**섹션 05.프로세스와 실행**

**01.컴퓨터의 구조**

Process

프로세서 (Processor): CPU, 프로세스와 다름.

저장장치 (Storage, 스토리지): SSD,HDD, 정보를 저장

비싸고 저장용량이 작다. 저장하고 읽는 속도가 빠르다.

메모리(Memory): RAM, DDR, 정보를 저장.

저렴하고 저장용량이 크다, 저장하고 읽는 속도가 느리다.

명령어는 디렉토리에 파일의 형태로 저장이 되어있다. 그 파일은 기본적으로 스토리지에 저장되어있는 것이다. 그것을 프로그램이라 부른다. 프로그램을 시작하면 메모리에 적재가 되고 cpu가 실행을 하는것이다. 실행되고 있는 상태의 프로그램을 프로세스라고 하고, 그 프로세스를 프로세서가 처리한다.

**02.프로세스 모니터링 (ps, top, htop)**

#ps :프로세스 리스트 보여줌

#ps aux :백그라운드에서 실행되는 모든 프로세스 보여줌

#ps aux | grep apache : apache라는 텍스트를 포함하는 프로세스만을 보여줌

#PID: 프로세스가 가지는 id, 식별자.

#sudo kill pid : 프로세스를 강제로 종료.

#sudo top

#sudo htop 정렬 기능 사용 가능.

**03.백그라운드 실행**

#multitasking :여러개의 일을 실행한다는 뜻

#background, forground

#Ctrl + z : 실행중인 프로그램을 백그라운드로 보내는 단축키, nano에서도 실행 가능.

이 기능을 실행하면 명령어가 일시 정지 됩니다.

#fg: 포어그라운드의 약자,

jobs에서 백그라운드로 물러나있다가 포어그라운드로 나오는 프로그램이 뭔지 보여줌. +는 1순위, -가 이순위.

#fg %2 : jobs에서 우선순위가 -인 프로그램을 실행

#kill %4 : jobs에서 번호가 4번인 프로그램 종료

%kill -9 %4 : : jobs에서 번호가 4번인 프로그램 강제적 종료

#jobs : 백그라운드에 돌아가는 프로세스 볼 수 있음

&가 명령어 뒤에 붙으면 명령어가 실행될 때 백그라운드로 실행됩니다.

#ex) ls -alR / > result.txt 2> error.log & : jobs : 백그라운드 작업들의 목록을 보여줍니다.

#실행시간이 오래걸리느 명렁의 끝에 &를 입력하면 백그라운드에서 실행되다가 종료시 Exit 1을 출력함

**04. 항상 실행 - 데몬의 개념**

daemon : ls, mkdir, rm 은 데몬이 아님.

ex)언제 사용될지 몰라서 늘 켜져있는 냉장고, 쓰지 않을때는 끄는 tv.

데몬이라는 형태의 프로그램들은 ex)server 늘 켜져있음.

언제나 실행되고 있어야 하는 프로그램

***-멀티태스킹 운영 체제에서 데몬은 사용자가 직접적으로 제어하지 않고, 백그라운드에서 돌면서 여러 작업을 하는 프로그램을 말한다***

***-윈도우의 서비스 같은 개념***

**05. 항상 실행 - service 와 자동실행**

#sudo apt-get install apache2

#etc/init.d : 데몬 프로그램들이 위치하는 디렉토리

이곳에 위치한 프로그램들을 키는건 우리가 아는것과 다름

#suido service apache2 start 라는 명령으로 시작함.

#ps aux : 프로세스 리스트 모두 출력

> 데몬 프로그램들은 대부분 #(sudo service apache2)+ start, stop이라는 2개의 명령어를 가지고 있음.

#cd.rc. +탭 : 디렉토리가 있음

rc3.d : 우리가 리눅스를 콘솔(CLI)로 구동하고 있는 경우 rc5.d: 리눅스를 GLI로 구동하고 있는경우 안으로 들어가서 #ls-l 을 해보면 .. : 부모디렉토리, 데몬들이 모여있는곳.

#리스트 s02apache2 -> ../init.d./apache2 에서

실제 프로그램은 ../init.d./apache2에 있으나, s02apache라고 링크를 건 것임

rc3.d에 링크를 걸면 데몬이 자동으로 실행되게 할 수 있음.

시작하는 단어가 s면 실행, k면 실행중지

**06. 정기적 실행 (cron) 1**

cron : 정기적으로 명령을 시켜주는 도구, 소프트웨어

정기적으로 해야하는 전송, 백업 등에 사용

#crontab -e : 정기적으로 실행시키고 싶은 어떤 작업이 있을 때 적으면 그대로 크론이 실행된다.

사용방법

#m h dom mon dow command

#m : 실행되는 주기

ex)10 > 매시간 10분 /10 > 10분에 한 번

#h : 시간

ex) \* : 시간과 상관없이 실행, 10 1: 1시10분에 실행,

#dom : day of month

ex) 24: 매달 24일에 실행

#mon: 달

#dow : 요일

구글에 crontab expression 검색



#date : 현재 시간을 알려주는 명령어

#date >> date.log :date의 값이 항상 date.log에 저장됨

*리눅스 쉘에서 >>는 파일에 데이터를 추가할 때 사용하는 리다이렉션 기호입니다. 일반적으로 이 기호는 파일이 이미 존재할 경우 기존 내용 끝에 추가하는 목적으로 사용됩니다.*

#tail date.log :제일 뒤쪽에 있는 텍스트 출력

#tail -f date.log : date.log 뒤에 텍스트가 추가되면 자동으로 화면에 리프레시

#2:표준에러

#1:표준출력

#2>1 : 에러가 있을 때 1에 출력됨

#2>&1 :표준에러를 표준출력으로 리다이렉션

**07. 정기적 실행 (cron) 2 - 사례**

ex)컴퓨터가 10만명의 사용자에 대해 가입메일을 보낸다. 한 명에 0.1초가 든다면, 꽤 긴 시간이 걸리게 된다.

전송이 완료될 때 까지 대기상태에 돌입한다.

이런 경우 사용자가 정보를 전송하면 saved라는 것만 어딘가에 기록해두고, 정보를 전송해준다. 그럼 사용자는 saved를 빠르게 확인할 수 있다.

crone이 정기적으로 서버에있는 saved정보를 확인하여, 추가된 정보에 대해 사용자와 상관없이 이메일을 보내는 작업을 백그라운드로 진행한다.

**08. 셸을 시작할 때 실행**

셸이 시작 됐을 때 어떤 특정한 명령어가 자동 시작되게 하는법

스타트업 설정 or 스타트업 스크리트 라고 함.

ex) alias(별명) l = 'ls -al' : l이라고 입력하면 ls-al이 실행되게 된다.

재미있는 alias들

https://www.cyberciti.biz/tips/bash-aliases-mac-centos-linux-unix.html

#echo $SHELL : /bin/bash

#cd~

#nano .bashrc : 프로그램이 시작될 때 자동으로 이 프로그램을 실행하게 되어있음, 여기있는걸 바꾸면 셸이 시작한 후에 내용을 변경 가능

#bash : bash가 실행이됨

#exit: 그 전 상태로 나와짐

프롬프트 정보의 형태를 바꾸기 가능.

$PATH의 값 변경 가능.