# LAB 4

**System Programming** 

잠금

#### 레코드 단위 잠금 – read lock

```
//rdlock.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include "student.h"
/* 잠금을 이용한 학생 데이터베이스 질의 프로그램 */
int main(int argc, char *argv[]){
 int fd, id;
 struct student record;
 struct flock lock;
 if (argc < 2) {
   fprintf(stderr, "사용법: %s file\n", argv[o]);
   exit(1);
 if ((fd = open(arqv[1], O_RDONLY)) == -1) {
    perror(argv[1]);
   exit(2);
```

```
printf("\n검색할 학생의 학번 입력:");
while (scanf("%d", &id) == 1)
 lock.l_type = F_RDLCK;
 lock.l whence = SEEK SET;
 lock.l_start = (id-START_ID)*sizeof(record);
 lock.l len = sizeof(record);
 if (fcntl(fd,F_SETLKW, &lock) == -1) { /* 읽기 잠금 */
   perror(ar O qv[1]);
   exit(3);
 lseek(fd, (id-START ID)*sizeof(record), SEEK SET);
 if ((read(fd, (char *) &record, sizeof(record)) > o) &&
   (record.id != o))
   printf("이름:%s\t 학번:%d\t 점수:%d\n",
      record.name, record.id, record.score);
 else printf("레코드 %d 없음\n", id);
 lock.l_type = F_UNLCK;
 fcntl(fd,F_SETLK, &lock); /* 잠금 해제 */
 printf("\n검색할 학생의 학번 입력:");
close(fd);
exit(o);
```

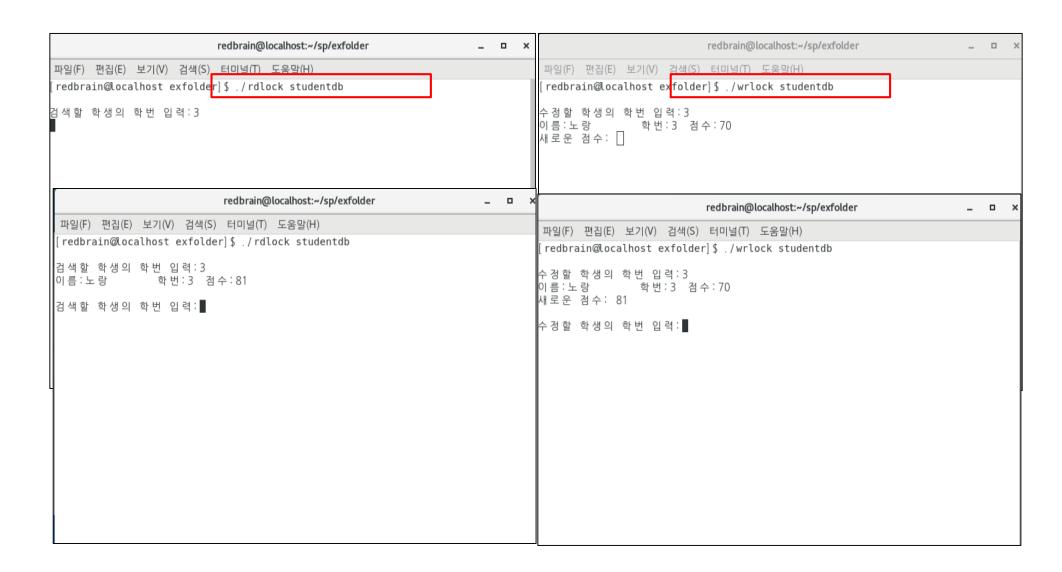
```
//wrlock.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include "student.h"
/* 잠금을 이용한 학생 데이터베이스 수정 프로그램 */
int main(int argc, char *argv[]){
 int fd, id;
  struct student record;
  struct flock lock;
  if (argc < 2) {
   fprintf(stderr, "사용법: %s file\n", argv[o]);
   exit(1);
  if ((fd = open(argv[1], O_RDWR)) == -1) {
   perror(argv[1]);
   exit(2);
```

#### 레코드 단위 잠금 – write lock

```
printf("\n수정할 학생의 학번 입력:");
while (scanf("%d", &id) == 1) {
    lock.l_type = F_WRLCK;
    lock.l_whence = SEEK_SET;
    lock.l_start = (id-START_ID)*sizeof(record);
    lock.l_len = sizeof(record);
    if (fcntl(fd,F_SETLKW, &lock) == -1) { /* 쓰기 잠금 */
        perror(argv[1]);
        exit(3);
    }
```

```
lseek(fd, (long) (id-START_ID)*sizeof(record), SEEK_SET);
  if ((read(fd, (char *) &record, sizeof(record)) > o) && (record.id != o))
   printf("이름:%s\t 학번:%d\t 점수:%d\n",
       record.name, record.id, record.score);
  else printf("레코드 %d 없음\n", id);
  printf("새로운 점수: ");
  scanf("%d", &record.score);
  lseek(fd, (long) -sizeof(record), SEEK_CUR);
  write(fd, (char *) &record, sizeof(record));
 lock.l_type = F_UNLCK;
 fcntl(fd, F_SETLK, &lock); /* 잠금 해제 */
  printf("\n수정할 학생의 학번 입력:");
close(fd);
exit(o);
```

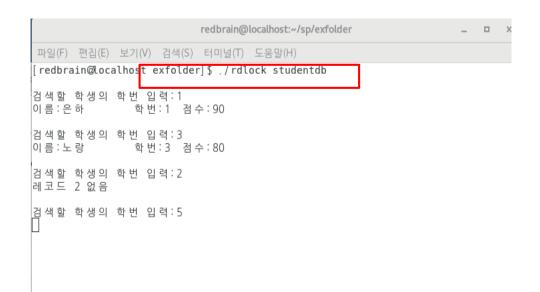
## 레코드 단위 잠금 실행 예

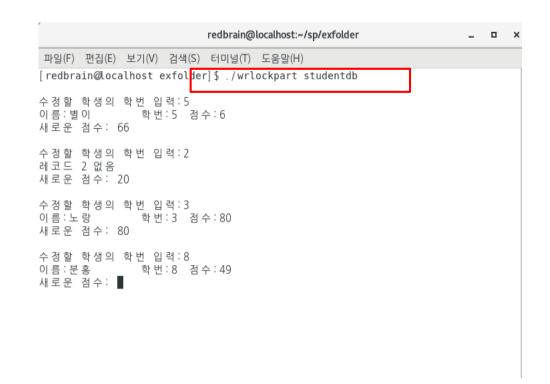


## 파일 일부 잠금 – write lock

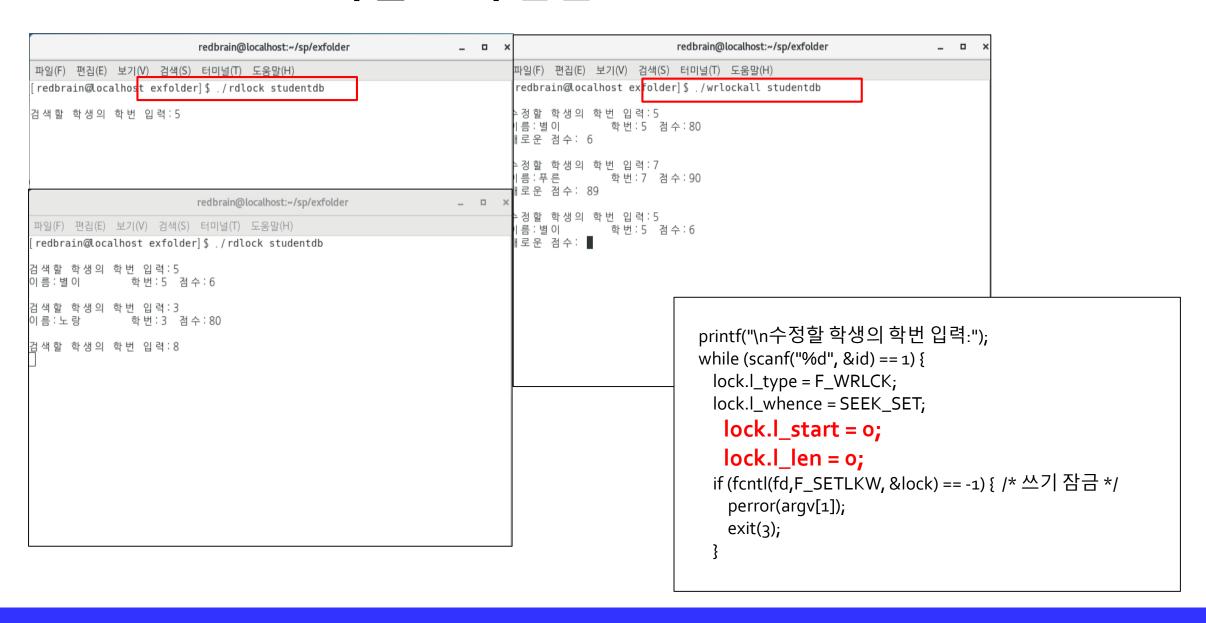
```
printf("\n수정할 학생의 학번 입력:");
while (scanf("%d", &id) == 1) {
    lock.l_type = F_WRLCK;
    lock.l_whence = SEEK_SET;
    lock.l_start = (4)*sizeof(record);
    lock.l_len = o;
    if (fcntl(fd,F_SETLKW, &lock) == -1) { /* 쓰기 잠금 */
        perror(argv[1]);
        exit(3);
    }
....
//lock.l_type = F_UNLCK;
```

#### 파일 일부 잠금 – write lock



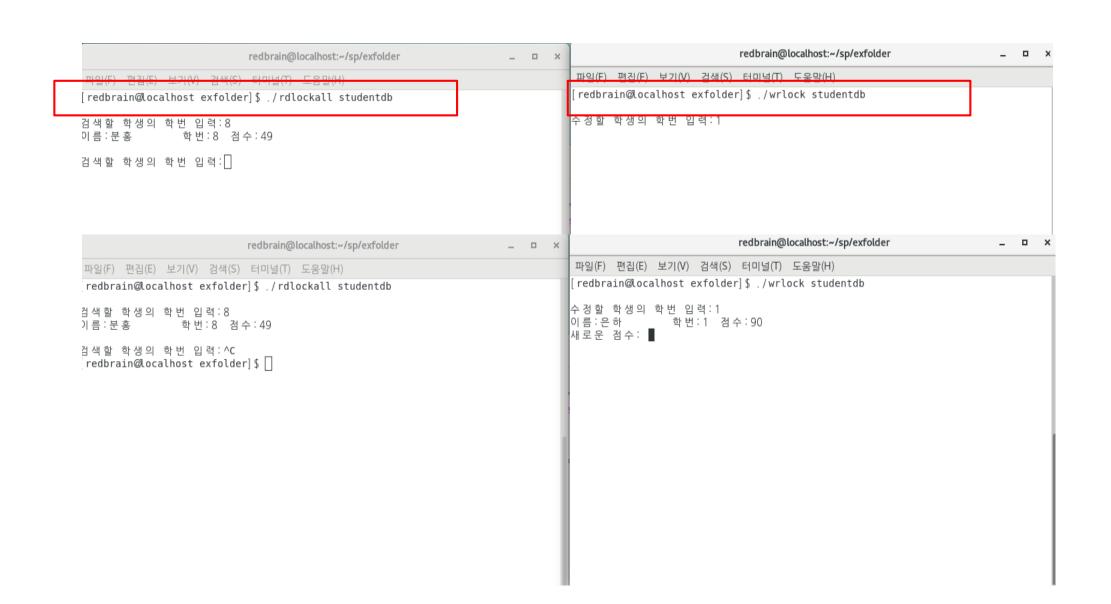


#### 파일 전체 잠금 -write lock



#### 파일 전체 - read lock

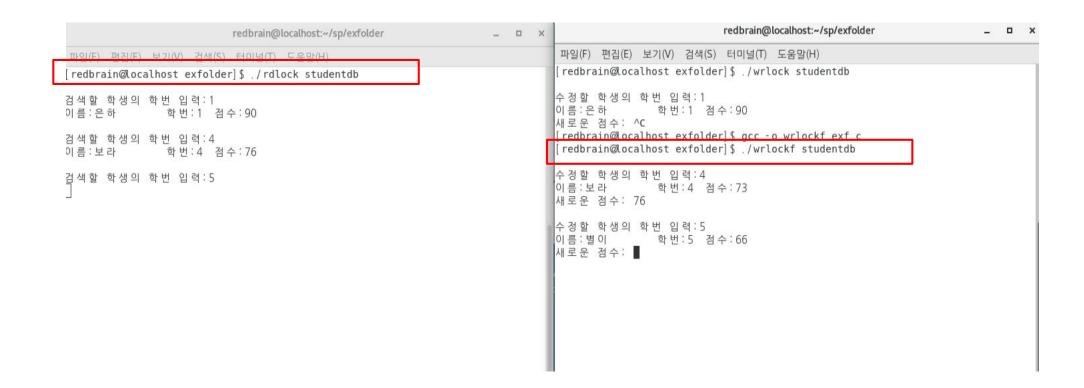
```
printf("\n검색할 학생의 학번 입력:");
 while (scanf("\%d", \&id) == 1) {
   lock.l_type = F_RDLCK;
   lock.l_whence = SEEK_SET;
   lock.l_start = o;
   lock.l_len = o;
   if (fcntl(fd,F_SETLKW, &lock) == -1) { /* 읽기 잠금 */
     perror(argv[1]);
     exit(3);
   lseek(fd, (id-START_ID)*sizeof(record), SEEK_SET);
   if ((read(fd, (char *) &record, sizeof(record)) > o) &&
     (record.id != o))
     printf("이름:%s\t 학번:%d\t 점수:%d\n",
        record.name, record.id, record.score);
   else printf("레코드 %d 없음\n", id);
   //lock.l_type = F_UNLCK;
```



#### 레코드 잠금 - write lock

```
//wrlockf.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include "student.h"
/* 잠금을 이용한 학생 데이터베이스 수정 프로그램 */
int main(int argc, char *argv[])
 int fd, id;
 struct student record;
 if (argc < 2) {
   fprintf(stderr, "사용법: %s file\n", argv[o]);
   exit(1);
  if ((fd = open(arqv[1], O_RDWR)) == -1) {
   perror(argv[1]);
   exit(2);
```

```
printf("\n수정할 학생의 학번 입력:");
 while (scanf("\%d", \&id) == 1) {
  lseek(fd, (long) (id-START_ID)*sizeof(record), SEEK_SET);
   if (lockf(fd, F_LOCK, sizeof(record)) == -1) { /* 쓰기 잠금 */
    perror(arqv[1]);
    exit(3);
   if ((read(fd, (char *) &record, sizeof(record)) > o) &&
    (record.id != o))
    printf("이름:%s\t 학번:%d\t 점수:%d\n",
       record.name, record.id, record.score);
  else printf("레코드 %d 없음\n", id);
   printf("새로운 점수: ");
  scanf("%d", &record.score);
   lseek(fd, (long) -sizeof(record), SEEK_CUR);
  write(fd, (char *) &record, sizeof(record));
  lseek(fd, (long) (id-START_ID)*sizeof(record), SEEK_SET);
   lockf(fd, F_ULOCK, sizeof(record)); /* 잠금 해제 */
  printf("\n수정할 학생의 학번 입력:");
close(fd);
 exit(o);
```



#### 파일 전체 잠금 - write lock

```
//wrlockf.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include "student.h"
/* 잠금을 이용한 학생 데이터베이스 수정 프로그램 */
int main(int argc, char *argv[])
 int fd, id;
 struct student record;
 if (argc < 2) {
   fprintf(stderr, "사용법: %s file\n", argv[o]);
   exit(1);
 if ((fd = open(argv[1], O_RDWR)) == -1) {
   perror(arqv[1]);
   exit(2);
```

```
printf("\n수정할 학생의 학번 입력:");
 while (scanf("%d", &id) == 1)
  lseek(fd, (long) (id-START_ID)*sizeof(record), SEEK_SET);
  if (lockf(fd, F_LOCK,o) == -1) { /* 쓰기 잠금 */
    perror(arqv[1]);
    exit(3);
   if ((read(fd, (char *) &record, sizeof(record)) > o) &&
    (record.id!= 0)
    printf("이름:%s\t 학번:%d\t 점수:%d\n",
        record.name, record.id, record.score);
   else printf("레코드 %d 없음\n", id);
   printf("새로운 점수: ");
  scanf("%d", &record.score);
  lseek(fd, (long) -sizeof(record), SEEK_CUR);
  write(fd, (char *) &record, sizeof(record));
  lseek(fd, (long) (id-START_ID)*sizeof(record), SEEK_SET);
   lockf(fd, F_ULOCK, sizeof(record)); /* 잠금 해제 */
  printf("\n수정할 학생의 학번 입력:");
 close(fd);
 exit(o);
```

# 파일 전체 잠금 - write lock, 수정된 레코드는 lock 해제

