

실습 4. 프로세스 관리

프로세스 개요

◆ 프로세스는 부모-자식 관계

- 부모가 자식 프로세스를 생성

◆ 프로세스 번호 (pid)

- 프로세스에 부여된 고유의 번호
- systemd 가 1번

◆ 프로세스 종류

- 데몬 프로세스 : 특정 서비스를 위해 커널이 구동(원격접속 등)
- 고아 프로세스 : 부모가 먼저 종료된 프로세스. 1번이 고아프로세스의 부모가 된다.
- 좀비 프로세스 : 부모가 자식의 종료를 제대로 처리하지 않은 프로세스(defunct). 실행되지 않지만 프로세스 목록에 표시되는 프로세스

프로세스 관리 명령

◆ ps :프로세스의 목록을 표시

– 사용 예) `ps` `ps -ef` (상세 정보)

- UID : 프로세스를 실행한 사용자ID PPID : 부모 프로세스 C : CPU 사용량
- STIME : 시작 시간 TIME : 실행 시간 CMD : 프로그램 이름(명령)

– 특정 프로세스 찾기

- 현재 특정 프로세스가 있는지 찾을 때
- `ps`로는 너무 많은 내용이 나와 원하는 결과를 눈으로 찾기 어렵다.
- `grep` 명령을 결합하여 사용
- `ps -ef | grep bash` (bash를 프로세스 목록에서 찾기)

프로세스 관리 명령

◆ 프로세스 종료하기

- 응답이 없는 프로세스나 불필요한 프로세스 강제 종료
- PID를 알아야 한다.
 - kill 1001 (PID가 1001인 프로세스 종료하기)
 - kill -9 1001 (강제 종료)
- 터미널을 하나 더 열어 1번 터미널에서는 실행을, 2번 터미널에서는 상태 확인 및 종료하기를 한다.
 - 터미널 1에서는 “man ps”, 터미널 2에서는 ps로 프로세스를 찾은 후 kill
- kill된 결과는 ps 로 확인한다.

리눅스는 백그라운드 프로세스가 많아 kill 할 일도 많다.

프로세스 관리 도구

◆ top

- 현재 실행중인 프로세스 정보를 주기적으로 출력
- 프로그램 표시 - 유틸리티 - 시스템 감시 (작업관리자와 같다)

포그라운드, 백그라운드

◆ foreground

- 사용자의 터미널에서 직접 실행중인 프로세스. 셸은 대기중이고 직접 구동중인 프로세스

◆ background

- 셸의 상태와 별개로 뒤에서 실행중인 프로세스
- 여러 개의 프로세스를 구동 시키려면 백그라운드가 필수

◆ 백그라운드로 실행하기

- 명령 뒤에 &를 붙인다.
- sleep 100 &
- find / -name passwd > pw.dat &

작업 전환

◆ job

- 백그라운드 작업을 모두 보여준다.

◆ 포그라운드/백그라운드 사이의 전환

- 포그라운드를 백그라운드로 전환할 때
 - Ctrl-Z 를 눌러 프로세스를 잠시 중단시킨 후, bg 명령으로 백그라운드로 전환한다.
- 백그라운드를 포그라운드로 전환할 때
 - fg 명령을 사용한다.

◆ 작업 종료 : Ctrl-C

◆ nohub

- 로그아웃 후에도 백그라운드 작업 계속하기
- nohub 명령 &

작업 예약

◆ 주기적인 반복 명령

- 백업, 로그를 메일 보내기, 알람하기

◆ 정해진 시간에 한번 실행 : at

- at -l : 대기중인 명령 list
- at -r : 대기중인 명령 삭제
- at 07:00 pm : 오후 7시에 명령 (추가할 프로세스는 이 명령 이후에 입력)

◆ 정해진 시간에 반복 실행 : crontab

- 사용법은 각자 찾아보자.

◆ 좀 더 쉬운 방법

- /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly 디렉토리에 쉘 스크립트를 만들어 놓는다.(관리자)

at이 없을 수 있다.
이 때에는 설치