

1. 시스템 부팅

1) 시스템 부팅의 전 과정

1 컴퓨터가 켜지면, POST 과정에 의해 시스템이 초기화 (Bios booting)

2 Boot record나 하드인 경우 MBR (Master boot record)를 읽어 들임 (Unix-BOS)

3 LILO (Linux Loader) (GRUB, GRand Unified Boot loader)가 실행

- 만일, 디폴트인 리눅스가 로딩되기 전에 Ctrl, Shift, Alt 중 하나를 누르고 있으면, LILO는 부팅할 운영체제를 물어봄 (리눅스가 여러 개 설치되어 있거나, 서로 다른 LILO가 설치되어 있는 경우)

4 Kernel이 메모리로 로딩되며, 만일 커널이 압축되었다면 압축을 해제

- vmlinux 커널은 2.6.4 이런 식으로 세자리 숫자 버전, 전 리눅스 공동

1) 시스템 부팅의 전 과정

5

커널은 하드디스크, 플로피, 네트워크 어댑터, CD롬 등을 검사하며, 디바이스 드라이버를 설정 : /dev/

6

root 파일 시스템을 마운트시킴

- root 파일 시스템은 'dev'나 LILO에 의해 설정되어 있으며, 파일 시스템의 형태는 자동적으로 검출

7

커널은 /etc/init을 백그라운드로 실행

- 'init', 'inittab' 파일의 내용에 따라 실행 (유닉스)

8

init는 /etc/rc를 실행한다.

- 'rc'는 /etc/rc.local이나 /etc/rc.[0-9]등을 실행

1) 시스템 부팅의 전 과정

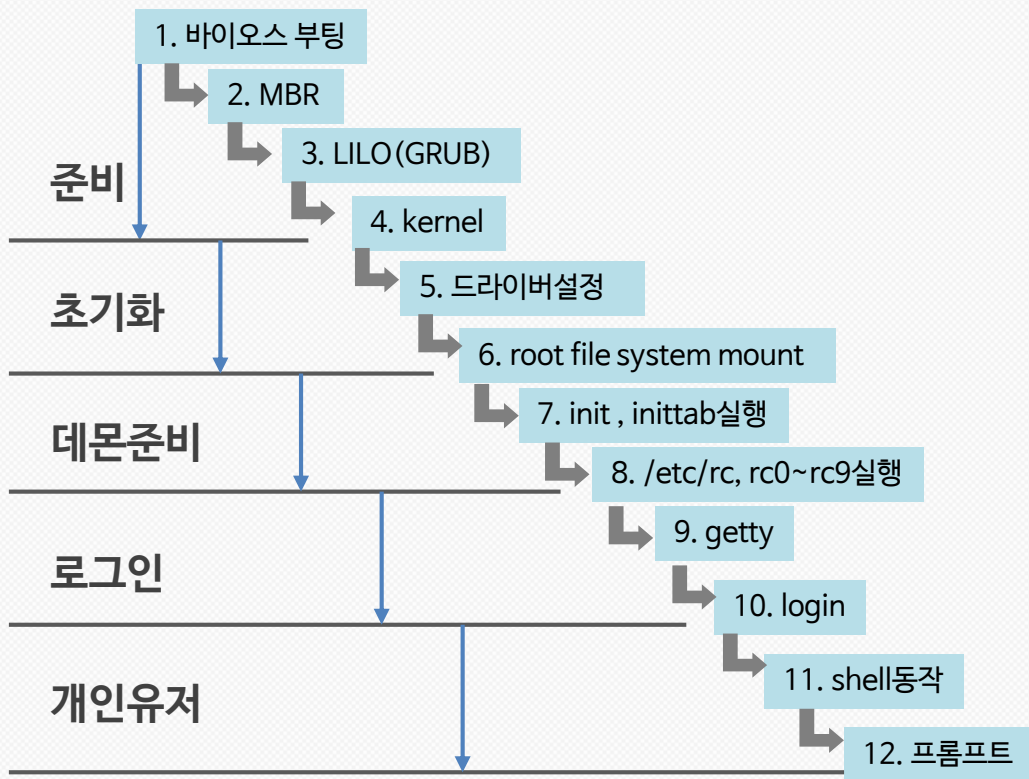
9 'init' 프로그램은 가상 콘솔을 위해 **getty**를 실행 (telnet, sshd)

10 **로그인**(ID와 패스워드를 입력)

11 **shell이 작동**하고 bash shell이면 .bashrc를 , tcsh shell이면 .profile을 불러들임

12 **프롬프트가 표시**됨

1) 시스템 부팅의 전 과정



2) run level

✓ rc, inittab 등에서 프로세스가 실행하는 레벨을 지정

✓ 지정된 숫자는 runlevel로 다음과 같은 의미임

✓ init runlevel로 명령 시 다음이 실행됨

2) run level

run level	내용
0	halt, 시스템 종료 레벨
1	single user mode, 네트워크를 지원하지 않는 단일사용자 레벨
2	multi-user without NFS, 3번과 같지만 네트워킹을 지원하지 않음
3	full multi-user mode, 일반적인 부팅레벨 (full network)
4	unused, 사용자 정의 레벨
5	X11, X윈도우로 시작을 의미하는 레벨 (xdm이 사용됨) ➡ X로 바로 부팅
6	reboot, 재부팅을 의미하는 레벨

2. 시동, 초기화, 셋다운

1) shutdown

✓ Power off 전 shutdown은 필수 : 디스크 등 parking, 프로세스 정리

구분	설명
shutdown	시스템 끄기
r	reboot
c	수행중인 shutdown 정지
y	shutdown 수행을 묻지 말고 정지
now	지금 수행
shutdown -r -y now	shutdown 수행을 묻지 말고 재시작
shutdown -y now	shutdown 수행을 묻지 말고 정지

2) at 데몬

at 데몬

시간을 지정하여 특정 프로세스를 기동시켜주는 데몬

✓ at 시간지정 (시간은 표준 방식)

예

at 09:20 04:25:2010
at now + 2 hours

2) at 데몬

✓ atq (at -l) at으로 지정한 job을 보여줌

✓ atrm job번호 : at으로 지정된 job을 삭제

✓ 다시 서버를 부팅하면 내용은 살아 있지 않음

✓ 단 telnet 등을 닫아도 실행하려면 nohup 명령사용

3) cron 데몬

cron 데몬

- 이미 학습한 부분
- 예약 작업을 지정함

✓ cron 테이블에 규칙에 맞도록 작업을 올려놓으면 해당 시간, 일자에 실행함

✓ crontab은 userid 단위로 생성

✓ crontab은 init 데몬이 실행해주며, 재부팅이 되더라도 상황이 계속 남음

3) cron 데몬

〈crontab 사용법〉

구분	설명
crontab -e	job 등록
crontab -l	crontab에 등록된 작업 확인
crontab -r	crontab에 등록된 작업 삭제

3. 시스템 측정 명령

1) sar

sar

- 복습 명령어
- CPU사용량을 순간적으로 조사

예1

- `sar 1 100` : CPU 사용량을 1초 간격으로 100번 조사
- `sar -d 1 100` : disk io 정보를 1초 간격으로 100번 조사

1) sar

〈보여지는 항목〉

구분	설명
%user	user level (application)이 사용하는 CPU의 양 (%)
%nice	nice priority(우선순위가 지정된 job)를 가진 user level이 사용하는 CPU의 양(%)
%system	system level(kernel)이 사용하는 CPU양(%)
%iowait	디스크등의 i/o의 속도차로 인하여 CPU가 쉬게 되는 시간 비율(%)
%steal	프로세스(CPU상 하나의 실행단위)가 CPU 등을 바꾸거나, 잡 우선순위에 의하여 다른 잡이 들어오거나 하는 순간 때문에 사용된 시간 비율(%)
%idle	CPU가 널널한 비율($100 - (\%user + \%nice + \%system)$)



apt-get install sysstat 설치

2) top

top

- 복습 명령어
- 현재 CPU에 수행되고 있는 프로세스 상황을 CPU 자원을 많이 사용하는 순서대로 보여줌

✓ 자동으로 계속 보여줌

✓ 메모리 상황, swap 상황을 보여줌

2) top

- ✓ 우선순위, 버추얼 메모리, 리소스 등을 보여줌
- ✓ 가장 중요한 것은 처리시간을 얼마나 사용하고 있는가임
- ✓ 만일 계속하여 처리시간을 잡아먹고 있는 프로세스가 있다면, 그 프로세스는 hang인 상태인 경우(무한루프)일 수 있으며 이 프로세스는 kill -9 job#로 처리해주는 경우가 있을 수 있음

3) fsck, du



fsck

- file system check
- 파일시스템의 에러나 크래시를 복구함
- 각 파일시스템을 unmount하고 단일사용자 모드에서 하는 것이 효과적 (장애 시)
- 디스크 작업 시 본 명령 실행 시 디스크 크래시 주의



du

- 파일 스페이스의 사용량을 보여줌
- 현재 디렉토리 하위 모든 파일의 스페이스 차지를 보여줌
- ls에서 보이는 실크기와 디스크에 할당된 크기는 차이가 있음

4) 명령어 총정리

✓ 각 명령어는 시스템에 따라 옵션이 조금씩 차이 있을 수 있음

✓ man으로 확인 필수

4) 명령어 총정리

구분	명령어	설명
프로세스 확인	jobs	현재 id, prompt에서 수행되는 job 확인
	ps	각종 프로세스 상황 확인
	pstree	트리구조로 프로세스 상황 확인
	top	프로세스가 CPU를 차지하는 순서로 확인
	ulimit	ulimit -a 로 프로세스 자원한도 확인
	/proc 디렉토리	현재 수행중인 프로세스는 /proc 디렉토리에 정보파일로 존재함 (대부분 리눅스, 유닉스는 아님)
CPU 상황확인	vmstat	CPU 활동 상황 확인
	sar	순간적 CPU활당률, 디스크(sar -d)i/o 상황을 보여줌
	vi /proc/CPUinfo	현재 설치된 CPU의 자세한 상황을 보여줌

4) 명령어 총정리

구분	명령어	설명
Memory 확인	top	메모리 상황도 top에서 보임
	free	메모리 상황보기
Disk 확인	df -k	파일 시스템별 마운트 상황, 배분 크기가 보임
	du -a	파일별 스페이스 사용량 보임
	iostat	현재 io발생상황을 보여줌
네트워크 관련	ifconfig	랜카드 등 인터페이스 상황을 보여줌
	ping	네트워크 연결설정을 보는 기본 명령
	netstat -na	Tcp/ip 어플리케이션 차원의 서비스 상태를 보여줌

4. 시스템 시동, 셧다운 실습

1) 실습하기

실습내용

(1) shutdown → 이후 부팅과정 점검

(2) at 데몬

(3) cron 데몬

5. 시스템 측정 명령 실습

1) 실습하기

실습내용

- (1) 프로세스 확인
- (2) CPU 상황확인
- (3) Memory 확인
- (4) Disk 확인
- (5) 네트워크 관련



학습활동

* 일시정지 버튼을 클릭하고 학습활동에 참여해 보세요.

Q

이번 강의에서 cron 데몬과 at 데몬에 대해 배웠습니다.
데몬(demon/daemon)이란 무엇일까요? 스스로 조사해 보도록 합시다.

Q 이번 강의에서 cron 데몬과 at 데몬에 대해 배웠습니다.
데몬(demon/daemon)이란 무엇일까요? 스스로 조사해 보도록 합시다.

A

- 데몬(daemon)은 주기적인 서비스 요청을 처리하기 위해 계속 실행되는 특별한 프로그램(telnetd, ftpd...)입니다.
- 데몬이라는 특별한 프로그램이 항상 백그라운드로 실행되고 있으며, 어떠한 요청이 들어오면 바로 처리해주는 특별 프로그램입니다.
- cron과 at은 데몬이 처리해 주는 명령입니다.
- cron과 at에 등록된 일에 대하여 데몬이 항상 실행되고 있다가 지정된 시간 등에 해당 일을 수행되도록 하고 있습니다.

Q 이번 강의에서 cron 데몬과 at 데몬에 대해 배웠습니다.
데몬(demon/daemon)이란 무엇일까요? 스스로 조사해 보도록 합시다.

A

- 우리가 웹서버를 사용할 때, 웹서버에는 데몬이 항상 수행되고 있어서 웹페이지의 요청이 들어오자마자 웹 페이지를 보여주는 역할을 수행하는 것입니다.
- ssh의 접속이나 ftp도 데몬이 응답합니다. 우리가 프로세스를 조회해 보면 sshd, ftpd와 같이 해당 데몬이 항상 수행되고 있는 것을 알 수 있습니다.