

Introduction to 2015 Wheel Seminar

2015. 7. 6

chocho

목차

- 목표
- 실습 안내
- 일정
- 벌금 및 정책

목표

월이 되고 싶어요!

목표

- **서버 관리에 필요한 전반적인 지식을 익혀두자!**

- ↳ ~~우리에게 구글갓이 있긴 하지만,~~

- 어떤 상황에서 어디서 무엇을 해야 할 지는 알아야

- 찾아가면서 작업을 진행할 수 있다.

- 따라서 스팍스 서버 안의 여러 파트가

- 각각 어떻게 구성되어 있으며, 어떤 기능을 수행하는지

- 파악하고 있어야 한다.

목표

- **실습을 통해 실전에서의 삽질을 예방하자!**

- 이론을 써먹어보자!

이론만 가지고는 실제 상황에 대비할 수 없다.

- 직접 손으로 해보고 눈으로 봐야 기억에 오래 남는다.

- 실습 때 삽질을 많이 해보고

실전에서 하지 않도록 조심해야 한다.

실습 안내

삽질은 어디서 하나요?

실습서버

- 2인 1조로 사용하게 될 실습서버
- 세미나 준비를 위해 이것저것 해볼 때 이 서버를 사용하세요!
- sparcs.org 나 bit.sparcs.org 등등 다른 서버에서 실습하지 말아요!ㅠㅠ

실습서버

조	도메인	IP
1	wseminar1.sparcs.org	143.248.234.108
2	wseminar2.sparcs.org	143.248.234.120
3	wseminar3.sparcs.org	143.248.234.211
4	wseminar4.sparcs.org	143.248.234.232

일정

3주간의 빡빡한 일정은 어떻게 되나요?

날짜	주제	담당자	완성 기한	조 배정
	1주 (7/6 ~ 7/10)			1 성원
7/6 월	월 세미나 Intro, 리눅스 개요, 디렉토리 구조	힐장	7/3 금	2 종훈 - 문영
	리눅스 명령어, 패키지	2		3 정운 - 형준
7/7 화	셸스크립트, cron	3	7/4 토	4 준혁 - 강원
7/8 수	Database	4	7/5 일	
7/9 목	파일시스템, 스왑영역	3	7/6 월	
7/10 금	네트워크, TCP/IP	4	7/7 화	
	2주 (7/12 ~ 7/16)			
7/12 일	DNS, BIND	4	7/9 목	
7/13 월	Web, Apache, nginx	1	7/10 금	
7/14 화	메일서버 - postfix, exim, sendmail	2	7/11 토	
7/15 수	NFS, FTP	3	7/12 일	
7/16 목	LDAP	4	7/13 월	
	3주 (7/19 ~ 7/24)			
7/19 일	가상화, Xen	2	7/16 목	
7/20 월	클라우드, AWS	전힐장	7/17 금	
7/21 화	Docker	1	7/18 토	
7/22 수	하드웨어	3	7/19 일	
7/23 목	보안	1	7/20 월	
7/24 금	백업 및 비상상황 대비	2	7/21 화	
	월 인수인계 세미나	힐장		

일정

- 모두에게 편한 최적의 세미나 시간을 정해봅시다.

벌금 및 정책

빡빡한 스케줄을 잘 지켜봅시다 ㅎㅎ

벌금 및 정책

- 지각
 - 지각 시 1분당 1000원. 최대 10000원.
- 세미나 자료 완성:
 - 세미나 3일 전까지 완성하여 wheel-seminar@sparcs.org 로 메일링.
 - ex) 7월 6일 세미나 자료는 7월 3일 11:55PM 까지 메일링.
 - 지연 시 하루당 5000원.
- 세미나 자료 업로드:
 - 세미나가 있었던 주의 일요일 밤 11:55PM까지 업로드.
 - 지연 시 하루당 10000원.

벌금 및 정책

- 모인 벌금으로는 무엇을 하나요?
 - 월 세미나가 모두 끝나고 난 뒤 성대한 회식을 합니다!!
 - (작년에는 베스타 갔대요~~)

휠 세미나 이수 요건

- 전체 세미나 중 4회 이하로 결석
- 2주차 세미나 5회 중 2회 이하로 결석
- 두 조건을 모두 만족해야만 휠이 될 수 있습니다.

이제
본격적으로
시작해봅시다.

리눅스 개요 및 디렉토리 구조

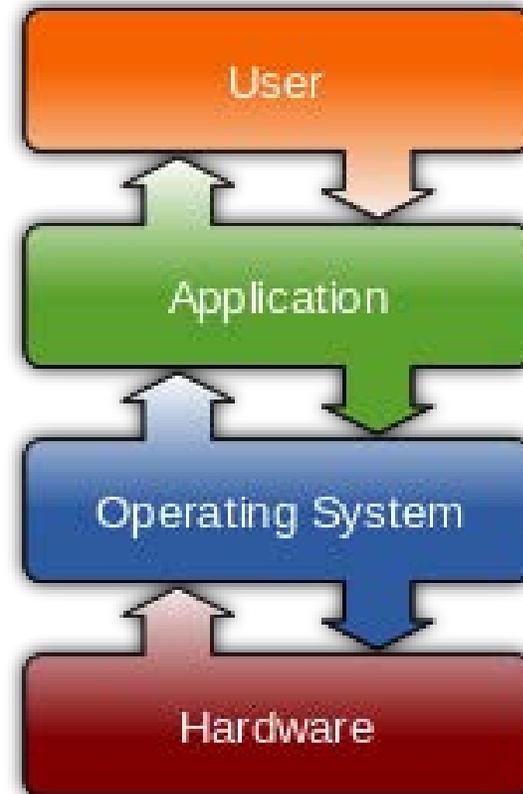
가장 기초적인 것들부터 알아보시다!

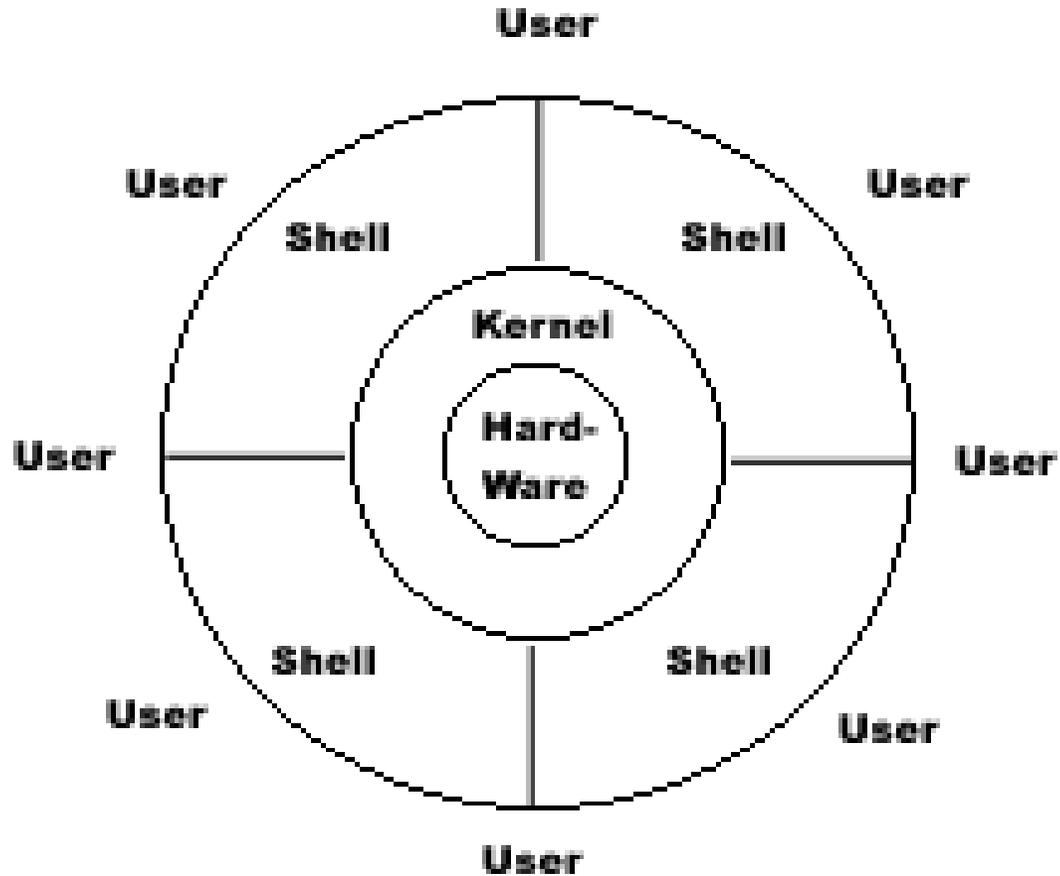
LINUX

- UNIX-like 운영체제 (OS)
 - UNIX: 다중 사용자/다중 태스크의 실행을 지원하는 소프트웨어 개발용 대화형 운영 체제
- 오픈소스 + 무료
 - 전세계적으로 약 5백만 명이 넘는 프로그램 개발자 그룹을 형성
 - 다수의 개발자가 함께 소스를 공유하고 개발에 참여 -> 지속적인 업그레이드
- LINUX distributions: Ubuntu, Debian, CentOS, Android, etc.
 - 현재 스팍스 서버는 Debian을 사용 중

Operating System

- 사용자/어플리케이션과 하드웨어 사이의 매개체 역할을 하는 소프트웨어
- 컴퓨터시스템의 하드웨어와 소프트웨어 자원 (resources)을 관리
 - 자원: 프로세서, 메모리, I/O 디바이스 등등
 - 자원을 다중 사용자 간에 효율적으로 할당, 관리 보호
 - 사용자 프로그램의 오류나 잘못된 자원 사용 감시
 - 입출력장치 등의 자원에 대한 연산과 제어





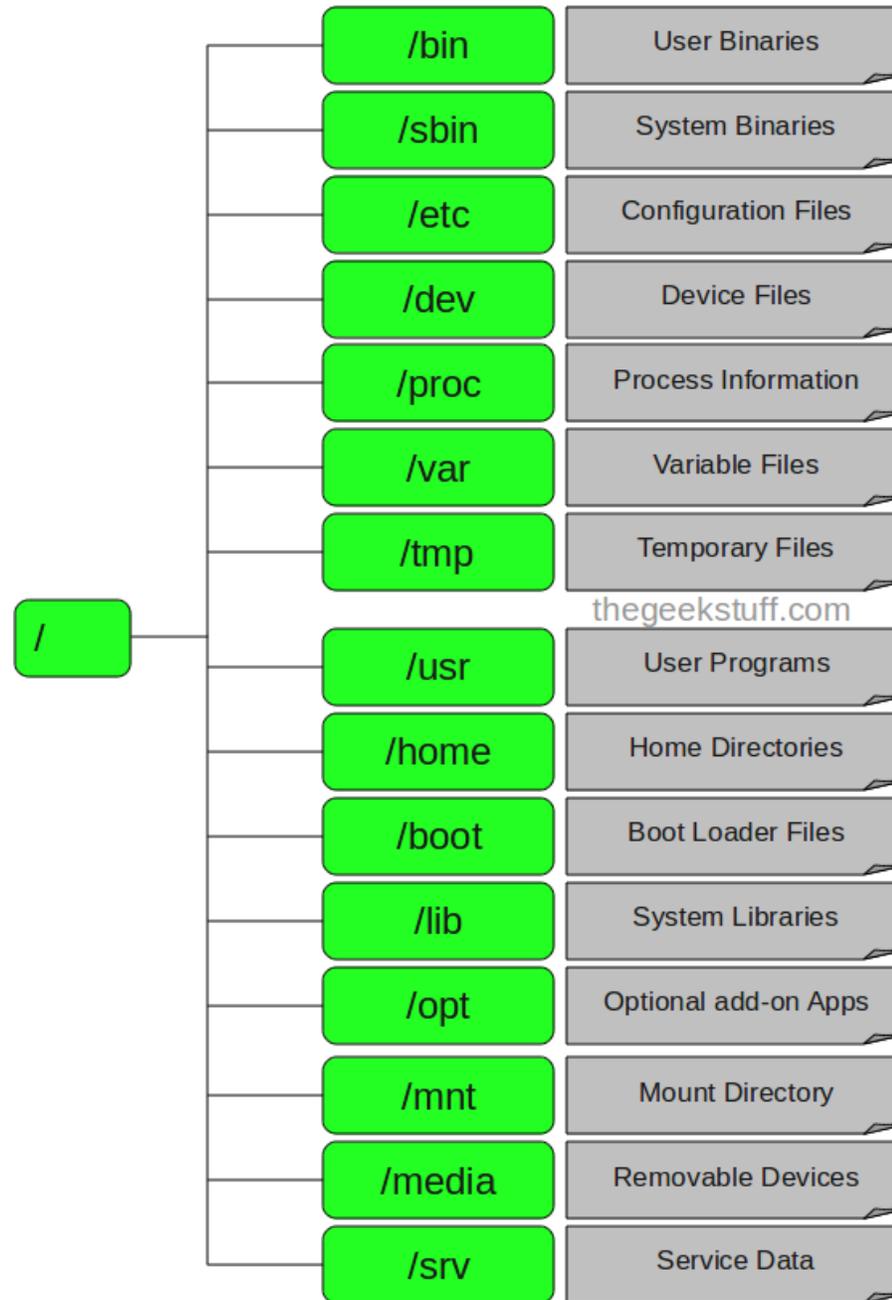
- Kernel 커널
 - = 씨앗, 핵
 - 운영체제의 핵심 부분
 - 프로세스, 메모리, 파일 시스템, 입출력장치, 네트워크 등 많은 것을 관리
 - 커널을 유지하기 위한 코드는 지속적으로 사용되기 때문에, 보통 커널은 나머지 부분에 의해 덮어쓰워져 훼손되지 않도록, 보호된 메모리 영역에 적재된다
- Shell 셸
 - = 껍질, 껍데기
 - 커널을 둘러싼 껍데기. Interface to Kernel
 - 사용자와 운영체제 사이의 층!
 - 유저로부터 명령(input)을 받아 해석(interpret)하여 커널에서 시스템 기능을 수행(output)

운영체제의 구조

리눅스의 디렉토리 구조

리눅스의 디렉토리 구조는 최상위 디렉토리 혹은 루트 디렉토리인 / 로 부터 뻗어나가는 트리 구조로 되어있다.

상위 디렉토리의 역할을 하나하나 살펴보자.



/ - Root

- 리눅스 시스템의 최상위 디렉토리
- root user 만 write 권한을 가진다.
- * /root 는 root user의 홈 디렉토리로 / 와는 다르다.

/bin – User Binaries

- 모든 사용자들이 사용하는 기본적인 리눅스 명령어가 위치
- 예: cat, cp, echo, mv, ps, ping, rm 등등

/sbin – System Binaries

- 주로 시스템 관리자가 이용하는 명령어들이 위치
- 예: iptables, reboot, shutdown, ifconfig

/etc – Configuration Files

- 서버 내 설치된 모든 프로그램의 시스템 설정 파일 configuration file 이 위치
- 시스템이나 사용자 관리를 위한 자료 파일 관리자용 명령어도 위치

/dev – Device Files

- 하드디스크, CD-ROM 등의 장치 파일이 위치
- 예: /dev/sda, /dev/cdrom
- /dev/null
 - “블랙홀”과 같은 파일
 - 주로 어떠한 작업이 출력하는 내용을 보고 싶지 않을 때 이곳으로 redirect
 - 파일을 지우지 않고 내용만 지움 (e.g. 로그 파일을 지울 때)
- /dev/zero
 - 무수히 많은 0으로 이루어진 파일
 - 파일을 비울 때, 비우면서 무한한 0으로 채워 무한대로 파일 사이즈가 늘어남
- /dev/urandom
 - random bits를 생성하는 파일

Try!

```
$ vi test.log
```

```
$ cat /dev/null > test.log
```

```
$ cat test.log
```

```
$ cat /dev/zero > test.log
```

```
$ cat test.log
```

```
$ cat /dev/urandom > test.log
```

```
$ cat test.log
```

/proc – Process Information

- 프로세스와 시스템 정보를 제공하기 위한 목적으로 설계된 가상 파일 시스템을 사용하는 디렉토리
- 실행 중인 프로세스, 사용 중인 디바이스, 커널이 수집한 하드웨어 정보가 저장.
 - /proc/meminfo – 메모리 사용 현황
 - /proc/cpuinfo – CPU 사용 현황
 - /proc/stat – 시스템 상태
 - /proc/uptime – 부팅 후 흐른 시간
 - /proc/version – 실행되고 있는 커널 버전
- 각 프로세스는 고유의 식별자(PID)를 가진다.
 - 문자 디렉토리/파일: 시스템 및 커널 정보
 - 숫자 디렉토리: 현재 실행 중인 프로세스의 정보

Exercise. 메모리와 CPU 사용 현황을 직접 확인해보자!

```
$ cat /proc/meminfo  
$ cat /proc/cpuinfo
```

/var – Variable Files

- 내용이 자주 변경되는 variable file이 저장
 - /var/log
 - 시스템 로그 파일
 - /var/spool
 - 메일, 프린트 등을 위한 각종 스푼 파일
 - /var/lib
 - 시스템 운용 시 계속 바뀌는 파일
 - /var/tmp
 - /tmp보다 좀 더 오래 유지되어야 하는 파일
 - 부팅시 초기화되지 않음
 - /var/run
 - 시스템 정보를 저장
 - 부팅시 초기화

/tmp – Temporary Files

- 임시 파일을 저장
- 시스템 종료 시 모든 파일이 삭제됨

/usr – User Programs

- 일반 사용자들을 위한 대부분의 프로그램이 위치
- 프로그램을 설치할 때 패키지에 포함된 대부분의 파일이 이곳에 설치
- 윈도우의 Programs Files 폴더와 유사
- /usr/src: 시스템에 빌드하는 프로그램의 소스
- /usr/share: 매뉴얼 등의 자료
- /usr/local: 잡다한 것들
- /bin, /usr/bin, /usr/local/bin의 차이
 - /bin: 리눅스를 돌리기 위한 최소한의 것들
 - /usr/bin: “Distribution-managed”. 각 리눅스 배포판의 패키지 관리자에 의해 관리
 - /usr/local/bin: “Not distribution-managed”. 배포판의 패키지 관리자에 의해 관리 X (locally compiled packages)

/home – Home Directories

- 기본적으로, 모든 유저들의 홈 디렉토리가 위치
- `adduser` 라는 명령어로 새로운 사용자를 생성하면 `user ID`와 동일한 이름의 디렉토리가 자동으로 생성됨
- 예: `/home/chocho`, `/home/suckzoo`

/boot – Boot Loader Files

- 부팅에 필수적인 파일들이 위치
 - 부트로더, 커널이 위치

/lib – System Libraries

- 커널이 필요로 하는 커널 모듈 파일들과 프로그램(C, C++ 등)에 필요한 각종 라이브러리 파일들이 위치
- 대부분의 라이브러리들은 링크로 연결되어 있다

/opt – Optional add-on Applications

- 기본적으로 설치되는 프로그램을 제외한 소프트웨어들과 add-on들이 설치되는 디렉토리

/media – Removable Media Devices

- DVD, CD-ROM, USB 메모리 스틱 등과 같은 탈부착이 가능한 외부 저장매체를 mount하여 하위 디렉토리인 것 같이 사용할 수 있는 디렉토리

References

- ‘2015 봄 신입생 세미나 – Linux, Vim, Toy’ – by chocho
- ‘2013 월세미나 리눅스 개요 및 디렉토리 구조 세미나’ – by suckzoo
- <http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>
- <http://blog.naver.com/koromoon/120106924291>
- <http://blog.naver.com/s1sho9676/220207770989>

끝