

<9-1>

1. /etc/passwd : 사용자 한명에 대한 정보가 기록
2. /etc/shadow : 사용자의 패스워드에 관한 정보를 관리 -> 읽기만 가능
3. /etc/login.defs : 사용자 계정의 설정과 관련된 기본 값을 정의
4. /etc/group : 그룹에 대한 정보가 저장
5. /etc/gshadow : 그룹 암호를 저장
6. adduser : 사용자 계정 생성 -> 직접 적어서 지정
7. useradd : 옵션 추가시 사용 -> 옵션 이용해 지정
 - A. `sudo adduser -m -d /home/student -s /bin/bash student`
 - i. 디렉토리(-d) 경로에 사용자 만들고 홈디렉토리 생성(-m). 사용자 셸 지정 (-s)
8. Usermod : 사용자 계정을 수정할 수 있는 것
 - A. `uesrmod -s /bin/bash student` : student 계정의 셸을 /bin/bash로 변경
9. Userdel : 사용자 계정 삭제
 - A. `Sudo userdel -r student` : student 계정을 홈디렉토리와 메일정보를 포함 삭제
 - B. `-f` : 사용자 로그인에도 강제 삭제

<9-2>

1. groupadd : 그룹생성 -> gid같으면 다른 이름이더라도 같은 그룹에 접근 권한 가짐
 - A. `sudo groupadd -g 4000 student` : 4000번의 GID를 가지는 student 그룹 생성
 - B. `addgroup` : 동일한 gid를 가지는 그룹을 만들 수 없음
 - i. `sudo addgroup --gid 4000 student`
2. groupmod : 그룹 정보를 수정
 - A. `sudo groupmod -g 3000 student` : gid를 3000으로 변경
 - B. `sudo groupmod -n guest student` : 그룹명 변경
3. gpasswd : 그룹 암호 설정
 - A. `sudo gpasswd kdh`
4. newgrp : 소속 그룹 변경
 - A. `newgrp admin` -> 이렇게 하면 해당 디렉토리는 admin 그룹에서 실행한 명령어가 됨
5. who : 어떤 계정의 사용자가 접속해 있는지 확인

A. who -H : 타이틀을 출력해서 보기 쉽게 표시

B. who -q : 총 몇 명인지만 출력

C. -b : 재시작한 날짜 표시

D. -r : 실행 레벨을 나타냄

6. W 사용자명 : 현재 접속한 사람에 대해 자세하게 출력

7. Last 사용자명 : 현재 시스템에 접속한 사용자 이름과 로그인,로그아웃 정보 출력

8. Groups 사용자명 : 현재 사용자가 어떤 그룹에 속해있는지를 확인

9. Su : root 권한 사용

10. Sudo : root 권한으로 명령 수행

11. Sudo passwd -l student : 계정의 암호를 잠금

A. -u : 암호 잠금 해제

B. -d : 패스워드 삭제

<11>

1. 프로세스마다 고유 번호 PID를 가진다. 프로세스는 부모-자식 관계를 가지고 있다.
2. 프로세스 종류
3. Ps : 프로세스 목록 확인
 - A. -f : 현재 실행중인 프로세스의 자세한 정보 출력
 - B. -u : 프로세스 소유자 이름 , cpu사용량, 메모리 사용량 등 상세 정보 출력
 - C. -e : 현재 시스템에서 모든 사용자의 프로세스 정보 출력
 - D. Ps -fu user1 : 특정 사용자에게 대한 프로세스 상세 정보 -> u가 뒤에 와야함
4. Pgrep : 특정 프로세스 정보 검색
 - A. Pgrep vi : vi와 패턴 일치하는 프로세스 id출력
5. 시그널 : 프로세스에게 신호를 전달하기 위해 사용
 - A. Kill -9 PID : 해당 PID를 -9(강제종료) 함
 - B. Pkill -9 sleep : sleep이라는 이름의 프로세스 종료
6. 포그라운드 vs 백그라운드
 - A. 포그라운드 : 입력한 명령이 실행되어 결과가 출력될 때까지 기다려야 하는 경우
 - B. 백그라운드 : 처리결과와 관계없이 곧바로 프롬프트가 출력되어 다른 작업을 계속 할 수 있는 경우
 - i. 백그라운드 사용방법 : sleep 1000& => &붙이면 됨
 - C. Ctrl + Z 로 변경가능
7. 작업 : jobs [%작업번호] => 셸에서 수행된 모든 작업을 출력
 - A. 셸이 관리할 수 있는 프로세스
 - B. Fg %3 => 3번 작업을 포그라운드로 변경
 - C. Nohup sleep 1000& => 원래라면 로그아웃시 중단되는데 nohup 하면 계속 실행됨
 - D. At : 특정 시간에 작업수행을 예약
 - i. At 08:00 pm => sleep 1000 => ctrl+d로 완료
 - ii. At -l : 작업예약 목록

<12>

1. 컴파일러
2. `$gcc -o long long.c` => -o뒤에는 무조건 파일명 와야함
3. 다중모듈프로그램
 - A. 여러 개의 .c파일들로 이루어진 프로그램 일반적으로 복잡하며 대단위 프로그램에 적합
 - B. 여러 파일을 컴파일해서 .o 파일로 만든 후 하나의 실행파일로 만든다.

<13>

1. Makefile : 실행파일을 만들기 위해 필요한 파일들을 나열하고 조건 만족시 수행할 명령을 기술

 Main: main.o copy.o

 Gcc -o main main.o copy.o

 Main.o : main.c copy.h

 Gcc -c main.c

 Copy.o : copy.c => copy.c를 컴파일해 copy.o를 만든다.

 Gcc -c copy.c

2. `$make -f 파일이름` => makefile이 실행됨
3. 가장 대표적인 디버거 : GNU debugger gdb
 - A. `$gcc -g -o longest longest.c` => -g를 해줘야 디버거 사용가능
 - B. `$gdb 실행파일명` : 해당 파일을 디버깅함
 - C. gdb기능
 - i. l 줄번호: 지정된 줄 프린트 => l copy : 지정된 파일을 프린트
 - ii. b copy : copy파일을 breakpoint로 설정한다.
 - iii. Info b : breakpoint 의 정보 보여줌
 - iv. r : 인수 실행
 - v. c : 다음 포인트까지 진행
 - vi. quit : 디버깅 종료
 - vii. p 변수명 : 변수값 프린트
 - viii. n : 다음줄 수행 후 다시 break

4. java 컴파일러

- A. `javac a.java` => 자바 파일 생성
- B. `java a` => 자바 파일을 실행시키기 위한 명령어 `java` 사용
- C. `jdb` : 자바 디버깅
 - i. `jdb -g 소스코드`
- D. `next` : 한 라인 실행
- E. `cont` : 계속 실행
- F. `exit` : 종료
- G. `stop at 클래스` : 라인번호 => 해당 라인에 breakpoint 설정
- H. `stop in add.addition` => `add`클래스의 `addition()` 함수에 breakpoint 설정
- I. `clear` : breakpoint 제거

5. python 실행

- A. `python3 파이선소스.py`
- B. `python3 -m pdb.run` 파이썬 파일 => 디버깅할수 있도록 파일 생성