

서버 기술 기초 요약 - 리눅스 쉘 사용법 이해 (프로세스 관리)

본 강의는 도커 강의이므로,
도커 작업을 위해 꼭 필요한 핵심 기능을 익히기로 함

프로세스 vs 바이너리

- 코드 이미지 또는 바이너리: 실행파일
- 실행 중인 프로그램: 프로세스
 - 가상 메모리 및 물리 메모리 정보
 - 시스템 리소스 관련 정보
 - 스케줄링 단위

리눅스는 다양한 프로세스 실행 환경

- 리눅스는 기본적으로 다양한 프로세스가 실행됨
 - 유닉스 철학: 여러 프로그램이 서로 유기적으로 각자의 일을 수행하면서 전체 시스템이 동작하도록 하는 모델

foreground process / background process

- foreground process: 쉘(shell)에서 해당 프로세스 실행을 명령한 후, 해당 프로세스 수행 종료까지 사용자가 다른 입력을 하지 못하는 프로세스
- background process: 사용자 입력과 상관없이 실행되는 프로세스
 - 쉘(shell)에서 해당 프로세스 실행시, 맨 뒤에 & 를 붙여줌
 - 사용 예

```
# find / -name '*.py' > list.txt &
[1] 57
```

- [1] 은 작업 번호 (job number), 57 은 pid (process ID) 를 나타냄

foreground process 제어하기

- [CTRL] + c : 프로세스 작업 취소 (해당 프로세스는 완전히 종료됩니다.)

프로세스 상태 확인 - ps 명령어

- 사용법 : ps [option(s)]
- option(s)

-a : 시스템을 사용하는 모든 사용자의 프로세스 출력
(보통 aux 와 같이 u, x 옵션과 함께 사용)

-u : 프로세스 소유자에 대한 상세 정보 출력

-l : 프로세스 관련 상세 정보 출력

-x : 터미널에 로그인한 후 실행한 프로세스가 아닌 프로세스들도 출력함.
주로 데몬 프로세스(daemon process)까지 확인하기 위해 사용.
본래 ps 명령은 현재 쉘(shell)에서 실행한 프로세스들만
보여주기 때문에 이 옵션을 사용하는 경우가 많음.

-e : 해당 프로세스와 관련된 환경 변수 정보도 함께 출력

-f : 프로세스 간 관계 정보도 출력

- 데몬 프로세스(daemon process): daemon은 악마를 의미함. 사용자 모르게 시스템 관리를 위해 실행되는 프로세스로 보통 시스템이 부팅될 때 자동 실행 (예: ftpd, inetd 등)

프로세스 상태 확인 - ps 명령어

- 주요 ps 출력 정보 항목

USER : 프로세스를 실행시킨 사용자 ID

PID : 프로세스 ID

%CPU : 마지막 1분 동안 프로세스가 사용한 CPU시간의 백분율

%MEM : 마지막 1분 동안 프로세스가 사용한 메모리 백분율

VSZ : 프로세스가 사용하는 가상 메모리 크기

RSS : 프로세스에서 사용하는 실제 메모리 크기

STAT : 프로세스 상태

START : 프로세스가 시작된 시간

TIME : 현재까지 사용된 CPU 시간(분:초)

COMMAND : 명령어

프로세스 중지시키기

- kill 명령어
 - 사용법
 - a. kill % 작업 번호(job number)
 - b. kill 프로세스 ID(pid)
 - c. 작업 강제 종료 옵션 -9
 - 예

```
# find / -name '*.py' > list.txt &
[1] 57
# kill -9 57
```

주로 사용하는 프로세스 명령 (적어도 이 명령은 편하게 사용해야 함)

- ps aux | grep 프로세스명 : 프로세스가 실행 중인지를 확인하고, 관련 프로세스에 대한 정보 출력
- kill -9 프로세스 ID(pid) : 해당 프로세스를 강제로 죽임
- 명령 & : 터미널에서 다른 작업을 해야하거나, 프로세스 실행에 오랜 시간이 걸릴 경우 background 로 실행
- [CTRL] + c : 프로세스 종료(실행 취소)

모든 자료와 관련 영상 컨텐츠는 저작권법 제25조 2항에 의해 보호를 받습니다. 본 영상 및 자료/파일 일부 문구 등을 외부(git/블로그등)에 공개하거나, 요약해서 게시하지 말아주세요. 좋은 강의를 꾸준히 만들 수 있는 문화를 만들어갔으면 좋겠습니다.