3강. 리눅스 시작

- 운영체제의 로딩 과정(일반적인 운영체제의 부팅 과정)
- 전원이 들어오면 ROM BIOS의 부트 코드가 실행됨
- 디스크의 MBR에 있는 부트로더를 로드하여 실행
- 커널이 로드되고 시작됨

-커널 이미지는 /boot/vmlinuz
-하드웨어 검사
-init 프로세스 생성

- init 프로세스에 의해 초기화 작업이 수행

- MBR(Master Boot Record) 의 정보
- 부트로더

운영체제를 메모리에 로드하여 실행시킴

- 파티션 테이블

-파티션의 시작과 끝 주소
-부팅 가능한 파티션 여부
-파티션 타입

- 부트로더의 종류
- LILO(LInux LOader)

전통적인 리눅스 부트로더

- GRUB(Grand Unified Bootloader)

다양한 기능을 제공하는 편리한 부트로더

- 부트로더의 동작
- 부팅할 OS 선택 가능
- 부팅 파티션에 있는 부트섹터를 통해 실행
- 부트로더는 커널 이미지를 찾아 메모리에 적재

- 커널의 부팅 과정
- 압축된 커널 이미지를 메모리에 로드하여 압축을 풂
- 메모리에서 커널을 실행
- 커널의 기능을 점검하고 설치된 하드웨어를 점검
- root 파티션을 읽기 전용으로 마운트(mount)하여 디스크 검사
- 디스크 검사 후 root 파티션을 쓰기 가능 모드로 다시 마운트
- 커널이 /sbin/init을 실행
- 커널은 init으로 제어권을 넘김

- 프로세스 기초
- 시스템에서 실행 중인 프로그램

- 프로세스 A가 프로세스 B를 실행시킬 수 있음

A : 부모 프로세스, B : 자식 프로세스

- PID(Process ID) : 프로세스 고유의 번호
- PPID(Parent Process ID) : 부모 프로세스의 고유 번호

- init
- init은 제일 먼저 실행되는 프로세스이고 PID = 1
- 모든 프로세스의 최상위 부모 프로세스

- init
- init 관련 파일
  - /etc/rc.d/rc.sysinit : 제일 먼저 실행되는 파일
  - /etc/inittab : init 설정 파일로 초기 런레벨 설정
  - /etc/rc.d./rc runlevel : 각 runlevel에 해당하는 스크립트를 실행시키는 파일
  - /etc/rc.d/rc[0-6].d/* : 각 runlevel의 데몬을 실행시키기 위한 파일들 (symbolic link)
  - /etc/rc.d/init.d/* : 데몬을 실행시키기 위한 실제 스크립트들
  - /etc/rc.d/rc.local : 제일 마지막에 실행되는 파일

- init과 runlevel
- init은 /etc/inittab 설정 파일을 읽음
- 기본 값으로 Runlevel이 5로 설정되어 있음
- Runlevel의 의미

0 : 시스템 종료,
1 : 단일 사용자 모드
2 : NFS를 지원하지 않는 다중 사용자 모드
3 : 모든 네트워크 기능을 지원하는 다중 사용자 모드
5 : X Window로 로그인 모드
6 : 재부팅

- Shutdown 과정
- init에 의해 runlevel 0 실행
- runlevel 0에서의 중요 동작

-동작 중인 데몬들을 모두 종료시킴
-프로세스를 종료 시 시간 간격을 두고 TERM, KILL 시그널을 보냄 (signal : 프로세스에게 보내는 신호)
-디스크를 언마운트(unmount)
-커널 동작을 멈춤

- 리눅스 시스템의 시작
- X Window로 로그인 모드 실행
- 사용자 이름과 암호를 입력

• 명령 행 환경 실행
-[프로그램]-[시스템 도구]-[터미널]
-프롬프트 모양으로 root 권한은 #, 일반 사용자 권한은 $
• root 계정은 시스템 관리자에게 부여되는 모든 권한을 가짐
시스템 설정 시에만 root 계정 사용
• 터미날 창이 뜸

● 접속 종료
• 계정의 사용을 마치고 빠져 나오는 것
• X Window : [시스템]-[로그아웃]
• Shell로 로그인 한 경우
'exit' 명령 또는 'logout'

● 리눅스 시스템의 종료
• X Window : [시스템]-[끄기]
• Shell 명령으로 'init 0' 또는 'shutdown' 명령

● 리눅스 시스템의 종료
• shutdown 명령 : 시스템 관리자(root) 권한만 실행 가능
• 시스템 종료는 다음 순서로 진행
-사용자들에게 시스템 종료메시지를 보내고 로그인을 제한
-지정된 시간 내 종료되지 않은 프로세스를 강제 종료
-지정된 시간 내 로그아웃하지 않은 사용자를 강제 로그아웃
-메모리에 데이터를 디스크에 저장 (sync)
-시스템 종료와 관련된 정보를 시스템 로그 파일에 기록
-마운트 되어 있는 디바이스들을 마운트 해제(unmount)
-시스템 종료

● shutdown 명령
• 사용법 : shutdown [-krhfc][-t sec] time [경고메시지]
◦ -t sec : 프로세스에 경고 보냄과 kill 시그널 사이 초 단위 간격
◦ -k : 사용자에게 경고 메시지만 전달
◦ -r : 시스템 종료 후에 재부팅
◦ -h : shutdown 후에 시스템 정지(halt)
◦ -f : 재부팅할 때 fsck(file system check)를 하지 않음
◦ -c : shutdown명령어를 취소한다.
◦ time : +m(몇 분 후) 또는 hh:mm(지정 시각)에 시스템 종료
◦ 경고메시지 : 사용자에게 보내지는 메시지

● 리눅스 명령의 특징
• 대소문자 구분, 여러 명령을 한 줄에 실행 가능

- ';'로 명령어 구분
- 파일 사용권한으로 root 외의 사용자는 권한이 있는 파일(디렉터리)에만 접근 가능
- 내부 명령(shell 명령)과 외부 명령(실행 파일)

-대부분의 명령이 파일로 존재, 실행 파일의 확장자 없음
-PATH 환경변수에 명령어의 위치가 설정되어 있어야 함

- 명령어 형식은 command [options] [parameters]

- date

- 서버의 날짜와 시간을 확인하거나 설정 (root 만 실행)
- 사용법 : date MMDDhhmm[[CC]YY][.ss]
  - MM : 월
  - DD : 월 중 일
  - hh : 시
  - mm : 분
  - CC : 연도의 처음 두 숫자 (선택적)
  - YY : 연도의 나중 두 숫자 (선택적)
  - ss : 초 (선택적)

- who

- 현재 시스템에 접속해 있는 사용자들을 조회
- 사용자명, 터미널 정보, 로그인 시간 등을 표시
- 사용 예
  - who [-Hq] (H는 각 열의 제목 보이기, q는 로그인 사용자 수, 기타 여러 가지 옵션이 있음)
  - who am i (자신의 정보 조회)
  - whoami (자신의 사용자 이름 조회)

- cal

- 달력을 출력
- 사용법 : cal [-3mjy] [[month] year]
  - 3 : 해당 달의 이전 달부터 다음 달까지 세 달의 달력 출력
  - m : 요일의 시작을 월요일로
  - j : Julian 일자 형식으로 출력 (1월 1일이 1일, ... , 12월 31일이 365일)
  - y : 해당 연도의 전체 달력 출력

- man

- 커맨드에 대한 온라인 설명서(manual)
- 매뉴얼의 구성

-명령어의 이름(Name)
-개요(Synopsis)와 설명(Description)
-모든 옵션의 목록과 설명

-환경(Environment)과 매개변수(Parameter) 등
- 매뉴얼 사용법
  - man [명령어] (man 페이지의 사용)
  - 명령어 --help (help의 사용)

● 파일 시스템
운영체제가 디스크(파티션) 상에 파일들을 구성하는 방식

● 리눅스 파일 시스템의 구조
- super block : 파일 시스템에 대한 정보
- inode block과 inode

파일에 대한 모든 정보(파일 이름은 제외)로 각 inode는 파일의 형태, 소유자, 크기, 링크 카운트, data block 번호를 포함
- directory block : 파일 이름과 inode 번호를 저장
- data block : 파일의 실제 데이터 저장

● 리눅스에서 관리하는 파일의 종류
- 정규 파일

일반적인 텍스트 파일이나 이진 파일
- 디렉터리 파일

-특별한 형식으로 디스크에 저장
-다른 파일들을 조직화하고 사용하는데 필요한 정보를 유지
- 특수 파일

-프린터, CD-ROM 드라이브, 디스크와 같은 주변 장치
-프로세스 간 상호통신에 사용되는 파일

● 리눅스가 지원하는 파일 시스템의 종류
- minix : Minix의 파일 시스템, 기본적인 파일 시스템
- xiafs : minix의 제한을 보완한 수정 버전
- msdos : FAT(File Allocation Table) 파일 시스템
- umsdos : msdos 파일 시스템을 긴 파일명과 소유자, 접근 허가, 링크와 장치 파일 등을 사용할 수 있도록 확장
- isofs : ISO 기준을 따르는 표준 CD-ROM의 파일 시스템
- hpfs : OS/2의 파일 시스템, 읽기 전용
- nfs (Network File System) : 네트워크상의 컴퓨터가 파일을 공유
- sysv : System V/386, Xenix, Coherent 파일 시스템
- ext : 리눅스 초기의 파일 시스템
- ext2 : 리눅스의 기본 파일 시스템으로 사용되었음
- ext3 : ext2를 Journaling을 지원하도록 확장

-현재 리눅스에서 가장 많이 사용
-Journaling : 변경을 기록하는 로그(journal)를 두어 시스템 비정상 종료 시 파일 시스템 복

구를 쉽게 하는 방법
- ext4 : ext3의 확장

● 리눅스 파일 시스템의 특징
- 파일 이름 길이는 255자까지
- 파일의 확장자는 필요 없으나 파일의 특성을 알리기 위해 확장자 사용 가능
  - 예 : *.c, *.java 등
- .으로 시작하는 파일은 숨겨진 파일
  - ls -a 명령으로 볼 수 있음
  - .은 현재 디렉토리, ..는 부모 디렉토리
- 파일 시스템의 크기는 최대 2TiB~16TiB (변동 가능)
- 파일의 크기는 최대 16GiB~2TiB (변동 가능)

● 다음 문제의 정답을 고르시오.

11. 리눅스 파일 시스템의 설명으로 옳지 않은 것은?
① 유닉스의 네트워크 파일 시스템인 NFS를 지원한다.
② 과거 MS-DOS의 파일 시스템인 FAT16을 지원한다.
③ 윈도즈의 파일 시스템인 NTFS는 지원하지 않는다.
④ CD-ROM 드라이브의 파일 시스템인 ISO9660을 지원한다.

12. 다음 중 리눅스가 지원하는 파일 시스템이 아닌 것은?
① tcsh ② ext ③ ext2 ④ minix

13. 리눅스 파일시스템의 구조에 대한 설명으로 올바르지 않은 것은?
① Super block : 파일시스템의 전체적인 정보를 가지고 있다.
② i-node : 파일 이름을 포함한 파일의 모든 정보를 가지고 있다.
③ Data block : 파일에서 데이터를 저장하는 공간이다.
④ Indirection block : inode에 Data Block의 위치 정보를 저장할 공간이 부족할 때 사용된다.

14. 다음 중 리눅스의 파일명 부여 방식이 아닌 것은?
① 파일명 중간에 공백을 포함할 수 없다.
② 대소문자가 구별된다.
③ 파일명 최대 길이가 128개까지로 제한되어 있다.
④ 윈도즈와 다르게 확장자의 의미가 별로 없다.

15. 다음 중 리눅스 명령어의 도움말을 볼 때 사용되는 명령이 아닌 것은?
① man ② help ③ halt ④ info

● 다음 문제에 대한 정답을 서술하시오.

16. 현재부터 20분 후에 'This System is going down'이라는 메시지를 전달하고 리부팅

하는 명령은?
(답) shutdown +20 “This System is going down”

17. 오늘은 2014년 4월 24일이다. 오늘부터 대입을 위한 수능 평가가 꼭 200일 남았다고 한다. 언제가 수능 평가일인지 알아보고자 한다면 어떠한 명령을 이용하여야 하는가?(옵션까지 정확하게 적용하시오)
(답) $date -d ‘200 days’