상명대학교 소프트웨어학부 "EA0014: 유닉스 프로그래밍" 실습 지침서

- 실습 번호: lab-02
- 실습 디렉토리: ~/unix/lab-02 이름: _____
- 실습 날짜: 년 월 일 분반:
- 실습 제목: File I/O
- 실습 내용: 학번: _____

1. (파일 "2-1.c") 다음과 같이 "2-1.txt"를 생성하는 "2-1.c"를 vi를 이용해 만드시오.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>

int main(void) {
    int fd;
    mode_t mode;

    mode = S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH;

    fd = open("2-1.txt", O_CREAT, mode);
    if (fd == -1) {
        perror("Creat");
        exit(1);
    }
    close(fd);

    return 0;
}
```

gcc를 이용 다음과 같이 컴파일하시오.

$gcc –o 2-1 2-1.c

ls로 생성된 파일을 확인하시오.

$./2-1 로 실행하고

ls로 생성된 파일을 확인하시오.

2. (파일 "2-2.c") 아래와 같이 "2-1.txt"에 UNIX SYSTEM PROGRAMMING 이란 문자열을 write하는 프로그램 2-2c를 작성하고

$ gcc –o 2-2 2-2.c

를 이용하여 컴파일 하시오.

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(void) {
    int fd;
    int n;

    char str[] = "UNIX SYSTEM PROGRAMMING";

    fd = open("2-1.txt", O_WRONLY);
    if (fd == -1) {
        perror("Open");
        exit(1);
    }
    n = write(fd, str, strlen(str));
    if (n != strlen(str)) {
        perror("Write");
        exit(2);
    }
    close(fd);

    return 0;
}
```

프로그램 2-2를 수행하기 전에 ls -l과 cat 2-1.txt를 통해 디렉토리와 파일의 상태를 확인 하시오.

$ ./2-2 를 통해 프로그램을 수행하고

전에 ls -l과 cat 2-1.txt를 통해 디렉토리와 파일의 상태를 확인 하시오.

3. (파일 "2-3.c") UNIX SYSTEM PROGRAMMING 이 들어 있는 "2-1.txt"를 "2-3.txt"로 복사하는 프로그램 2-3.c을 read()와 write()를 이용하여 작성하고 gcc –o 2-3 2-3.c를 이용하여 컴파일한 후 ./2-3으로 수행하시오.
ls -l과 cat을 가지고 결과를 확인하시오.

4. (파일 "2-4.c") 아래 프로그램과 같이 "2-4.c"를 만들고 컴파일 후 수행하여 결과를 확인하고 설명하시오.

```
#include <sys/types.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>

int main(void) {
    int fd, n;
    off_t start, cur;
    char buf[256];

    fd = open("2-1.txt", O_RDONLY);
    if (fd == -1) {
        perror("Open unix.txt");
        exit(1);
    }

    start = lseek(fd, 0, SEEK_CUR);
    n = read(fd, buf, 255);
    buf[n] = '\0';
    printf("Offset start=%d, Read Str=%s, n=%d\n", (int)start, buf, n);
    cur = lseek(fd, 0, SEEK_CUR);
    printf("Offset cur=%d\n", (int)cur);

    start = lseek(fd, 7, SEEK_SET);
    n = read(fd, buf, 255);
    buf[n] = '\0';
    printf("Offset start=%d, Read Str=%s", (int)start, buf);

    close(fd);

    return 0;
}
```

5. (파일 "2-5.c") 파일 입출력 시스템 콜이 아닌 파일 입출력 함수를 이용하여 "2-1.txt"를 "2-5.txt"로 복사하는 프로그램 2-5.c를 작성하고 실습 3번과 같은 과정을 거쳐 확인하시오.

6. (파일 "2-6.c") 아래 임시파일의 이름을 만드는 프로그램을 "2-6.c"로 작성하고 gcc –o 2-6 2-6.c를 이용하여 컴파일한 후 2-6을 수행하여 어떤 결과가 나오는지를 적으시오.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(void) {
    char *fname;
    char fntmp[BUFSIZ];
    char template[32];

    fname = tmpnam(NULL);
    printf("1. TMP File Name(tmpnam) : %s\n", fname);

    tmpnam(fntmp);
    printf("2. TMP File Name(tmpnam) : %s\n", fntmp);

    fname = tempnam("/tmp", "coredump");
    printf("3. TMP File Name(tempnam) : %s\n", fname);

    strcpy(template, "/tmp/coredumpXXXXXX");
    fname = mktemp(template);
    printf("4. TMP File Name(mktemp) : %s\n", fname);

    return 0;
}
```

7. 위 6번의 컴파일 과정에서 warning이 발생하지만 수행에는 지장을 주지 않는다. 그러나 gcc 컴파일러는 mkstemp()라는 함수의 사용을 권한다. mkstemp()에 대해 조사하여 적고 mkstemp()를 사용하여 임시파일 이름을 만들어 출력하는 프로그램 "2-7.c"를 작성하고 컴파일하여 2-7을 만들고 이를 수행하여 결과를 적으시오.

8. (2-8.c, 2-8) 다음과 같은 파일 2-8.dat를 만들고 그 내용을 확인하시오.

```
    국 영 수
Kim 90 80 60
Lee 75 90 90
Park 95 70 88
Choi 90 67 45
Jung 90 80 80
```

위의 파일 2-8.dat를 읽어 2-8.out 파일에 각자의 합과 평균을 구하여 합이 높은 순서로 출력하고 마지막에 과목별 평균을 출력하는 프로그램 2-8.c를 fscanf()의 사용 없이 작성하고 컴파일하여 2-8을 만들고 이를 수행, 확인하시오.

끝.