상명대학교 컴퓨터과학과 "EA0014: 유닉스 프로그래밍" 실습 지침서

- 실습 번호: lab-03
- 실습 디렉토리: ~/unix/lab-03 이름: _____
- 실습 날짜: 년 월 일
- 실습 제목: File & Directory
- 실습 내용: 학번: _____

1. (stat() syscall, 파일 "3-1.c") $cp /etc/passwd password.txt 를 통해 디렉토리에 password.txt 파일을 만든다.
아래와 같이 3-1.c를 편집하여 만들고

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

int main(void) {
    struct stat buf;

    stat("password.txt", &buf);

    printf("Inode = %d\n", (int)buf.st_ino);
    printf("Mode = %o\n", (unsigned int)buf.st_mode);
    printf("Nlink = %o\n",(unsigned int) buf.st_nlink);
    printf("UID = %d\n", (int)buf.st_uid);
    printf("GID = %d\n", (int)buf.st_gid);
    printf("SIZE = %d\n", (int)buf.st_size);
    printf("Atime = %d\n", (int)buf.st_atime);
    printf("Mtime = %d\n", (int)buf.st_mtime);
    printf("Ctime = %d\n", (int)buf.st_ctime);
    printf("Blksize = %d\n", (int)buf.st_blksize);
    printf("Blocks = %d\n", (int)buf.st_blocks);

    return 0;
}
```

gcc를 이용 다음과 같이 컴파일 하시오.
$gcc –o 3-1 3-1.c

$./3-1 로 실행하고 결과를 적으시오.

$ls –lai 하여 위의 결과와 비교 설명하시오.

2. (파일종류 매크로 사용, 파일 "3-2.c") 아래와 같이 "3-2.c"를 편집하여 만든다.

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
#include <stdio.h>

int main(void) {
    struct stat buf;

    stat("password.txt", &buf);

    printf("Mode = %o (16진수: %x)\n", (unsigned int)buf.st_mode, (unsigned
int)buf.st_mode);

    if(S_ISFIFO(buf.st_mode)) printf("password.txt: FIFO\n");
    if(S_ISDIR(buf.st_mode)) printf("password.txt: Directory\n");
    if(S_ISREG(buf.st_mode)) printf("password.txt: Regualr File\n");

    return 0;
}
```

$ gcc –o 3-2 3-2.c 로 컴파일하고 $ ./3-2 를 통해 프로그램을 수행하고 결과를 적으시오.

3. (chmod() syscall, 파일 "3-3.c") ls –l로 password.txt의 접근 권한을 확인하시오.
아래와 같이 3-3.c를 만들고

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

int main(void) {
    struct stat buf;

    stat("password.txt", &buf);
    printf("mode before change = %o\n", (unsigned int)buf.st_mode);

    buf.st_mode |= S_IWGRP;
    buf.st_mode &= ~(S_IROTH);
    chmod("password.txt", buf.st_mode);
    stat("password.txt", &buf);

    printf("mode after change = %o\n", (unsigned int)buf.st_mode);

    return 0;
}
```

$ gcc –o 3-3 3-3.c 를 통해 컴파일한후 $./3-3 으로 수행하고 결과를 적으시오.
다시 ls -l하여 처음과 비교하시오.

4. (link(), symlink() syscall, 파일"3-4.c") 아래 프로그램과 같이 "3-4.c"를 만들고 컴파일 후 수행하여 결과를 확인하고 왜 그런 결과가 나오는지를 설명하시오.

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

int main(void) {
    struct stat buf;

    stat("password.txt", &buf);
    printf("Link Count before link()= %d\n", (int)buf.st_nlink);

    link("password.txt", "password.ln");

    stat("password.txt", &buf);
    printf("Link Count after link()= %d\n", (int)buf.st_nlink);

    stat("password.ln", &buf);
    printf("Link Count of password.ln= %d\n", (int)buf.st_nlink);

    symlink("password.txt", "password.sln");

    stat("password.txt", &buf);
    printf("Link Count of password.txt= %d\n", (int)buf.st_nlink);

    stat("password.sln", &buf);
    printf("Link Count of password.sln= %d\n", (int)buf.st_nlink);

    lstat("password.sln", &buf);
    printf("Link Count of password.sln itself= %d\n", (int)buf.st_nlink);

    return 0;
}
```

5. (mkdir() syscall, 파일 "3-5.c") linux, programming이란 이름의 디렉토리를 만드는 프로그램 3-5.c를 작성하고 컴파일, 수행하시오.

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

int main(void) {
    if (mkdir("linux", 0755) == -1) {
        perror("linux");
        exit(1);
    }

    if (mkdir("programming", 0644) == -1) {
        perror("programming");
        exit(1);
    }

    return 0;
```

```
}
```

$ ls –l을 통해 디렉토리가 만들어 진 것을 확인하시오.

6. (rename() syscall, 파일 "3-6.c") 시스템 콜 rename()을 이용하여 디렉토리 linux를 LINUX로 바꾸는 프로그램 3-6.c 를 작성하고 컴파일, 수행하시오. $ ls –l로 디렉토리의 이름이 바뀐 것을 확인하시오.

7. (rmdir() syscall, 파일 "3-7.c") 시스템 콜 rmdir()을 이용하여 디렉토리 programming을 지우는 프로그램 3-7.c 를 작성하고 컴파일, 수행하시오. $ ls –l로 결과를 확인하시오.

8. (opendir(), readdir(), 파일 "3-8.c") 다음과 같이 현재 디렉토리의 내용을 읽어 출력하는 프로그램 3-8.c를 작성하고 컴파일하여 3-8을 만들고 수행하여 그 결과를 확인하시오.

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

int main(void) {
    DIR *dp;
    struct dirent *dent;

    if ((dp = opendir(".")) == NULL) {
        perror("opendir: .");
        exit(1);
    }

    while ((dent = readdir(dp))) {
        printf("Name : %s  ", dent->d_name);
        printf("Inode : %d\n", (int)dent->d_ino);
    }

    closedir(dp);

    return 0;
}
```

끝.