root@jyn-sslalab:/#
```

USER	프로세스 소유자의 username
%CPU	프로세스가 사용한 CPU 사용 비율의 추정치 (BSD)
%MEM	프로세스가 사용한 메모리 사용 비율의 추정치 (BSD)
VSZ	프로세스가 사용하는 가상 메모리의 양
RSS	프로세스가 사용하는 실제(상주) 메모리 양
TTY	프로세스 제어 터미널 (값이 물음표 : 더 이상 제어 터미널과 연결되지 않음을 뜻한다.)

프로세스 관리 (cont`d)

□ ps (cont`d)

■ STAT : 프로세스의 현재 상태

-R	프로세스가 CPU를 사용하고 있는 상태(실행 상태)
-S	Sleep 상태 (실행준비가 되어 있는 모든 프로세스)
-T	정지된 상태(suspend)
-Z	좀비(zombie) 프로세스
-D	디스크 관련 대기 상태 (BSD)
-X	Dead 상태
-W	Paging 상태
-N	낮은 우선순위 상태 (BSD)
->	우선 순위가 인위적으로 높아진 상태 (BSD)
+	Foreground 프로세스 그룹 표시 (BSD)
-l	멀티 쓰레드 프로세스 (BSD)
-L	Lock (BSD)

프로세스 관리 (cont`d)

□ pstree

- 현재 진행중인 프로세스를 트리 구조로 출력한다.

■ Syntax

`pstree` [option]

- | | |
|----|--|
| -a | 프로세스들이 실행이 될 때, 인자나 옵션들이 함께 실행이 되었다면 이들을 모두 함께 표시해 준다. |
| -h | 현재 프로세스와 그의 부모프로세스들을 강조하여 출력해준다. |
| -p | 프로세스들의 오른쪽에 PID 정보를 함께 출력해준다. |
| -n | 같은 부모프로세스들을 가진 자식프로세스들끼리 정렬하여 보여준다. |

프로세스 관리 (cont`d)

□ pstree (cont`d)

```
[root@localhost ~]$ pstree -p
```

```
lly@lly-virtual-machine:~$ pstree -p
init(1)─ModemManager(860)─{ModemManager}(893)
                        └─{ModemManager}(895)
    ─NetworkManager(897)─dhclient(1003)
                        └─dnsmasq(1228)
    ─{NetworkManager}(900)
    ─{NetworkManager}(901)
    ─{NetworkManager}(902)
    ─accounts-daemon(1424)─{accounts-daemon}(1426)
                        └─{accounts-daemon}(1427)
    ─acpid(1222)
    ─bluetoothd(772)
    ─colord(1890)─{colord}(1929)
                └─{colord}(1931)
    ─cron(1121)
    ─cups-browsed(1177)
    ─cupsd(2322)
    ─dbus-daemon(668)
    ─getty(1059)
    ─getty(1063)
    ─getty(1069)
    ─getty(1070)
    ─getty(1073)
    ─getty(1213)
    ─kernelevents(1126)
```

프로세스 관리 (cont`d)

□ top

■ ps 명령어의 대화형 버전

- 2, 3 초 마다(사용자가 조정 가능) 프로세스의 목록을 보여주는 명령어
- top 프로그램의 단점은 CPU를 많이 소모한다는 것이다.

■ Syntax

top [option]

- | | |
|----|-----------------------------|
| -d | 스크린에 업데이트 간격을 조정한다. |
| -p | 해당 ID에 대한 프로세스의 정보만 출력한다. |
| -i | Zombie나 Idle 상태를 무시하고 시작한다. |

프로세스 관리 (cont`d)

□ top (cont`d)

```
[root@localhost ~]$ top
```

```
root@jyn-sslabs: /
File Edit View Search Terminal Help
top - 20:43:24 up 1:06, 2 users, load average: 0.02, 0.02, 0.05
Tasks: 129 total, 1 running, 128 sleeping, 0 stopped, 0 zombie
Cpu(s): 6.0%us, 3.3%sy, 0.0%ni, 90.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 508476k total, 422952k used, 85524k free, 46528k buffers
Swap: 492540k total, 0k used, 492540k free, 225452k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1020	root	20	0	52236	19m	7256	S	7.3	3.9	0:52.04	Xorg
2328	jyn	20	0	90148	13m	10m	S	1.3	2.7	0:11.43	gnome-terminal
1451	jyn	20	0	74968	19m	15m	S	0.7	3.9	0:22.12	vmware-user-loa
1453	jyn	20	0	76896	15m	11m	S	0.3	3.1	0:02.31	gnome-panel
1454	jyn	20	0	102m	10m	8760	S	0.3	2.2	0:02.03	metacity
2930	root	20	0	2620	1116	840	R	0.3	0.2	0:00.12	top
1	root	20	0	2888	1708	1240	S	0.0	0.3	0:02.27	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.20	ksoftirqd/0
4	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
6	root	20	0	0	0	0	S	0.0	0.0	0:00.44	events/0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuset
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pm

프로세스 관리 (cont`d)

□ signal 보내기

■ 프로세스 간의 통신 수단

- 하나의 프로세스가 다른 프로세스에게로 메시지를 보내기 위한 수단

이름	번호	의미
HUP	1	Hangup (실행 종료, 로그아웃 하거나 모뎀 접속을 끊을 때 보내진다)
INT	2	Interrupt (실행 종료, CTRL + C를 쳤을 때 보내진다)
QUIT	3	QUIT (실행 종료. CTRL+\ 를 쳤을 때 보내진다)
KILL	9	Kill (무조건 종료)
SEGV	11	Segmentation Violation (허가 되지 않은 메모리 영역에 접근하였다)
TERM	15	Terminate (가능한 한 정상 종료)
STOP	19	무조건적으로 그리고 즉각적으로 정지
TSTP	20	Stop (실행을 정지 후 다시 실행을 계속하기 위하여 대기, CTRL + Z)
CONT	18	Continue (STOP이나 TSTP에 의해 정지된 프로세스가 다시 실행을 계속한다)

프로세스 관리 (cont`d)

□ kill

- 프로세스에 시그널 보내기

- Syntax

`kill [option] PID`

`-n`

n 은 시그널 번호를 나타낸다.
(-9 : 프로세스를 강제로 종료시킨다.)

```
[root@localhost ~]$ kill 10137  
[root@localhost ~]$ kill -9 10138
```

프로세스 관리 (cont`d)

□ killall

- 프로세스 이름을 직접 이용

- Syntax

killall [option] process_name

```
[root@localhost ~]$ killall -9 ftp
```

□ Multitasking

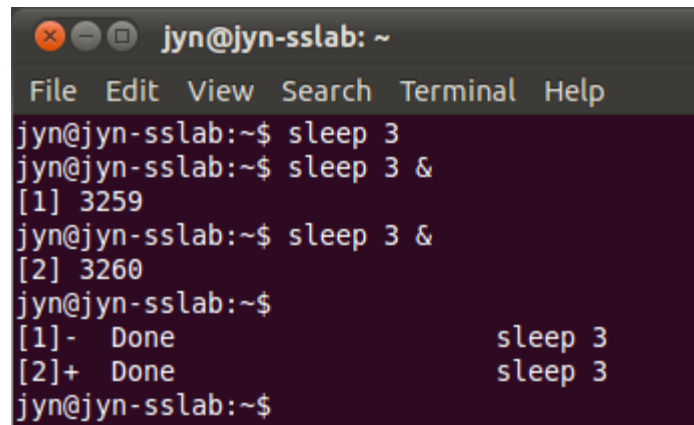
- 동시에 하나 이상의 프로그램을 수행시킬 수 있는 다중처리
- 리눅스에서는 foreground와 background로 구별한다.
 - foreground : 화면에 보여주면서 실행되는 상태
 - background : 화면에 보여주지 않으면서 실행되는 상태
 - suspend : 메모리에 올라가 있지만 정지된 상태

프로세스 관리 (cont`d)

□ background

- 여러 가지 프로그램을 동시에 실행 시키기 위해 &를 사용
 - 한 작업이 끝나기 전에는 다른 작업을 할 수 있음
 - vi는 입력을 받아서 실행하는 프로그램이므로 background로 실행될 수 없다.

■ Syntax



```
jyn@jyn-sslslab: ~  
File Edit View Search Terminal Help  
jyn@jyn-sslslab:~$ sleep 3  
jyn@jyn-sslslab:~$ sleep 3 &  
[1] 3259  
jyn@jyn-sslslab:~$ sleep 3 &  
[2] 3260  
jyn@jyn-sslslab:~$  
[1]-  Done                sleep 3  
[2]+  Done                sleep 3  
jyn@jyn-sslslab:~$
```

- [1] 5164, [2]5165는 process id
- Done 은 작업 완료를 알림
- +는 가장 최근에 수행한 작업을 의미

프로세스 관리 (cont`d)

□ jobs

- background로 수행중인 작업을 확인

- Syntax

jobs

```
jyn@jyn-sslabs: ~  
File Edit View Search Terminal Help  
jyn@jyn-sslabs:~$ sleep 10 &  
[1] 3265  
jyn@jyn-sslabs:~$ sleep 10 &  
[2] 3267  
jyn@jyn-sslabs:~$ jobs  
[1]-  Running                  sleep 10 &  
[2]+  Running                  sleep 10 &  
jyn@jyn-sslabs:~$
```

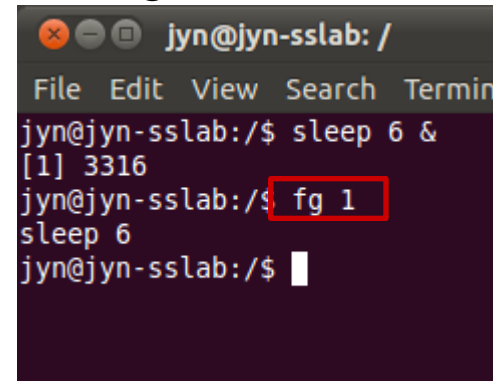
프로세스 관리 (cont`d)

□ foreground

- background로 돌고 있는 프로세스를 foreground로 돌릴 때

- Syntax

fg 작업번호



```
jyn@jyn-sslslab: /  
File Edit View Search Termin  
jyn@jyn-sslslab:/$ sleep 6 &  
[1] 3316  
jyn@jyn-sslslab:/$ fg 1  
sleep 6  
jyn@jyn-sslslab:/$
```

□ foreground와 background 차이점

- foreground 프로세스는 셸 프로세스의 하위 프로세스로 인식하고 셸 프로세스는 하위 프로세스가 종료 될 때까지 아무 작업도 하지 않는다. background 프로세스는 셸이 job ID와 프로세스 ID를 알려주고 바로 프롬프트를 사용자에게 돌려준다. 일반적으로 명령어 결과값을 얻기에 오랜 시간이 소요되는 명령어를 사용하는 경우에 사용한다.

데몬 관리

□ 데몬 (daemon)

■ 데몬 이란?

- 커널 상에서 백그라운드 모드로 작동하여 비활성화 상태에서 요청이 있을때만 동작하는 프로세스

- 데몬들이 커널 상의 백그라운드 모드로 존재하여 작동하지 않고 있을 때는 **CPU**에 부하를 주지 않지만, 시스템 자원(메모리)을 차지한다.

데몬 관리 (cont`d)

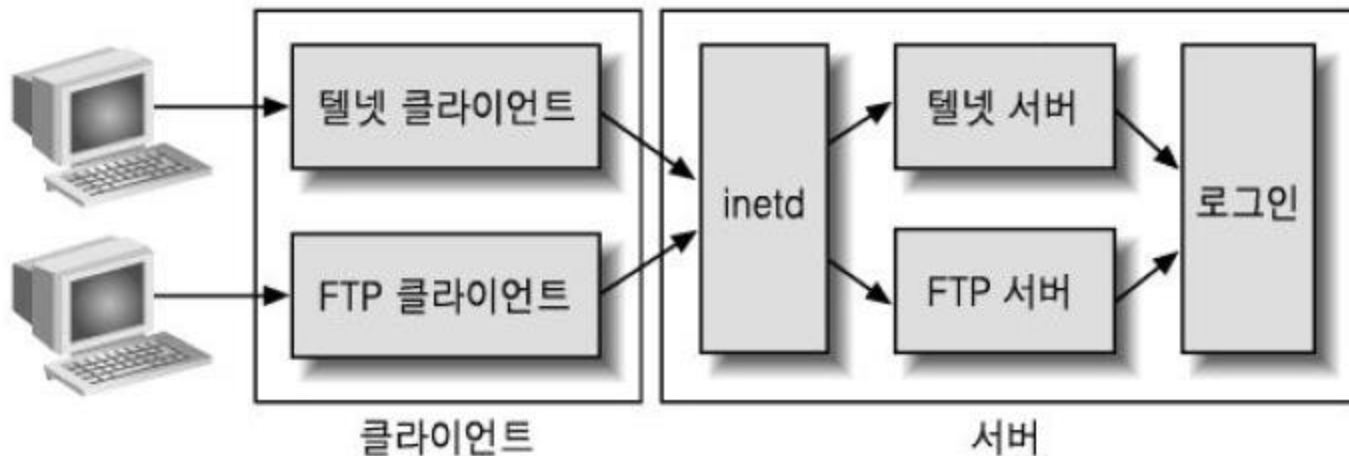
구 분	XINETD 환경에서 서비스되는 데몬들	STANDALONE으로 서비스되는 데몬들
의 미	xinetd데몬에 의해 실행되고 종료됨	xinetd와는 무관하게 독립적으로 서비스됨
메 모 리 상 주	xinetd에 의해 불러졌을 때만 메모리에 올려졌다가 서비스가 종료되면 메모리에서 제거됨. 따라서 메모리에 상주하지 않음. 주의: xinetd데몬 자체는 메모리에 상주함	항상 메모리에 독립적인 데몬으로 상주하고 있음.
응답속도	xinetd를 거쳐서 해당 서비스를 실행하기 때문에 standalone보다는 느림.	서비스 요청이 있을 경우 바로 응답을 하기 때문에 xinetd환경의 서비스 보다는 응답속도가 빠름.
설정파일	대부분 /etc/xinetd.d/서비스명 으로 설정파일이 존재함.	해당 프로그램이 설치된 위치에 독립적인 설정파일이 존재함. (rpm으로 설치되었을 경우에는 대부분 /etc디렉토리에 존재함.)
접근제어	tcp_wrapper에 의해서 접근제어됨. 데몬명 : tcpd (/etc/hosts.allow, /etc/hosts.deny)	접근제어를 자체적으로 하거나 아니면, 하지 않음. (시스템 자체의 방화벽 iptables등)과는 별도의 의미)
장 점	통합관리가능, 자원절약(메모리절약등), 접근제어가능	빠른응답속도
단 점	응답속도가 느림	자원낭비(메모리상주), 개별관리
서비스예	telnet, ftp, pop등	httpd, mysql, sendmail, named, sshd등
적 용	비교적 서비스요청이 많지 않은 서비스에 적용하는 것이 효율적임.	비교적 서비스 요청이 빈번한 서비스에 적용하는 것이 효율적임
기 타	xinetd로 서비스되는 서비스를 standalone으로 변경이 가능하고, standalone으로 서비스되는 서비스를 xinetd로 변환이 가능함. (지원하지 않는 서비스들도 있음.)	

데몬 관리 (cont`d)

□ inetd 데몬

■ inetd 데몬의 구성 및 동작

- /etc/inetd.conf 파일에서 데몬에 대한 설정을 읽고,
/etc/services 파일에서 설정된 포트 번호에 대해 클라이언트
의 요청이 있을 때 요청 데몬을 실행한다.

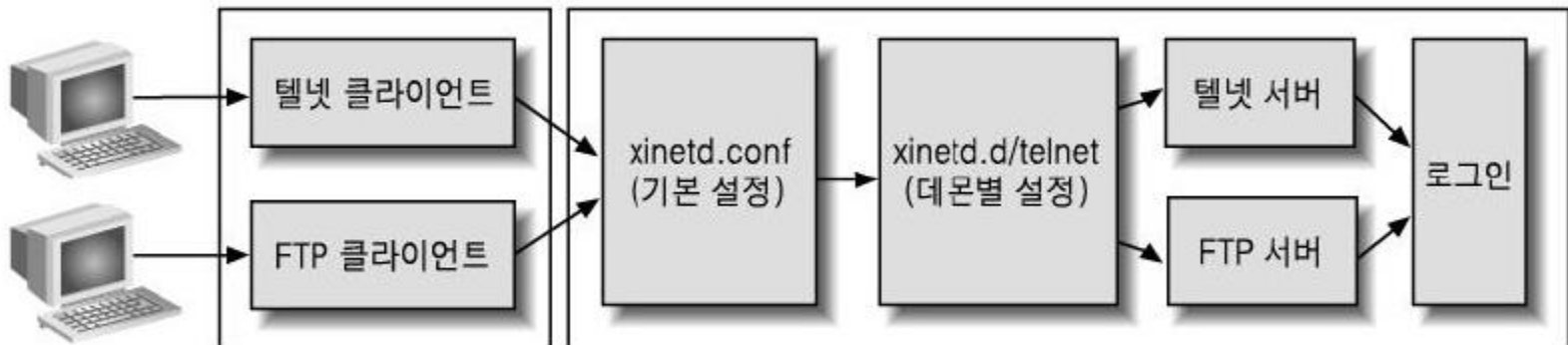


데몬 관리 (cont`d)

□ xinetd 데몬

■ Xinetd(eXtended inetd) 데몬의 구성 및 동작

- **/etc/xinetd.conf** : xinetd 데몬에 대한 공통적인 설정
- **/etc/xinetd.d/** : 디렉터리 안에 각 데몬별 설정 파일이 따로 존재



데몬 관리 (cont`d)

□ 데몬 종류와 기능

데몬 명	기능 설명
amd	요청이 있을 때 장치와 MFS 호스트를 마운트시켜 주는 자동 마운트 데몬
apmd	배터리 상태를 조회하는데 사용되는 데몬으로 배터리가 부족한 경우 시스템을 shutdown 하는데 이용
arpwatch	이더넷 카드와 IP 할당 관계를 보여주는 데몬
atd	특정 시간에 지정된 명령을 실행하거나 시스템 부하가 적을 때 배치 명령을 실행하도록 해준다.
autofs	파일 시스템을 이용하고 있을 때는 자동적으로 mount 되게 하고, 파일시스템을 이용하지 않을 때는 자동적으로 umount 시켜 준다.
bootparamd	옛 쉘 워크스테이션이 리눅스 박스에서 네트워크 부팅이 될 수 있게 해 주는 데몬
crond	사용자가 지정한 프로그램을 특정 시간에 주기적으로 실행할 수 있도록 해 주는 데몬
dhcpcd	동적 호스트 통제 프로토콜에 접근할 수 있도록 해 주는 데몬
gated	라우팅 데몬을 시작하거나 종료시키는 데몬
gpm	마우스를 지원해주는 데몬으로 시스템 shutdown 하는 옵션을 제공
httpd	아파치 데몬으로 웹 서버를 운영하고자 할 때 반드시 선택
inetd	인터넷을 관장하는 인터넷 슈퍼 서버 데몬으로 텔넷, FTP 등 서비스 요청 시에 될 수 있도록 관여한다.

데몬 관리 (cont`d)

□ 데몬 종류와 기능

데몬 명	기능 설명
inn	유즈넷 뉴스 서버로 지역 뉴스 서버를 설정할 수 있도록 해준다.
keytable	이를 선택해야 /usr/sbin/kbdconfig 로 키보드 유형을 변환할 수 있다.
kernelld	커널 모듈을 동적으로 적재하는 것을 관리하는 데몬
lpd	프린터 서비스 데몬
mcserve	Midnight Command(mc)끼리 네트워크를 공유시키는 데몬
named	호스트 이름을 IP로 변환시켜 주는 DNS 데몬
netfs	삼바, NFS, NCP 등을 mount 하거나 umount 할 때 필요한 데몬
network	설정된 네트워크 인터페이스가 부팅 시 작동될 수 있도록 해주는 스크립트
nfs	NFS 서버로 작동할 수 있게 해 주는 데몬
nscd	NIS 또는 NIS+를 사용하고자 할 때 선택해 주어야 한다.
portmap	RPC 연결에 관여하는 데몬으로 NFS, NIC 사용시 선택
postgresql	postgresql DB를 이용하고자 할 때 선택
random	시스템에 난수 발생 및 저장 스트립트
routed	자동 IP 라우터 테이블이 가능하도록 해준다.

데몬 관리 (cont`d)

□ 데몬 종류와 기능

데몬 명	기능 설명
ruserd	어떤 네트워크를 특정 사용자가 있는지 검색해 주는 서비스 데몬
rwhod	원격 사용자가 접속되어 있는 사용자의 목록을 볼 수 있게 해주는 데몬
sendmail	메일 서버를 설정하기 위해서 반드시 설정
smb	삼바 서버가 활성화 되도록 해 주는 데몬
sound	부팅과 종료 시 사운드 믹서 설정을 저장하고 복원해 주는 데몬
syslog	시스템에서 발생된 각종 사건을 기록해 주는 서버로 /var/log에 기록된다.
ypbind	NIS/YP 클라이언트용을 실행하는 서버
yppasswd	NIS 클라이언트 사용자가 패스워드를 변경할 수 있도록 해주는 데몬
ypserv	표준 NIS/YP 네트워크 프로토콜 서버

데몬 관리 (cont`d)

□ 데몬 확인

```
jyn@jyn-sslab: ~  
File Edit View Search Terminal Help  
jyn@jyn-sslab:~$ ps -acx | more  
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html  
PID TTY STAT TIME COMMAND  
1 ? Ss 0:02 init  
2 ? S 0:00 kthreadd  
3 ? S 0:00 ksoftirqd/0  
4 ? S 0:00 migration/0  
5 ? S 0:00 watchdog/0  
6 ? S 0:00 events/0  
7 ? S 0:00 cpuset  
8 ? S 0:00 khelper  
9 ? S 0:00 netns  
10 ? S 0:00 async/mgr  
11 ? S 0:00 pm  
12 ? S 0:00 sync_supers  
13 ? S 0:00 bdi-default  
14 ? S 0:00 kintegrityd/0  
15 ? S 0:00 kblockd/0  
16 ? S 0:00 kacpid  
17 ? S 0:00 kacpi_notify  
18 ? S 0:00 kacpi_hotplug  
19 ? S 0:00 ata_aux  
20 ? S 0:00 ata_sff/0
```


소프트웨어 관리

- 리눅스 디렉토리 구조
 - 파일 시스템
 - 부팅 과정과 시스템 초기화
 - 프로세스 및 데몬 관리
 - 소프트웨어 관리
 - APT
-

□ APT (Advanced Packaging Tool)

- 새로운 소프트웨어의 패키지 설치, 존재하는 소프트웨어의 업그레이드, Ubuntu 전체 시스템의 업그레이드를 수행
- 단순한 Command – Line 도구
- 터미널에서 간단히 실행 가능

□ apt-get 유틸리티

- 패키지 설치, 업데이트, 삭제 등이 가능한 유틸리티
- Sudo apt-get install package_name

```
jyn@jyn-sslab: ~  
File Edit View Search Terminal Help  
jyn@jyn-sslab:~$ sudo apt-get install tilda  
[sudo] password for jyn:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  linux-headers-2.6.35-22 linux-headers-2.6.35-22-generic  
Use 'apt-get autoremove' to remove them.  
The following extra packages will be installed:  
  libconfuse0  
The following NEW packages will be installed:  
  libconfuse0 tilda  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 107kB of archives.  
After this operation, 737kB of additional disk space will be used.  
Do you want to continue [Y/n]? y  
Get:1 http://kr.archive.ubuntu.com/ubuntu/ maverick/universe libconfuse0 i386 2.7-1 [26.2kB]  
Get:2 http://kr.archive.ubuntu.com/ubuntu/ maverick/universe tilda i386 0.09.6-2 [80.8kB]  
Fetched 107kB in 3s (35.1kB/s)  
Selecting previously deselected package libconfuse0.  
(Reading database ... 143557 files and directories currently installed.)  
Unpacking libconfuse0 (from .../libconfuse0_2.7-1_i386.deb) ...  
Selecting previously deselected package tilda.  
Unpacking tilda (from .../tilda_0.09.6-2_i386.deb) ...
```

□ apt-get 유틸리티(con't)

■ sudo apt-get remove package_name

- package_name에 해당하는 패키지를 삭제하는 명령
- 두 개 이상의 패키지를 지울 때는 스페이스 키로 구분

```
jyn@jyn-sslalab: ~  
File Edit View Search Terminal Help  
jyn@jyn-sslalab:~$ sudo apt-get remove tilda  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  linux-headers-2.6.35-22 linux-headers-2.6.35-22-generic  
Use 'apt-get autoremove' to remove them.  
The following packages will be REMOVED:  
  tilda  
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.  
After this operation, 602kB disk space will be freed.  
Do you want to continue [Y/n]? y  
(Reading database ... 143597 files and directories currently installed.)  
Removing tilda ...  
Processing triggers for man-db ...  
Processing triggers for desktop-file-utils ...  
Processing triggers for python-gmenu ...  
Rebuilding /usr/share/applications/desktop.en_US.utf8.cache...  
Processing triggers for python-support ...  
jyn@jyn-sslalab:~$
```

□ apt-get 유틸리티(con't)

■ sudo apt-get update

- /etc/apt/sources.list 에 나열된 서버들의 최신 정보를 로컬에 업데이트 해준다

■ sudo apt-get upgrade

- Ubuntu 의 버전을 업데이트 한다 ex) 9.10 -> 10.04

■ sudo apt-get dist-upgrade

- sudo apt-get update 를 의존성 검사를 하면서 업데이트 한다

□ apt 유틸리티

■ apt-cache search package_name

- 해당하는 패키지와 연관된 패키지들을 확인하는 명령

■ apt-cache show package_name

- 패키지의 정보를 보는 명령

■ dpkg -l

- 현재 설치된 패키지들의 목록을 보여주는 명령

```
jyn@jyn-sslalab: ~  
File Edit View Search Terminal Help  
jyn@jyn-sslalab:~$ dpkg -l | more  
Desired=Unknown/Install/Remove/Purge/Hold  
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend  
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)  
||/ Name                               Version  
| Description  
+++-----  
ii acpi-support                        0.137  
    scripts for handling many ACPI events  
ii acpid                              1.0.10-5ubuntu4  
    Advanced Configuration and Power Interface event daemon
```

□ apt 유틸리티

■ sudo add-apt-repository "URL"

- 저장소 목록에 원하는 서버를 추가하는 명령
- 수행 한 후 sudo apt-get update 를 수행해야 한다

```
jyn@jyn-sslab:~$ sudo add-apt-repository 'deb http://packages.linuxmint.com/ julia main'  
jyn@jyn-sslab:~$ sudo apt-get update
```



```
Get:1 http://security.ubuntu.com/ubuntu/ maverick-security Release.gpg [190B]  
Ign http://security.ubuntu.com/ubuntu/ maverick-security/main Translation-en  
Ign http://security.ubuntu.com/ubuntu/ maverick-security/main Translation-en_US  
Get:5 http://extras.ubuntu.com/ubuntu/ maverick Release.gpg [72B]  
Ign http://extras.ubuntu.com/ubuntu/ maverick/main Translation-en  
Hit http://kr.archive.ubuntu.com/ubuntu/ maverick/main Sources  
Ign http://packages.linuxmint.com/ julia/main Translation-en  
Ign http://packages.linuxmint.com/ julia/main Translation-en_US  
Ign http://security.ubuntu.com/ubuntu/ maverick-security/multiverse Translation-en  
Ign http://security.ubuntu.com/ubuntu/ maverick-security/multiverse Translation-en_US
```