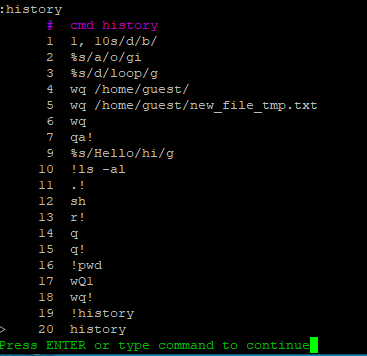
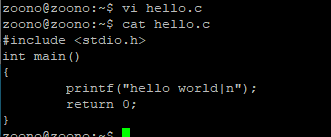
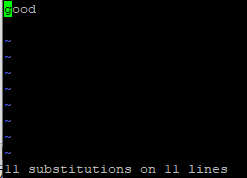
Vi 편집기

vi :q :q! :w :wq :wq! I, a esc yy(1줄) p(입력) yew(한문자) dw(문자삭제) dd(1줄삭제) u(취소)

!pwd !ls –al !cat newfile.txt sh (쉘 확인) ctrl + d :5,10s/a/b/ (5~10줄 a를 b로 변경) :%s/a/b/gi (전체 a, A를 b로 변경) :%s/new/old/g (전체 new를 old로 변경)



1. vi 에디터에서 입력모드로 사용 중 명령모드로 전환해야 할 경우 사용하는 키로 알맞은 것은?
   1. insert
   2. tab
   3. esc
   4. /
2. 첫 그림과 같은 hello.c 파일을 작성한다.
   1. 저장 및 종료 후 cat으로 내용 확인
   2. Hello.c를 열자. 이후 새 창을 새로 추가한다.
   3. 비쥬얼 모드 또는 마우스 모드 활성화
   4. 블록 선택 복사로 새 창으로 소스 코드 복사
   5. /home/아이디/hello2.c 파일로 저장하고 종료한다. 
3. 실행모드에서 외부 파일의 내용을 삽입하시오.
   1. 내용 삽입할 위치에 키보드 커서를 이동한다.
4. 둘째 그림과 같이 작성한 후, 문자를 치환하시오.
   1. bad를 모두 good으로 치환

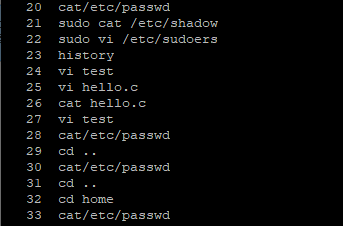


cat/etc/passwd

sudo 접근 권한 부여

su guest 또는 root (다시 계정 전환) sudo vi /etc/sudoers

sudo apt-get update cat /etc/apt/sources.list %s /kr.archive.ubuntu.com/mirror.kakao.com



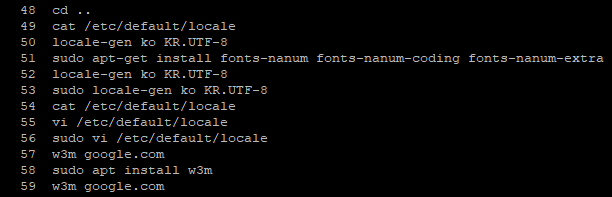
locale (지원 언어 확인) cat /etc/default/locale (언어 설정)

locale-gen ko\_KR.UTF-8 (기본 언어팩)

apt-get install fonts-nanum fonts-nanum-coding fonts-nanum-extra (추가 언어팩)

w3m google.com

timedatectl set-timezone Asia/Seoul date



1. 현재 기본 언어는 영문이다. 한글 출력에 설정해야 할 환경변수인 것은? (echo 명령어로 직접 변수 출력)
2. locale
3. lang = ko\_KR.UTF-8로 설정해야됨
4. language
5. local
6. SSH(Secure shell)의 설명으로 틀린 것은?
7. 원격복사(scp)를 지원한다.
8. 안전한 파일 전송(sftp)를 지원한다.
9. 기본 23번 포트를 사용한다. -> ssh는 기본적으로 22포트 사용함
10. rsh와 같이 원격 쉘을 지원한다.
11. 리눅스의 저장소(repository) 패키지 관리 기법으로 틀린 것은? (구글링 확인)
12. yast -> openSUSE 계열의 리눅스에서 사용하는 시스템 설정 도구
13. yum
14. zypper
15. apt-get
16. Ntp 서버의 세부 상태를 확인하는 명령어는? ( **ntpq -p** )

프로세스 확인

ps –ef (부모, 자식) ps -aux (상세 상태)

Init은 유저 레벨

Kthreadd는 커널 레벨

ps – ejH (트리 구조) ps –U root –u root (root의 모든 프로세스) ps -ex(실행 프로세스)

ps -ef | grep .py(프로세스 검색 : .py)

ps –p 프로세스 번호 –o comm= (Pid로 프로세스/사용자 이름 찾기)

ps –p 프로세스 번호 –o user (Pid로 프로세스/사용자 이름 찾기)

ps –C 프로세스 이름 –o pid(프로세스의 모든 pid 찾기)

프로세스 모니터링 예) – 부모, 관계 등

ps –fp 프로세스 번호 : Pid 및 ppid의 번호 정보

ps –C 프로세스 이름 : 상/하위 프로세스

프로세스를 실행하는 방법 2가지 : 포그라운드 / 백그라운드

프로세스의 상태 : 실행, 정지, 종료

cat /etc/login.defs | more -> 실행 유지, 다른 작업 하는 경우(정지)

cat /etc/login.defs -> 실행 바로 종료

ps –ex -> Stat 필드 상태 값 확인

cat /etc/login.defs | more & -> 백그라운드로 pid 번호 등록됨(1~x)

프로세스 확인 : jobs –l

프로세스 포그라운드로 전환 : fg %번호

프로세스 정지 : Ctrl + z

프로세스 종료 : kill -9 프로세스 번호, 실행중인 화면에서 Ctrl + c, Kill –l ( 전체 목록)

at 명령어를 활용한 예약 실행하기 : 일회성의 프로세스(명령어) 수행

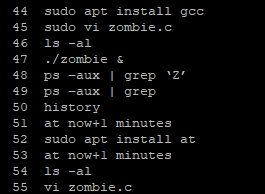
일반적인 사용 방식

현재 시간 기준 +

1. at 작성하기 : at now+1 (minutes, hours, days)
2. 실행하고 싶은 명령어를 작성 : ex) ls -al > test.txt
3. <EOT>
4. ctrl + d 단축키 (저장 후 종료)

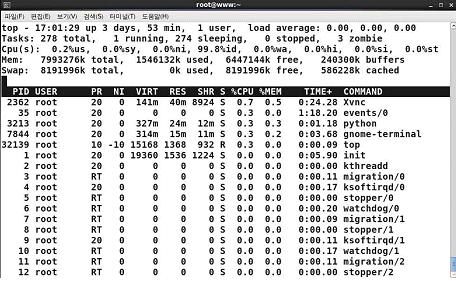
참고 : 특정 시간 실행도 가능.

at 14:00 2023-05-08 –f ./zombie\_test



1. 다음 중 프로세스에 관련된 설명으로 알맞은 것은?
2. 최초의 프로세스인 init은 PID 번호가 1이다.

* . 최초의 프로세스인 init은 PID 번호가 1이고, 이후에 생성되는 모든 프로세스들은 init의 자식이다. Ex) fork(프로세스 복사), exec(프로세스 변경)

1. 리눅스 부팅 시에 발생하는 프로세스는 exec 방식이다.
2. 리눅스는 다중 프로세스를 지원하지 않는다.
3. 새로운 프로세스를 만들 때 기존 프로세스를 일부 변경한다.
4. 다음 명령어는 어떤 유형인가? (find / -name -type d 2>/dev/null > dir.txt &)
   1. 포그라운드 프로세스 -> fg %번호
   2. 백그라운드 프로세스 -> 명령어 뒤에 &를 붙여서 백그라운드로 실행
   3. 좀비 프로세스
   4. 데몬 프로세스 -> d
5. 그림과 같이 실시간 프로세스를 출력하는 명령어는?
   1. ps -> 정적인 프로세스 상태
   2. pstree -> 트리구조
   3. jobs -> 백그라운드
   4. top
6. 서버의 주요 서비스 등록 처리를 수행하는 프로세스는?
   1. 포그라운드 프로세스
   2. 백그라운드 프로세스
   3. 좀비 프로세스
   4. 데몬 프로세스 -> 사용자의 요청에 따라 특정 기능이나 서비스를 제공하는 백그라운드 프로세스. Ex)  httpd, sshd, named

리눅스 패키지 관리

대표 명령어 종류

APT, DPKG : 우분투 및 데비안 계열

RPM, YUM : 센투스, 오라클, 페도라, 레드햇 계열

전체 패키지 목록 : apt list 또는 dpkg –l

설치된 S/W 목록 : apt --installed list (전체)

패키지 직접 검색 : apt-cache search 패키지 이름

패키지의 세부 정보 : apt-cache show 패키지 이름

의존성 : apt-cache depends 패키지 이름

패키지 업그레이드 : apt-get upgrade (전체 버전 업그레이드), apt-get upgrade 패키지 명 (특정)  
업데이트만 있는 패키지는? 업그레이드 여부를 묻는다

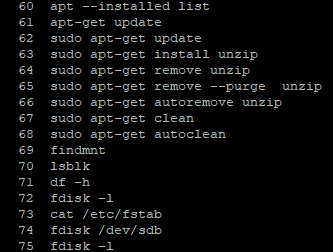
특정 패키지 설치 : apt-get install 패키지이름

소프트웨어 삭제하기 : apt-get remove 패키지이름

apt-get remove --purge 패키지이름 : 추가된 설정파일까지 모두 삭제

apt-get autoremove : 종속 패키지까지 전부 삭제

소프트웨어 정리하기 : apt-get clean (다운로드 받은 아카이브 파일을 모두 삭제 /var/cache/apt)

apt-get autoclean : 최신 버전 패키지만 남기고 아카이브 정리

전체 하드디스크 정보 : df –h 또는 lsblk 또는 fdisk –l

findmnt : 파일 시스템 및 세부 정보

디스크 별 역할과 의미

udev : /dev 장치 관리(장치 연결, 동작, 권한, 정책 등)

tmpfs : 임시 파일 시스템 (다수 존재, 휘발성(부팅 시 지워짐), 빠름)

/dev/loop번호 : 가상 장치 (실제 장치 x, 다른 스토리지와 맵핑)

sda : 실제 저장 장치 (Storage Drive a의 약자, 부팅 드라이브 Sda1은 sda의 첫번째 디스크가 된다.)

리눅스 부팅 시 저장장치 설정 : cat /etc/fstab

Uuid 전용 식별자 사용 : blkid /dev/장치명 -> 장치를 식별하는 고유 아이디

추가된 저장 장치 확인/실행 : fdisk –l fdisk /dev/sdb

mkfs를 통한 파티션 포맷 : mkfs.ext4 /dev/sdb1

mount를 통한 경로 연결하기 : mkdir /new\_disk -> mount /dev/sdb1 /new\_disk

Vi로 직접 fstab 에 하드디스크 등록 : vi /etc/fstab

Mdadm으로 레이드 그룹을 생성한다. : mdadm –create /dev/md/raid0 –level=0 –raid-devices=2 dev/sdb1 /dev/sdc1

mdadm –detail –scan

Raid0을 ext4로 포맷한다. : mkfs.ext4 /dev/md/raid0

사용할 폴더를 생성하고 마운트한다. : mkdir /new\_raid0 -> mount /dev/md/raid0 /new\_raid0/

CPU 과부하 테스트(프로세스 급격한 증가)

vi cpu\_test.sh (작성하기)

chmod 777 cpu\_test.sh (권한 풀기)

./cpu\_test.sh & (백그라운드 실행)

CPU 자원 확인 : uptime(시스템 부하 평균치 출력), mpstat(시스템 부하 세부 내용)

메모리 자원 확인 : vmstat, free(간략한 메모리 사용량 출력)

실시간 자원 확인 : top

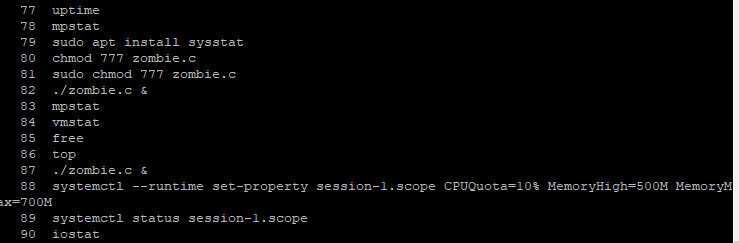
systemd-cgls

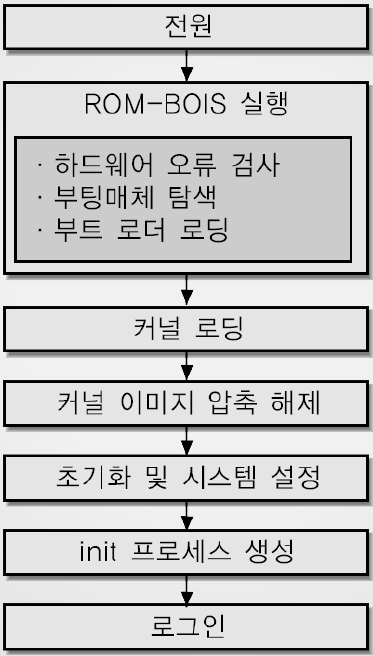
서비스 및 프로세스의 각 계층 관계

systemctl --runtime set-property session-1.scope CPUQuota=10% MemoryHigh=500M MemoryMax=700M : CPU제한 10프로, 메모리 제한 500메가, 700메가 이상부터는 자동 프로세스 KILL 런타임 옵션 : 부팅시 초기화됨(없으면 영구)

systemctl status session-1.scope

top –u guest : CPU 및 메모리 상태 확인

I/O 자원 확인 : iostat

리눅스 FS-캐싱

cpu, 메모리 대역폭 차이(L1 > L2 > L3 > Memory > Disk)

가용영역 🡪 캐시로 활용(점차 반환) 🡪 최후 /swap 영역 활용

Init 프로세스(0번 프로세스)

실행모드에 따라 런레벨 결정

0, 1, 6 기본

Server는 기본 3레벨이다.

os 설치때는 5 -> 3레벨 동작

Ps 명령어

-a : 모든 사용자, 모든 프로세스 출력

-e : 모든 프로세스 정보 출력

-l : Long Format 정보 출력

-u : 사용자 정보와 프로세스 시작 시간 등 출력

-x : 제어 터미널을 갖지 않는 프로세스 출력

-f : Full format 정보 출력

부모 -> 자식 관계(트리)

P TO P : 공유 메모리 통신 사용. 대부분 선점형 스케쥴링. 대표 : cfs(공평 시간 분배)

pstree –ah | more

동일한 자식 = 형제 프로세스 Sibiling process 라고 한다.

자식은 새로운 pid 부여

Ppid에 부모 id 저장

부팅 시 자동 실행 = 데몬 프로세스

좀비 프로세스란

실행 완료이후 반환되지 않은 프로세스 -> 메모리 공간에 남아 있어 리소스 낭비

RAID(Redundant Array of Inexpensive Disks) 기법이란?

디스크의 효율적 사용을 위한 저장방법.

종류 : 일반적 0, 1, 5~

0번 : STRIPE – 디스크 수 만큼 I/O

랜덤 접근

1번 : MIRROR – 디스크 중복 기록

최소 디스크 2개 이상

5번 : 패리티 정보 디스크 사용

PARITY : 오류 데이터를 재구축하는 계산 값

최소 디스크 3개 이상, 1개 고장나도 복구 가능

일반적 안전성은 1, 속도는 0번

웹 서비스 : 5번

FDISK 프로그램을 통해 설정 가능

운영체제가 하는일 : 중간, 메모리 중심

메모리 관리 : 공유, 보호, 확장, 최적화

계층구조와 주소 관계

계층 순서 : CPU -> 메인메모리 -> 저장장치

알려진 기법들

다수 프로세스 쌓다보면 메모리 빈 공간 발생 -> 메모리 공간을 작은 단위로 분할 할당 - 페이징

BUT, 랜덤 분포 및 순서 보장 X 문제

CPU가 MMU를 통해 별로 테이블 참조 – 페이지 테이블

성능 향상을 위해 캐시 메모리 – TLB 공유 메모리 : 자주 쓰는 LIB 공유, 프로세스들이 나눠씀

가상 메모리 : 메모리 공간 부족 – SWAP

서버 성능 저하 요인 = 복합적

CPU : 메모리 SWAP 활성화 : CPU 사용률 큰 하락

네트워크 : 트래픽 과부하 집중 : 서버 접속, 프로그램 지연, 과금 등

즉, CPU 및 메모리 등 자원의 제한 필요

스래싱(Thrashing)

발생원인 : 메모리 사용/접근 시에 메모리 페이지 부재 = 페이지 폴트(Page fault)

과도한 페이지 교체를 반복 수행

해결 및 예방 : 다중프로그래밍 정도(DOM) 줄이기, 하드 디스크 대신 빠른 SSD 사용, 메모리 늘리기  
CMS(Content Management System) : 워드 프레스(78% 이상), Wix 등

전세계 시장 점유율 압도적, 편리한 UI 방식과 화면 구성, 웹 사이트에 구축/운영에 필요한 모든 도구, 다양한 테마 템플릿 무료, 광범위한 확장 모듈

웹 서버 운영 = 리눅스 + APACHE2 + 워드프레스