Report

서버관리실무

**제목 : 11주차 과제**

**학과 : 컴퓨터소프트웨어과  
학번 : 2017100301  
이름 : 이상남  
 담당교수 : 김형철 교수님  
제출일 : 2020.11.09**

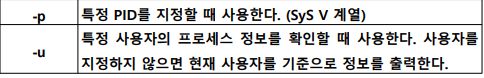
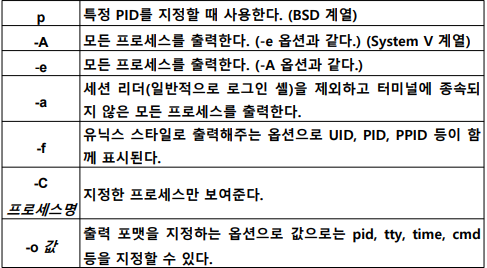
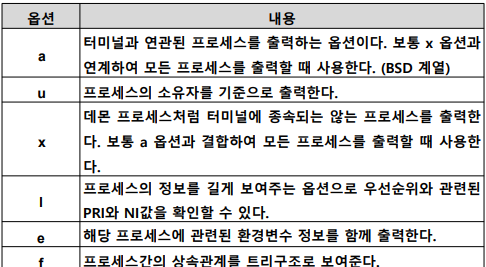
1. **멀티태스킹(Multitasking)과 작업 전환**
2. **멀티태스킹(Multitasking)이란 동시에 하나 이상의 프로그램을 실행하는 것을 말한다.**
3. **사용자 기준으로 보면 다수의 프로그램들을 백그라운드 프로세스로 실행시킨 상태에서 추가로 포어그라운드 프로세스로 다른 작업을 수행할 수 있다.**
4. **사용자가 실행한 프로세스는 중간에 중지시킬 수도 있고, 백그라운드 프로세스를 포어그라운드 프로세스로 전환하거나 포어그라운드 프로세스를 백그라운드 프로세스로 전환할 수 있다.**
5. **포어그라운드 프로세스 -> 백그라운드 프로세스를 전환 하기위해 서는 프로세스를 대기시켜야 한다 (CTRL +z)**
6. **서스펜드 : 메모리 내에서 작업 중인 프로세스를 일시적으로 중지시키는 것**
7. **작업의 상태는 jobs 명령으로 확인가능 중단된 상태에서 bg입력하면 백그라운드 프로세서로 변경**
8. **fg명령 : 백그라운드 프로세스 ->포어그라운드 프로세서 fg %직업번호 or fg 작업번호**
9. **jobs명령을 실행하면 +가 주로 처리되는 프로세서를 뜻한다. 우선순위 +다음 – 기호**
10. **데몬**
11. **데몬(daemon)은 주기적이고 지속적인 서비스 요청을 처리하기 위해 계속 실행되는 프로세스로 백그라운드로 실행된다.**
12. **리눅스에서는 서버 역할을 하는 프로그램들이 이에해당하고 보통 이름 뒤에 데몬을 뜻하는 d를 붙인다.**
13. **방법에는 standalone 방식과 inet 방식이있다.**
14. **standalone방식은 보통 부팅 시에 실행되어 해당 프로세스가 메모리에계속 상주하면서 클라이언트의 서비스 요청을 처리하는 방식이다. 웹, 메일 등과 같이 빈번한 요청이 들어오는 서비스 에 쓰인다.**
15. **inet방식은 프로세스가 메모리에 항상 상주하는 것이 아니라, 클라이언트의 서비스 요청이 들어왔을 때 관련 프로세스를 실행시키고 접속 종료 후에는 자동으로 프로세스를 종료시키는 방식이다.**
16. **자주 사용하지 않는 서비스들에 대한 효율적인 메모리 관리에 매우 유용하다.**
17. **데몬은 주기적이고 지속적인 서비스 요청을 처리하기 위한 프로세스이기 때문에 보통 부팅 시에 실행된다.**
18. **ntsysv**
19. **텍스트 환경에서 커서를 이용하여 부팅 중 자동으로 실행되는 서비스를설정할 수 있는 유틸리티이다.**
20. **명령행에서 옵션 없이 ntsysv를 실행하면 현재 구동된 실행레벨에 대한설정을 할 수 있는데, 관련 서비스는 커서로 이동하고 적용유무는 스페이스키를 사용한다.**
21. **참고로 ntsysv 대신에 명령행에서 setup 입력 후 나타나는 메뉴 항목에서 'System services'를 선택해도 된다.**



**[사용법]**

**# ntsysv [option]**

1. **ps : 동작 중인 프로세스의 상태를 출력해주는 명령이다.**
2. **ps의 옵션은 전통적인 유닉스인 System V, BSD, GNU에 따라 결과가 다르게 나타나고 표기법에도 차이를 보인다. 옵션 사용 시 System V 계열은 대시(dash, -)를 사용하고, BSD 계열은 대시(-)를 사용하지 않는다.**

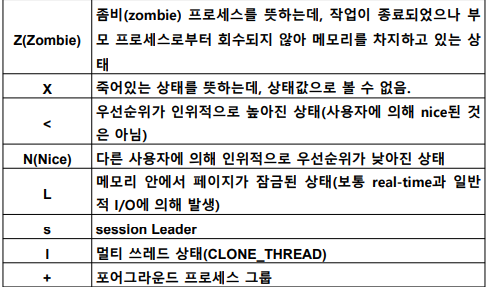
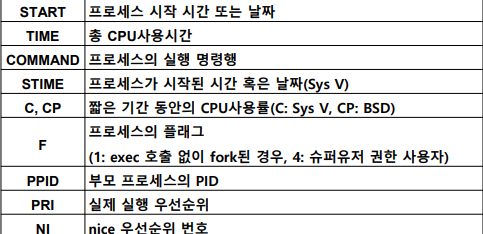
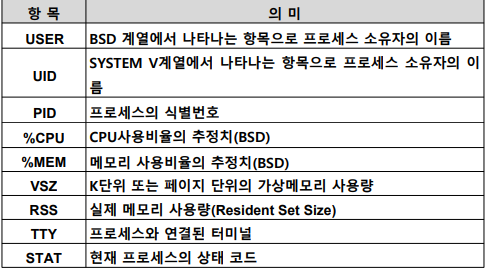


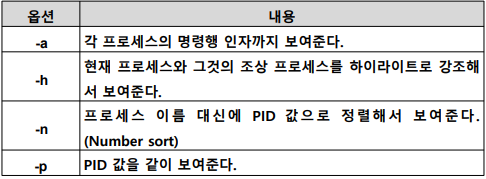
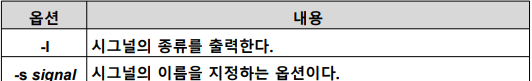
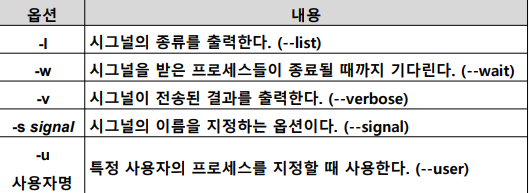
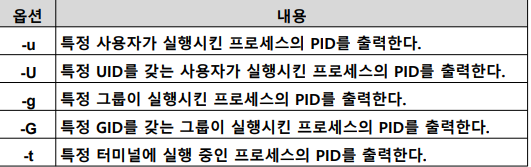
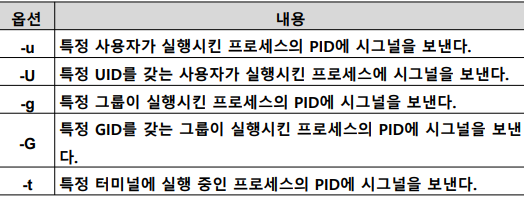
**[사용 예]**

**$ ps aux |grep sendmail → 동작중인 프로세스 중에 sendmail이라는 이름의 프로세스를 찾아서**

**출력한다**

**[ps 실행 결과로 나타나는 주요항목]**



1. **pstree : 프로세스의 상태를 트리 구조로 출력해주는 명령이다.  
   [사용 법] $pstree [option]  
   [주요 옵션]**
2. **kill : 프로세스에 특정한 시그널을 보내는 명령으로 옵션없이 실행하면 프로세스 종료 신호를 보낸다.  
   [사용 법] #kill [option] psignal] [PID또는%job\_number]  
   [주요옵션]** **# kill -l → 시그널의 종류를 출력한다  
   # kill -9 756 757 758→ pid가 756, 757, 758인 프로세스를 강제 종료한다. 'kill -KILL PID','kill –SIGKILL PID' 라고 명령을 내려도 된다.  
   # kill -HUP 10118→ pid가 10118인 프로세스를 재시작한다. 'kill -1 10118'과 같다.**
3. **killall : 같은 데몬의 여러 프로세스를 한 번에 종료시킬 때 사용하는 명령으로 프로세스명을 사용한다.  
   [사용 법] #killall [option] 프로세스명  
   [주요 옵션]** **# killall httpd→ Apache 웹서버 데몬인 httpd를 모두 종료한다.  
   # killall -HUP httpd→ httpd 데몬을 다시 실행시킨다. 이 경우 주 데몬이 아닌 웹서비스 요청시 연결을 담당하는 httpd의 프로세스들이 재시작된다.  
   # killall -v -9 httpd→ httpd 데몬에 9번 시그널(SIGKILL, KILL)을 전송하고, 전송결과를 출력한다.  
   # killall -9 –u posein→ posein 사용자의 모든 프로세스를 강제로 종료한다.**
4. **jobs : 백그라운드로 실행중인 프로세스나 현재 중지된 프로세스의 목록을 출력해주는 명령이다.  
   [사용 법] $jobs [option] -i : 프로세스 번호를 추가로 출력해준다.**
5. **fg : 백그라운드 프로세스를 포어그라운드 프로세스로 전환하는 명령이다.  
   [사용 법] $jobs [option] $ fg [%Job\_number] $ fg [Job\_number]  
   $ fg %2→ 작업번호가 2번인 프로세스를 포어그라운드로 전환한다. ‘fg 2’라고 실행해도 된다.**
6. **bg : 포그라운드 프로세스를 백그라운드 프로세스로 전환하는 명령이다.  
   포그라운드로 실행중인 프로세스를 [CTRL]+[z]를 눌러 작업을 일시 중지시킨 후에 bg명령을 사용해서 백그라운드로 전환할 수 있다.**
7. **pgrep 프로세스를 이름 기반으로 검색하는 명령으로 PID 값을 출력한다.  
   [사용법] $ pgrep [option] [pattern]  
   [주요 옵션]** **$ pgrep –u root sshd → root 소유의 sshd 프로세스의 PID를 출력한다.  
   $ pgrep –t tty2 → /dev/tty2 터미널에서 실행 중인 프로세스의 PID를 출력한다.**
8. **pkill : 프로세스명을 사용해서 특정 프로세스에 시그널을 보내는 명령이다  
   [사용법] #pkill [option] [pattern]** **[사용 예]  
   # pkill httpd → httpd 프로세스를 모두 종료시킨다. 시그널은 기본 종료시그널인 TERM(15)을 전송한다.  
   # pkill -9 -u yuloje → yuloje 사용자의 프로세스에 강제 종료 시그널인 SIGKILL(9)을 전송한다.**