

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[0]);
        exit(1);
    }
    if ((fd = open(argv[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(argv[1]);
        exit(3);
    }
    lseek(fd, (id-START_ID)*sizeof(record), SEEK_SET);
    if ((read(fd, (char *) &record, sizeof(record)) > 0) &&
        (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);

    lock.l_type = F_UNLCK;
    fcntl(fd, F_SETLK, &lock); /* 잠금 해제 */
    printf("\n검색할 학생의 학번 입력:");
}
close(fd);
exit(0);
}
```

//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[0]);
        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)) > 0) && (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));

lock.l_type = F_UNLCK;
fcntl(fd, F_SETLK, &lock); /* 잠금 해제 */
printf("\n수정할 학생의 학번 입력:");
}

close(fd);
exit(0);
}
```

레코드 단위 잠금 실행 예

<pre>redbrain@localhost:~/sp/exfolder 파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H) [redbrain@localhost exfolder]\$./rdlock studentdb 검색할 학생의 학번 입력:3 </pre>	<pre>redbrain@localhost:~/sp/exfolder 파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H) [redbrain@localhost exfolder]\$./wrlock studentdb 수정할 학생의 학번 입력:3 이름:노랑 학번:3 점수:70 새로운 점수: </pre>
<pre>redbrain@localhost:~/sp/exfolder 파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H) [redbrain@localhost exfolder]\$./rdlock studentdb 검색할 학생의 학번 입력:3 이름:노랑 학번:3 점수:81 검색할 학생의 학번 입력: </pre>	<pre>redbrain@localhost:~/sp/exfolder 파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H) [redbrain@localhost exfolder]\$./wrlock studentdb 수정할 학생의 학번 입력:3 이름:노랑 학번:3 점수:70 새로운 점수: 81 수정할 학생의 학번 입력: </pre>

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 = 0;
    if (fcntl(fd, F_SETLKW, &lock) == -1) { /* 쓰기 잠금 */
        perror(argv[1]);
        exit(3);
    }

    ....

    //lock.l_type = F_UNLCK;
```

파일 일부 잠금 – write lock

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./rdlock studentdb
검색할 학생의 학번 입력:1
이름:은하 학번:1 점수:90
검색할 학생의 학번 입력:3
이름:노랑 학번:3 점수:80
검색할 학생의 학번 입력:2
레코드 2 없음
검색할 학생의 학번 입력:5
█
```

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./wrlockpart studentdb
수정할 학생의 학번 입력:5
이름:별이 학번:5 점수:6
새로운 점수: 66
수정할 학생의 학번 입력:2
레코드 2 없음
새로운 점수: 20
수정할 학생의 학번 입력:3
이름:노랑 학번:3 점수:80
새로운 점수: 80
수정할 학생의 학번 입력:8
이름:분홍 학번:8 점수:49
새로운 점수: █
```

파일 전체 잠금 -write lock

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./rdlock studentdb

검색할 학생의 학번 입력:5
이름:별이 학번:5 점수:6

검색할 학생의 학번 입력:3
이름:노랑 학번:3 점수:80

검색할 학생의 학번 입력:8

```

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./wrlockall studentdb

수정할 학생의 학번 입력:5
이름:별이 학번:5 점수:80
새로운 점수: 6

수정할 학생의 학번 입력:7
이름:푸른 학번:7 점수:90
새로운 점수: 89

수정할 학생의 학번 입력:5
이름:별이 학번:5 점수:6
새로운 점수: 
```

```
printf("\n수정할 학생의 학번 입력:");
while (scanf("%d", &id) == 1) {
    lock.l_type = F_WRLCK;
    lock.l_whence = SEEK_SET;
    lock.l_start = 0;
    lock.l_len = 0;
    if (fcntl(fd, F_SETLKW, &lock) == -1) { /* 쓰기 잠금 */
        perror(argv[1]);
        exit(3);
    }
}
```


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 = 0;
    lock.l_len = 0;
    if (fcntl(fd, F_SETLK, &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)) > 0) &&
        (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);

    //lock.l_type = F_UNLCK;
```

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./rdlockall studentdb

검색할 학생의 학번 입력:8
이름:분홍      학번:8 점수:49

검색할 학생의 학번 입력: 
```

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./rdlockall studentdb

검색할 학생의 학번 입력:8
이름:분홍      학번:8 점수:49

검색할 학생의 학번 입력: ^C
[redbrain@localhost exfolder]$ 
```

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./wrlock studentdb

수정할 학생의 학번 입력:1
```

```
redbrain@localhost:~/sp/exfolder
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[redbrain@localhost exfolder]$ ./wrlock studentdb

수정할 학생의 학번 입력:1
이름:은하      학번:1 점수:90
새로운 점수: 
```

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[0]);
        exit(1);
    }

    if ((fd = open(argv[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(argv[1]);
        exit(3);
    }

    if ((read(fd, (char *) &record, sizeof(record)) > 0) &&
        (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(0);
}
```

redbrain@localhost:~/sp/exfolder

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

[redbrain@localhost exfolder]\$./rdlock studentdb

검색할 학생의 학번 입력: 1
이름:은하 학번:1 점수:90

검색할 학생의 학번 입력: 4
이름:보라 학번:4 점수:76

검색할 학생의 학번 입력: 5
]

redbrain@localhost:~/sp/exfolder

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

[redbrain@localhost exfolder]\$./wrlock studentdb

수정할 학생의 학번 입력: 1
이름:은하 학번:1 점수:90
새로운 점수: ^C
[redbrain@localhost exfolder]\$ gcc -o wrlockf exf.c
[redbrain@localhost exfolder]\$./wrlockf studentdb

수정할 학생의 학번 입력: 4
이름:보라 학번:4 점수:73
새로운 점수: 76

수정할 학생의 학번 입력: 5
이름:별이 학번:5 점수:66
새로운 점수: █

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[0]);
        exit(1);
    }

    if ((fd = open(argv[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, 0) == -1) { /* 쓰기 잠금 */
            perror(argv[1]);
            exit(3);
        }

        if ((read(fd, (char *) &record, sizeof(record)) > 0) &&
            (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(0);
}
```

파일 전체 잠금 - write lock, 수정된 레코드는 lock 해제

The image displays four terminal windows arranged in a 2x2 grid, all running on a system with the user 'redbrain' at 'localhost' in the directory '~/sp/exfolder'. Each window has a menu bar with options: 파일(F), 편집(E), 보기(V), 검색(S), 터미널(T), 도움말(H).

- Top-Left Window:** Shows the command `./rdlock studentdb` being executed. Below the command, it prompts for a student ID to search: `검색할 학생의 학번 입력: 6`, followed by a blank line for input.
- Top-Right Window:** Shows the command `./wrlockfall studentdb` being executed, which is highlighted with a red rectangle. Below the command, it prompts for a student ID to modify: `수정할 학생의 학번 입력: 1`. It then displays the current record: `이름:은하 학번:1 점수:90`, and prompts for a new score: `새로운 점수:` with a cursor.
- Bottom-Left Window:** Shows the command `./rdlock studentdb` being executed. It prompts for a student ID to search: `검색할 학생의 학번 입력: 6`. It then displays the record: `이름:그린 학번:6 점수:52`. It prompts for another student ID: `검색할 학생의 학번 입력: 7`, displays the record: `이름:푸른 학번:7 점수:89`, and prompts for a third student ID: `검색할 학생의 학번 입력:` with a cursor.
- Bottom-Right Window:** Shows the command `./wrlockfall studentdb` being executed. It prompts for a student ID to modify: `수정할 학생의 학번 입력: 1`. It displays the record: `이름:은하 학번:1 점수:90` and prompts for a new score: `새로운 점수: 91`. It then prompts for another student ID: `수정할 학생의 학번 입력: 7`, displays the record: `이름:푸른 학번:7 점수:89` and prompts for a new score: `새로운 점수: 89`. It then prompts for a third student ID: `수정할 학생의 학번 입력: 6`, displays the record: `이름:그린 학번:6 점수:52` and prompts for a new score: `새로운 점수: 52`. It then prompts for a fourth student ID: `수정할 학생의 학번 입력: 3`, displays the record: `이름:노랑 학번:3 점수:80` and prompts for a new score: `새로운 점수: 80`. It then prompts for a fifth student ID: `수정할 학생의 학번 입력: 7`, displays the record: `이름:푸른 학번:7 점수:89` and prompts for a new score: `새로운 점수: 89`. Finally, it prompts for a sixth student ID: `수정할 학생의 학번 입력:` with a cursor.