리눅스 마스터 정리

**리눅스란 ?**

Multi-User, Multitasking 시스템, 대부분의 응용프로그램 소스 공개. 안정성과 보안성이 높고 다양한 배포판이 존재(우분투, 슬랙웨어, 데비안, 레드햇, 수세), FSF(Free Software Foundation, 자유 소프트웨어)

**리눅스의 특징**

* 계층적인 파일 구조

: /(root) 를 기준으로 하위 디렉터리를 배치하는 트리 구조

* 장치의 파일화

: 장치들(키보드 하드디스크 모니터 등)을 모두 파일화 하여 사용한다 (/dev 디렉터리에 있음)

* 가상 메모리 사용

: 하드디스크의 일부를 메모리처럼 사용되는 것, 이 영역을 swap이라고 부른다. 메모리와 하드디스크 사이의 데이터 교환을 swapping이라고 한다 (free : 스왑 용량 확인 명령어)

* 동적 라이브러리 지원

: 라이브러리 = 프로그램에서 특정한 기능을 하는 루틴을 모아 놓은 것

공유 라이브러리 = 같은 루틴들을 공유하는 것

동적 공유 라이브러리는 실행파일 내부에 넣어두지 않고 프로그램을 실행 때 가져다 사용한다

* 가상 콘솔

: 하나의 모니터를 장착한 시스템에서 여러 개의 가상화면을 사용하게 하는 기능

기본적으로 6개 가상 콘솔 제공 (ctrl + alt + f1~f6)

* 파이프 (기호 : | )

: 어떤 프로세스의 표준 출력이 다른 프로세스의 표준 입력으로 쓰이게 하는 것,

여러 명령을 조합할 때 사용, 명령어와 명령어 사이에 | 기호를 넣는다

* 리다이렉션 (기호 : >, >>, <, <<)

: 프로세스의 입/출력 표준을 표준 입출력이 아닌 다른 입출력으로 변경할 때 사용,

출력 결과를 파일로 저장 or 파일의 내용을 프로세스의 입력으로 사용하는 기법

**리눅스 디렉터리 종류**

* / : root 디렉터리 (최상위 디렉터리)
* /bin : 모든 사용자사 실행 가능한 명령어들이 있다 (PATH 설정, cp mkdir mv, cat 등이 있다)
* /boot : 부팅 이미지 파일, 커널, grub 등 시스템 부팅 시 필요한 파일이 들어있다
* /dev : cd-rom, 하드디스크, 터미널 등 실제로 존재하는 물리적인 장치들을 파일화 하여 관리하

는 디렉터리

* /etc : 시스템 환경 설정 파일 및 부팅과 관련된 스크립트 파일이 들어있다
* /home : 계정 사용자들이 파일이나 디렉터리를 만들어 사용할 수 있다. 다수의 사용자 기반의

서버 운영하는 경우엔 반드시 할당, 한 명에게 할당한 용량에 따라 결정하면 된다

* /lib : 각종 라이브러리 저장, 커널 모듈도 있다
* /lost+found : fsck 명령어를 이용해 파일 시스템을 복구할 때 작업하는 디렉터리
* /mnt : 마운트 할 때 포인터가 되는 디렉터리
* /misc : 자동 마운트 프로그램인 autofs에 의해 사용되는 디렉터리
* /opt : 응용 프로그램의 설치를 위해 사용되는 디렉터리
* /proc : 가상파일 시스템, 시스템에서 운영되고 있는 다양한 프로세스의 정보를 담고 있다.
* /root : root 사용자의 홈 디렉터리 (root 사용자 = 시스템 관리자)
* /sbin : 시스템 관리에 필요한 명령어들이 들어있는 디렉터리, root사용자가 가용
* /tmp : 임시저장 디렉터리, 모든 사용자 접근 가능(공유모드) 이므로 보안 취약해 보안의 강화

를 위해 파티션 분할이 요구된다

* /usr : 명령, 응용프로그램들이 들어있는 디렉터리
* /var : 시스템 운영 로그(log), 파일과 스풀링과 같은 가변적인 데이터를 보관한다. 메일 서버로

운영하면 도착한 메일이 이 디렉터리의 하위에 담긴다. 최근에는 yum을 사용할 경우 다운로드한 패키지들이 임시로 저장되는 용도로 사용되고 있으므로 여유 있게 설정

**리눅스와 라이선스**

**GPL(General Public License)**

: 리눅스는 GNU(GNU’s Not Unix) 프로젝트와 함께 발전해 왔기 때문에 FSF의 GPL을 따른다.

**CPL 다섯가지 의무**

1. 목적은 상관없지만 법으로 제한하는 행위는 할 수 없다
2. 실행 복사본은 프로그램 소스코드와 함께 판매 or 소스코드 무료 배포해야 한다.
3. 소스코드를 용도에 따라 변경할 수 있다
4. 변경된 컴퓨터 프로그램 역시 프로그램의 소스코드를 반드시 공개 배포해야 한다
5. 변경된 프로그램 역시 반드시 똑 같은 라이선스 GPL을 적용해야 한다.

**주요 라이선스**

* LGPL(Library/Lesser General Public License)

: 소프트웨어 라이브러리에 주로 사용되며, 자유 소프트웨어뿐만 아니라 독점 소프트웨어에도 사용 가능하다, LGPL 개발 후 GPL 라이선스 변경 (O), GPL을 LGPL로 변경 (X)

* BSD(Berkeley Software Distribution)

: 버클리 캘리포니아 대학에서 배포한 라이선스, 누구나 재작, 배포 할 수 있다. 수정본의 재배포는 의무사항이 아니므로 2차 파생물에 대한 원시 소스코드의 비공개를 허용, 상용 소프트웨어(코드 비공개)에서도 사용 가능

* Apache

: 누구나 아파치 소프트웨어를 다운받아 이용가능, 재배포 시 아파치 라이선스 2.0을 포함시키고 아파치 소프트웨어 재단에서 개발된 소프트웨어임을 명시해야함

* MPL(Mozilla Public License)

: 모질라 재단에서 규정한 라이선스, BSD/GPL 괴 혼합적 성격, 소스 코드 수정 시 공개 필수지만 MPL 과 다른 코드를 결합해 만든 경우 MPL 코드를 제외한 다른 코드는 공개 안 해도 된다.

* MIT(Massachusetts Institute of Technology)

: MIT 대학에서 개발한 라이선스, BSD 기반으로 BSD라이선스와 조건이 거의 유사하다. JQuery / X window System / Node.js / Ruby on Ralis 등이 적용하였다.

**리눅스 설치**

**하드웨어 정보 파악**

* CPU

: 인텔 계열 CPU, AMD, 모토롤라, SPARC, Alpha, PowerPC, ARM, MIPS 지원

* 메모리(RAM)

: 메모리의 용량은 스왑 파티션 설정과 관련 있기 때문에 꼭 확인하기

* 하드디스크 드라이브

: 대부분의 하드디스크 지원, USB,SSD 지원, 모두 파일 형식으로 인식하여 사용

**파티션**

: 하드디스크의 영역을 설정하는 것, 기술적인 의미로는 하나의 물리적 디스크를 여러 개의 논리적인 디스크로 분할 하는 것. 분할된 파티션은 하나의 독립적인 디스크로 간주되어 Block 과 파일시스템 구성 및 다양한 정책을 독립적으로 설정할 수 있다.

**파티션 분할의 장점**

1. 하나의 디스크에 여러 OS 사용 가능
2. 파티션 단위의 다양한 정책 설정 가능
3. 사용자가 원하는 파일 시스템 구성이 가능, 원하는 블록 크기 설정으로 디스크 공간의 효율성을 높일 수 있다
4. 특정 파티션이 손상되어도 다른 파티션의 자료는 보존되어 안정성을 높일 수 있다. 파티션 손상으로 부팅이 되지 않아도 다른 파티션의 데이터는 가져올 수 있음
5. 자료 이전 및 관리, 백업 용이, 마운트 포인트 변경을 통해 자료의 이전 및 관리가 쉽다, dd dump등 파티션 기반의 다양한 백업 도구 지원
6. 부팅이 빨라지고 파일 시스템의 점검시간을 단축할 수 있다. (/boot 영역의 분할, 특정 파티션만 점검 하도록 설정)
7. 특정 영역 데이터 증가에 따른 시스템 및 프로세스 중단 방지

+ 스왑 영역도 파티션으로 반드시 분할해야 한다.

**디스크와 장치명**

* /dev/hdx : IDE 인터페이스에 연결되는 장치

연결 위치 파일명

Primary Master /dev/had

Primary Slave /dev/hdb

Secondary Master /dev/hdc

Secondary Slave /dev/hdd

* /dev/sdx : SCSI(Small Computer System Interface), S-ATA, USB 메모리, SSD 방식의 하드 디스크

**파티션 종류**

1. 주 파티션 (Primary Partition)

부팅 가능한 파티션

하드디스크에 하나 이상 존재해야 함

하나의 디스크에 4까지 만들 수 있음

확장파티션 사용할 경우 실질적으로 사용 가능한 주 파티션은 3개

1. 확장 파티션 (Extend Partition)

하나의 디스크에 하나만 생성 가능

Primary Partition 영역 내에 대체해서 사용되므로 번호도 1~4가 할당된다

자료 저장을 위해 직접 사용하는 파티션은 아니고 Logical Partition을 사용하기 위해 선언

1. 논리파티션 (Logical Partition)

Extend Partition 안에 생성되는 파티션, 생성 개수 제한 X, 12개 이상 생성은 권장하지 않음

5번 이후에 번호가 붙여진다

**파티션과 장치명**

분할된 파티션은 디스크 장치 파일명 뒤 숫자가 붙여서 관리된다.

**형식** */dev/하드디스크유형 + 디스크 우선 순위 + 파티션번호*

* 일반 IDE 방식 디스크 : /dev/hda1
* SCSI, S-ATA, USB 메모리 : /dev/sda1, /dev/sdb1

**분할이 필요한 디렉터리**

: /, swap, /boot, /usr, /var, /home, /tmp

**분할된 파티션 확인**

*fdisk –l*

이 명령으로 파티션 분할 후 재부팅하지 않으면 반영이 되지 않는다. -> 재부팅 or partprobe 명령 사용

실제 반영되는 파티션 정보는 /proc/partitions 파일에서 확인

**Boot Manager**

: 부팅을 도와주는 역할을 하는 프로그램, 여러 OS가 있는 경우 선택 해 부팅하도록 도와준다.

부트 로더로도 부르며, MBR영역에 설치된다.

부트 매니저 프로그램 종류 : LILO(Linux Loader), GRUB(Grand Unified BootLoader)

LILO

: 리눅스용 부트 매니저 프로그램

특정 파일에 구애 받지 않고 플로피 디스크와 하드디스크를 이용한 부팅 지원

현재는 잘 사용하지 않는다

GRUB

: GNU 프로젝트에서 만든 부트로더

다양한 파일 시스템 지원, 부팅 시 커널 인자를 조정해 동적인 부팅 지원

메뉴인터페이스 방식과 명령행 인터페이스 둘 다 지원

현재 대부분의 배포판에서 사용

다양한 부팅 모드 지원

**부팅 모드 종류**

* [a] : grub.conf 에서 커널과 관련된 부분의 매개변수를 추가할 수 있게 해줌, 싱글모드 진입
* [e] : grub.conf에 등록된 부팅 목록의 모든 항목을 직접 편집
* [c] : 상호 대화식으로 직접 입력할 수 있는 모드, Bash Shelll 과 유사한 환경 마지막에 boot라 입력하면 부팅 됨

**로그인과 로그아웃**

**실행레벨**

**리눅스 운영체제의 부팅 과정**

: 컴퓨터 전원 on -> BIOS에서 하드웨어 검사 -> GRUB 부트로더 실행 -> 커널 로드 -> 파일 시스템 마운트 -> init(초기화 관련 프로세스) 실행 -> 시스템 부팅에 필요한 스크립트 rc.sysinit, rc, rc.local 실행   
-> 로그인 창 실행

**일반적인 실행레벨**

* 0 : 시스템 중지 (기본값으로 설정 X)
* 1 : 단일 사용자 모드로 로그인 과정 없이 root 사용자로 리눅스 시스템을 부팅
* 2 : 네트워크 사용하지 않는 다중 사용자 모드
* 3 : 네트워크를 지원하는 다중 사용자 모드, X window를 사용하지 않고 메모리를 효율적으로 이용하  
   려면 가장 추천하는 레벨
* 4 : 사용되지 않으며, 사용자가 정의해 사용할 수 있다
* 5 : X 윈도를 사용하는 다중 사용자 모드, 최근 배포판 디폴드 설정 값이다
* 6 : 시스템 재부팅 시 호출되는 레벨

**로그인**

다수의 사용자가 사용하는 운영체제인 리눅스는 로그인 절차가 꼭 필요

접근 권한 + 사용 권한 획득하는 단계

로그인 메시지 관련 파일

* /etc/issue : 사용자가 로그인 할 때 “login:“ 메시지 전 출력되는 내용 적는 파일,   
   로컬 터미널로 접속시 출력된다
* /etc/issu.net : 위의 파일과 역할이 같으나 텔넷을 통한 네트워크 접속 시 출력된다
* /etc/motd : Message Of The Day 의 약어, 로그인 성공 시 출력될 메시지를 기록하는 파일

**로그아웃**

* 터미널 : logout 혹은 exit 명령어 입력 / ctrl + d, X-Window : 메뉴에서
* 자동 로그인 : (root 권한 상태) /etc/profile 에서 *TMOUT=초* 지정   
   (초 동안 작업하지 않으면 자동 로그아웃)