

# L08(과제10) 리포트-프로세스

Vector 사용가능/정렬은 직접 구현만

3.강좌관리가 정상 수행되어야 함.

로그인 정상 수행되어야 함.

<시스템소프트웨어>

1. 로그인
  2. 사용자 추가
  0. 종료
- 선택 : \_\_\_\_

1

로그인 화면입니다.  
사용자 id와 password를 입력하세요.

id : kwnam@naver.com  
pass : abc  
=> 비밀번호가 틀렸습니다. 다시 입력하세요.  
id : kwnam@naver.com  
pass: abcd  
=> 로그인 합니다.

<강좌관리 레코드 잠금>

1. 강좌 파일명 지정  
입력 : lecture.db  
=> 다음 데이터들이 이 파일에 저장
2. 강좌 목록 보기-강좌명정렬(Read Lock)  
=> 강좌 파일안의 모든 강좌들을 보임(삭제 강좌 제외)  
id name profName dept maxNum curNum, desc  
-----  
....
3. 신규 강좌 생성(레코드 write 잠금)  
=> 자유 입력, 강좌명 등 space 지원 필수  
강좌 정보 입력 : ....
4. 강좌 정보 변경(레코드 write 잠금)  
=> id 번호를 입력받아 seek하여 read, 해당 레코드 수정 저장  
id 입력 : 3  
수정정보 입력
5. 강좌명으로 강좌 정보 검색(2자이상 부분 문자열 검색가능)  
=> 강좌명 입력하여 검색, 2개 이상 검색될 수 있음  
검색할 강좌명 : 시스템  
id name profName dept maxNum curNum, desc  
-----
6. 강좌 삭제(레코드 write 잠금)  
=> 파일 레코드에 반영되어야 함.  
id 입력 : 2  
id 2번 강좌를 삭제하였습니다!
7. fork를 이용한 백그라운드 데이터 통계 저장
0. 위로  
선택 : \_\_\_\_

7 수행동안 다른 명령 정상 수행

User 구조체

```
int id;
char //userid, 이메일: 이메일인지 형식체크
char //password, 20자이하
char //이름, 20자: 영어 space 가능
char //주소, 도로명 주소 space 가능
int //출생년
int // 관리자 0, 일반사용자 1
time //최근 로그인 일시
int //로그인 횟수
```

<파일저장일시.log> 다음 연산 저장  
레코드 개수 : 5  
강좌 수강 최대인원 총합 : 250  
강좌당 평균 수강 인원 : 23.5  
강좌 최대인원 : 30  
강좌 최소인원 : 20

*fork() == 0 -> 과실일때  
exec() 실행해서 새로운 일 하게*

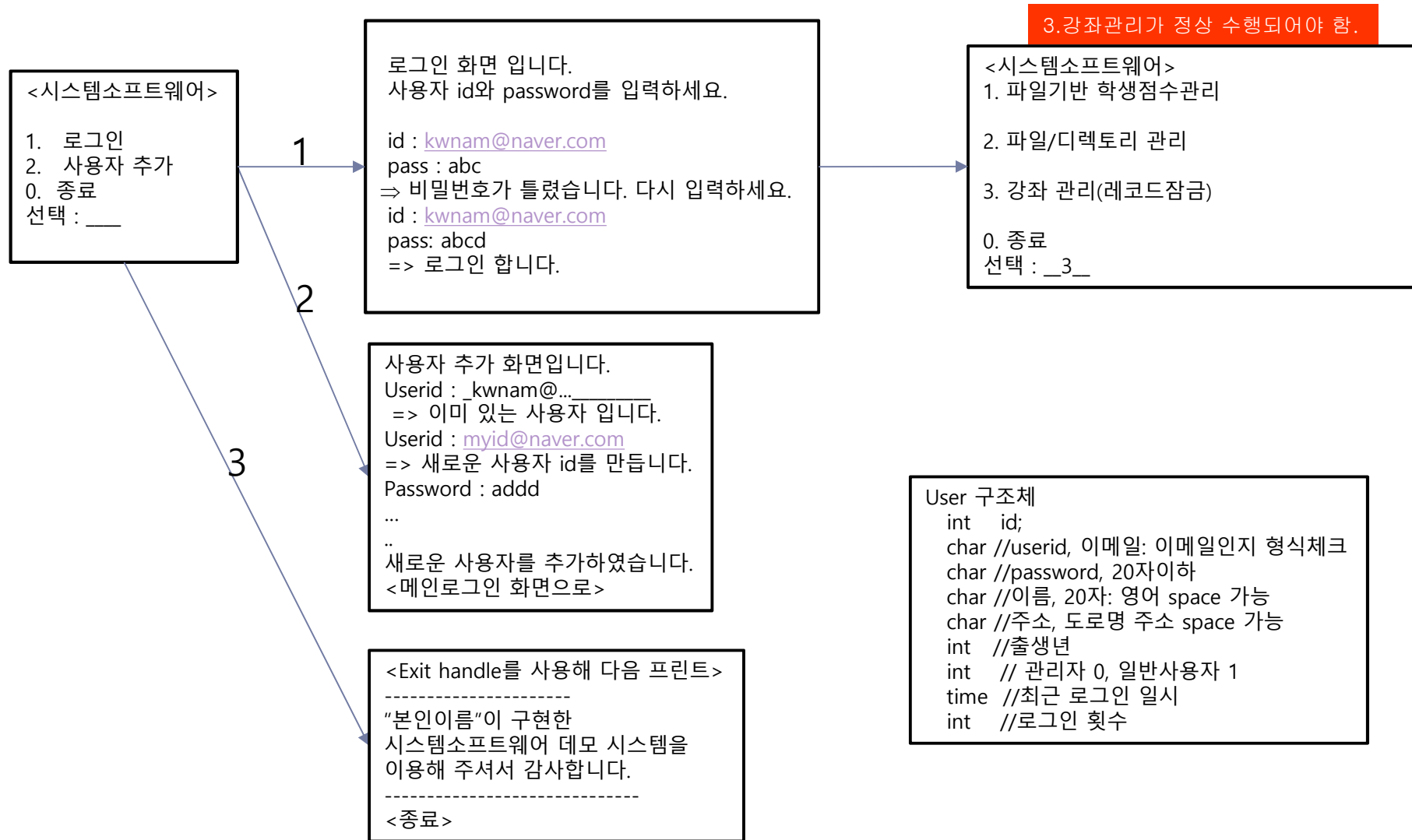
*동시에 정량할 때는*

*화면에 보이지 않고  
파일기보임.*

*백그라운드에서 처리*

*-> 파일에 계속 누적되어야 함.*

# L07(과제09) 리포트-프로세스 제어



# L06 리포트- 레코드 잠금 (시험대비)

Vector 사용가능/정렬은 직접 구현만

<시스템소프트웨어>

1. 파일기반 학생점수관리

2. 파일/디렉토리 관리

3. 강좌 관리(레코드잠금)

0. 종료

선택 : \_\_3\_\_

Lecture 구조체

int id;

char //강좌명-space 가능, 20자이상

char //교수명 -space 가능, 12자이상

char //개설학과 명, 15자

int //최대수강인원

char //비고-space 가능-40자

<강좌관리 레코드 잠금>

1. 강좌 파일명 지정

입력 : lecture.db

=> 다음 데이터들이 이 파일에 저장

2. 강좌 목록 보기-강좌명정렬(Read Lock)

=> 강좌 파일안의 모든 강좌들을 보임(삭제 강좌 제외)

id name profName dept maxNum curNum, desc

-----

....

3. 신규 강좌 생성(레코드 write 잠금)

=> 자유 입력, 강좌명 등 space 지원 필수

강좌 정보 입력 : ....

4. 강좌 정보 변경(레코드 write 잠금)

=> id 번호를 입력받아 seek하여 read, 해당 레코드 수정 저장

id 입력 : 3

수정정보 입력

5. 강좌명으로 강좌 정보 검색(2자이상 부분 문자열 검색가능)

=> 강좌명 입력하여 검색, 2개 이상 검색될 수 있음

검색할 강좌명 : 시스템

id name profName dept maxNum curNum, desc

-----

6. 강좌 삭제(레코드 write 잠금)

=> 파일 레코드에 반영되어야 함.

id 입력 : 2

id 2번 강좌를 삭제하였습니다!

0. 위로

선택 : \_\_\_\_

# L05 리포트

<시스템소프트웨어>

1. 파일기반 학생점수관리

2. 파일/디렉토리 관리

0. 종료

선택 : \_\_2\_\_

<파일/디렉토리 관리>

1. 디렉토리안의 목록 보기

보고싶은 디렉토리 입력: /home/kwnam

=> list2.c 이용해 파일/디렉토리 목록을 보여줌

2. 디렉토리 생성

생성할 디렉토리 입력: /home/kwnam/test

=> mkdir 함수 이용해 디렉토리 생성

3. 텍스트 파일의 내용 보기

보고 싶은 파일 filepath 입력 : /home/kwnam/a.txt

=> 파일 cat.c 이용해 파일 내용 출력

4. 특정 파일의 권한 수정하기

수정할 파일의 path : /home/kwnam/a.txt

권한 입력 : 766

=> a.txt의 권한을 700에서 766으로 수정하였습니다.

0. 위로

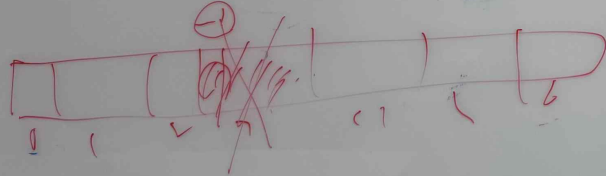
선택 : \_\_\_\_

# L04 리포트

Studente

```

int id; // 0 ~ 학생 수
char name[20]; // 이름
char sub[25]; // 과목명
short score; // 점수
float mod; // 보정점수
        
```



• 메뉴

- 1) 점수 따라 학생 찾기  
 이름:                         id:
- 2) 신학기 학생 정보 추가  
 이름:
- 3) 학생 정보 수정  
 이름:                         학번 (seek)
- 4) 학생 정보 삭제  
 이름:                         학번 (seek)  
 id:                         id: 1
- 5) 검색 (이름)  
 이름 입력 (                      )
- 0) 종료