경성대학교
KYUNGSUNG UNIVERSITY

시스템프로그래밍 기말과제

학번 : 2019775054

이름 : 전상훈

날짜 : 2023-06-13

과목 : 시스템프로그래밍

목차

1. cat

```
#include <ctype.h>
#include <getopt.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    FILE *fp;
    int c;
    int count_characters = 0;
    int current_line = 1;
    int line_number = 0;
    int print_char_count = 0;
    int print_line_number = 0;
    int one_line = 0;
    int no_double_empty_lines = 0;
    int is_empty_line = 0;
    int show_tab_as_caretI = 0;

    struct option long_options[] = {
        {"count", no_argument, 0, 'c'},
        {"one-line", no_argument, 0, 'o'},
        {"line-number", no_argument, 0, 'l'},
        {"no-double-empty-lines", no_argument, 0, 'd'},
        {"show-tab-as-caretI", no_argument, 0, 'T'},
        {0, 0, 0, 0}
    };

    int option_index = 0;
    int option;
    while ((option = getopt_long(argc, argv, "coldT", long_options, &option_index)) != -1) {
        if (option == 'c') {
            print_char_count = 1;
        } else if (option == 'o') {
            one_line = 1;
        } else if (option == 'l') {
            print_line_number = 1;
        } else if (option == 'd') {
            no_double_empty_lines = 1;
        } else if (option == 'T') {
            show_tab_as_caretI = 1;
        } else {
            fprintf(stderr, "사용법 : %s [--count] [--one-line] [--line-number] [--no-double-empty-lines] [--show-tab-as-caretI] [file]₩n", argv[0]);
            exit(EXIT_FAILURE);
        }
    }

    if (optind < argc) {
        fp = fopen(argv[optind], "r");
        if (fp == NULL) {
            perror("Error file");
            exit(EXIT_FAILURE);
        }
    } else {
        fp = stdin;
    }

    c = getc(fp);
    while (c != EOF) {
        if (print_line_number && !line_number) {
            printf("%d: ", current_line);
            line_number = 1;
        }

        if (!isspace(c) || (one_line && c != '₩n')) {
            count_characters++;
        }
        if (one_line) {
            if (c == '₩n') {
                c = getc(fp);
                continue;
            }
        }

        if (no_double_empty_lines && c == '₩n') {
            is_empty_line++;
            if (is_empty_line >= 2) {
                c = getc(fp);
                continue;
            }
        } else {
            is_empty_line = 0;
        }

        if (show_tab_as_caretI && c == '₩t') {
            printf("^I");
        } else {
            putc(c, stdout);
        }

        if (c == '₩n') {
            line_number = 0;
            current_line++;
        }

        c = getc(fp);
    }
    fclose(fp);

    if (print_char_count) {
        printf("₩n글자수는 : %d 개 입니다.₩n", count_characters);
    }

    return 0;
}
```

1. cat

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcat test -a
./spcat: invalid option -- 'a'
사용법 : ./spcat [--count / -c ] [--one-line / -o ] [--line-number / -l] [--no-double-empty-lines / -d ] [--show-tab-as-caretI / -T ]
```

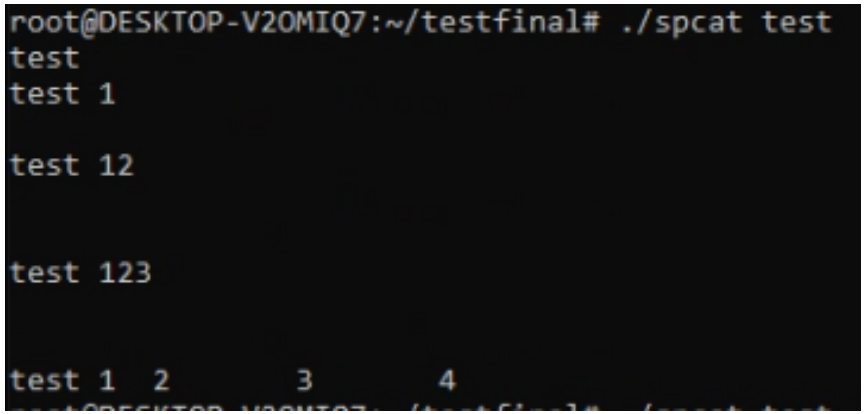
[spcat]

옵션 종류

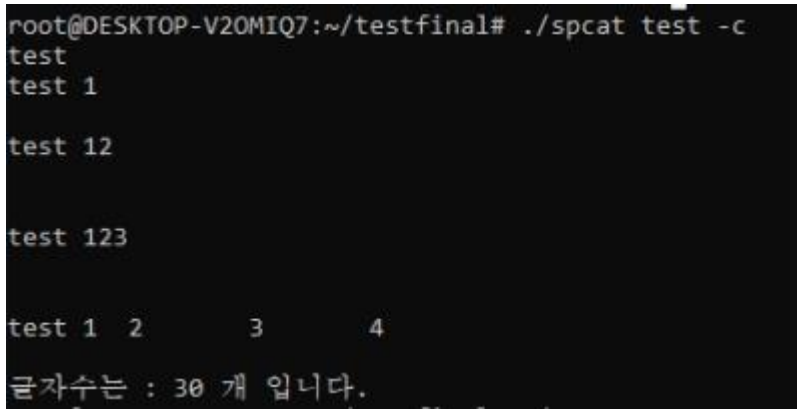
--count / -c	: 글자 수 count
--one-line / -o	: 한줄로 출력
--line-number / -l	: 라인번호 부여
--no-double-empty-lines / -d	: 두칸이상의 ₩n를 하나로 변경
--show-tap-as-caretI / -T	: tap 을 ^I문자로 변경

1. cat

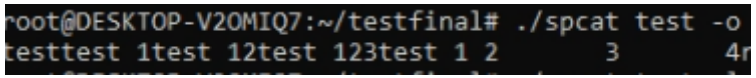
./spcat [파일명]



./spcat [파일명] -c



./spcat [파일명] -o



1. cat

./spcat [파일명] -l

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcat test -l
1: test
2: test 1
3:
4: test 12
5:
6:
7: test 123
8:
9:
10: test 1      2       3       4
```

./spcat [파일명] -d

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcat test -d
test
test 1
test 12
test 123
test 1  2       3       4
```

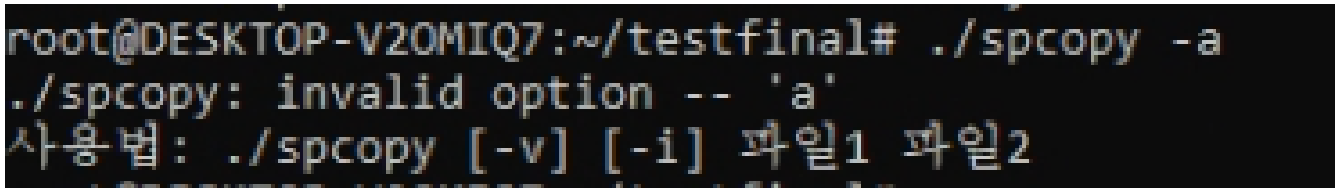
./spcat [파일명] -T

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcat test -T
test
test 1

test 12


test 123


test 1^I2^I3^I4
```

2. cp

```c
#include <stdio.h>
#include <string.h>
#include <getopt.h>

int main(int argc, char *argv[]) {
    char c;
    FILE *fp1, *fp2;
    int verbose = 0;
    int interactive = 0;
    int opt;
    static struct option long_options[] = {
        {"verbose", no_argument, NULL, 'v'},
        {"interactive", no_argument, NULL, 'i'},
        {NULL, 0, NULL, 0}
    };

    while ((opt = getopt_long(argc, argv, "vi", long_options, NULL)) != -1) {
        switch (opt) {
            case 'v':
                verbose = 1;
                break;
            case 'i':
                interactive = 1;
                break;
            default:
                fprintf(stderr, "사용법: %s [-v] [-i] 파일1 파일2₩n", argv[0]);
                return 1;
        }
    }

    if (argc - optind != 2) {
        fprintf(stderr, "사용법: %s [-v] [-i] 파일1 파일2₩n", argv[0]);
        return 1;
    }

    if (interactive) {
        char confirm;
        printf("파일 복사를 진행하시겠습니까? (y/n): ");
        scanf(" %c", &confirm);
        if (confirm != 'y' && confirm != 'Y') {
            printf("복사를 취소합니다.₩n");
            return 0;
        }
    }

    fp1 = fopen(argv[optind], "r");
    if (fp1 == NULL) {
        fprintf(stderr, "파일 %s 열기 오류₩n", argv[optind]);
        return 2;
    }

    fp2 = fopen(argv[optind + 1], "w");

    if (verbose) {
        printf("복사 시작: %s -> %s₩n", argv[optind], argv[optind + 1]);
    }

    while ((c = fgetc(fp1)) != EOF) {
        fputc(c, fp2);
    }

    fclose(fp1);
    fclose(fp2);

    if (verbose) {
        printf("복사 완료: %s -> %s₩n", argv[optind], argv[optind + 1]);
    }

    return 0;
}
```

2. cp



[spcopy]

옵션 종류

--verbose / -v : 실행결과 출력

--interactive / -i : 진행여부 질문

2. cp

./spcopy [파일명] [파일명]

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcopy test test2
root@DESKTOP-V2OMIQ7:~/testfinal# ll
-rw-r--r--  1 root root    47 Jun 12 00:17 test
-rw-r--r--  1 root root    47 Jun 12 22:08 test2
```

./spcopy [파일명] [파일명] -i

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcopy test test3 -i
파일 복사를 진행하시겠습니까? (y/n): n
복사를 취소합니다.
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcopy test test3 -i
파일 복사를 진행하시겠습니까? (y/n): y
```

./spcopy [파일명] [파일명] -v

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcopy test test3 -v
복사 시작: test -> test3
복사 완료: test -> test3
root@DESKTOP-V2OMIQ7:~/testfinal#
```

./spcopy [파일명] [파일명] -vi

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spcopy test test4 -vi
파일 복사를 진행하시겠습니까? (y/n): y
복사 시작: test -> test4
복사 완료: test -> test4
```

3. mv

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>

int main(int argc, char *argv[])
{
    int verbose = 0;
    int opt;
    int option_index;
    struct option long_options[] = {
        {"verbose", no_argument, 0, 'v'},
        {0, 0, 0, 0}
    };

    while ((opt = getopt_long(argc, argv, "v", long_options, &option_index)) != -1)
    {
        switch (opt) {
            case 'v':
                verbose = 1;
                break;
            default:
                printf("사용법: %s [-v | --verbose] [source] [destination]₩n", argv[0]);
                return 1;
        }
    }

    char *src = argv[optind];
    char *dst = argv[optind + 1];

    if (rename(src, dst) != 0)
    {
        perror("Error moving file");
        return 1;
    }

    if (verbose)
    {
        printf("'%s' -> '%s'₩n", src, dst);
    }

    return 0;
}
```

3. mv

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmv -a
./spmv: invalid option -- 'a'
사용법: ./spmv [-v | --verbose] [source] [destination]
```

[spmv]

옵션 종류

--verbose / -v : 실행결과 출력

3. mv

./spmv [파일명] [파일명]

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmv test4 test5
root@DESKTOP-V2OMIQ7:~/testfinal# ll
-rw-r--r--  1 root root    47 Jun 12 22:09 test5
```

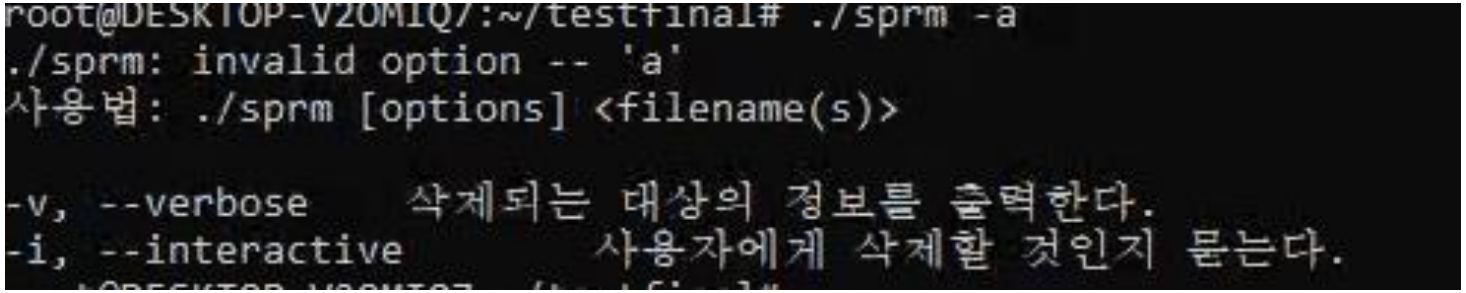
./spmv [파일명] [파일명] -v

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmv test5 test6 -v
'test5' -> 'test6'
```

4. rm

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>

int main(int argc, char *argv[]) {
    int opt;
    int verbose_flag = 0, interactive_flag = 0;

    static struct option long_options[] = {
        {"verbose", no_argument, 0, 'v'},
        {"interactive", no_argument, 0, 'i'},
        {0,0,0,0}
    };

    int long_index = 0;

    while ((opt = getopt_long(argc, argv, "vi", long_options, &long_index)) != -1) {
        switch (opt) {
            case 'v':
                verbose_flag = 1;
                break;
            case 'i':
                interactive_flag = 1;
                break;
            default:
                printf("사용법: %s [options] <filename(s)>₩n", argv[0]);
                printf("₩n-v, --verbose₩t 삭제되는 대상의 정보를 출력한다.₩n");
                printf("-i, --interactive₩t 사용자에게 삭제할 것인지 묻는다.₩n");
                return 1;
        }
    }

    for (int i = optind; i < argc; i++) {
        if (interactive_flag) {
            char user_input;
            printf("'%s' 파일을 삭제하시겠습니까? (y/n): ", argv[i]);
            scanf(" %c", &user_input);

            if (user_input != 'y' && user_input != 'Y') {
                if (verbose_flag) {
                    printf("'%s' 파일 삭제를 건너뜁니다.₩n", argv[i]);
                }
                continue;
            }
        }

        if (unlink(argv[i]) == -1) {
            perror(argv[i]);
            continue;
        }

        if (verbose_flag) {
            printf("'%s' 파일이 삭제되었습니다.₩n", argv[i]);
        }
    }

    return 0;
}
```

4. rm



[sprm]

옵션 종류

--verbose / -v : 실행결과 출력

--interactive / -i : 진행여부 질문

4. rm

./sprm [파일명]

```
-rw-r--r--  1 root root    47 Jun 12 22:09 test6
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprm test6
root@DESKTOP-V2OMIQ7:~/testfinal#
root@DESKTOP-V2OMIQ7:~/testfinal# ll
total 276
-rw-r--r--  1 root root    47 Jun 12 22:08 test2
-rw-r--r--  1 root root    47 Jun 12 22:09 test3
root@DESKTOP-V2OMIQ7:~/testfinal#
```

./sprm [파일명] -i

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprm test7 -i
'test7' 파일을 삭제하시겠습니까? (y/n): n
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprm test7 -i
'test7' 파일을 삭제하시겠습니까? (y/n): y
```

./sprm [파일명] -v

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprm test3 -v
'test3' 파일이 삭제되었습니다.
```

5. mkdir

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

int main(int argc, char *argv[])
{
    int opt;
    int verbose = 0;
    mode_t mode = 0755;
    struct option long_options[] = {
        {"verbose", no_argument, NULL, 'v'},
        {"mode", required_argument, NULL, 'm'},
        {NULL, 0, NULL, 0}
    };

    while ((opt = getopt_long(argc, argv, "vm:", long_options, NULL)) != -1) {
        switch (opt) {
            case 'v':
                verbose = 1;
                break;
            case 'm':
                mode = strtol(optarg, NULL, 8);
                break;
            default:
                fprintf(stderr, "사용법: %s [-v] [-m mode] directory_name₩n", argv[0]);
                exit(-1);
        }
    }
    int status;

    status = mkdir(argv[optind], mode);

    if (status == -1) {
        perror("mkdir");
        exit(-1);
    } else if (verbose) {
        printf("'%s' 디렉터리가 성공적으로 생성되었습니다. 권한 설정: %04o₩n", argv[optind],
mode);
    }

    return 0;
}
```

5. mkdir

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmkdir
사용법: ./spmkdir [-v] [-m mode] directory_name
```

[spmkdir]

옵션 종류

--verbose / -v : 실행결과 출력

--mode / -m : 생성시권한설정

5. mkdir

./spmkdir [폴더명]

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmkdir hello1
root@DESKTOP-V2OMIQ7:~/testfinal# ll
total 276
drwxr-xr-x  3 root root  4096 Jun 12 22:28 ./
drwx------ 15 root root  4096 Jun 12 00:40 ../
drwxr-xr-x  2 root root  4096 Jun 12 22:28 hello1/
```

./ spmkdir [폴더명] -v

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmkdir hello2 -v
'hello2' 디렉터리가 성공적으로 생성되었습니다. 권한 설정: 0755
```

./spmkdir [폴더명] –m [권한]

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmkdir hello4 -m 555
root@DESKTOP-V2OMIQ7:~/testfinal# ll
total 288
drwxr-xr-x  6 root root  4096 Jun 12 22:29 ./
drwx------ 15 root root  4096 Jun 12 00:40 ../
drwxr-xr-x  2 root root  4096 Jun 12 22:28 hello1/
drwxr-xr-x  2 root root  4096 Jun 12 22:29 hello2/
drwxr-xr-x  2 root root  4096 Jun 12 22:29 hello3/
dr-xr-xr-x  2 root root  4096 Jun 12 22:29 hello4/
```

./ spmkdir [폴더명] – vm [권한]

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spmkdir hello5 -vm 766
'hello5' 디렉터리가 성공적으로 생성되었습니다. 권한 설정: 0766
root@DESKTOP-V2OMIQ7:~/testfinal# ll
total 292
drwxr-xr-x  7 root root  4096 Jun 12 22:30 ./
drwx------ 15 root root  4096 Jun 12 00:40 ../
drwxr-xr-x  2 root root  4096 Jun 12 22:28 hello1/
drwxr-xr-x  2 root root  4096 Jun 12 22:29 hello2/
drwxr-xr-x  2 root root  4096 Jun 12 22:29 hello3/
dr-xr-xr-x  2 root root  4096 Jun 12 22:29 hello4/
drwxr--r--  2 root root  4096 Jun 12 22:30 hello5/
```

6. rmdir

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>

int main(int argc, char *argv[])
{
    int opt;
    int verbose = 0;
    struct option long_options[] = {
        {"verbose", no_argument, NULL, 'v'},
        {NULL, 0, NULL, 0}
    };

    while ((opt = getopt_long(argc, argv, "v", long_options, NULL)) != -1) {
        switch (opt) {
            case 'v':
                verbose = 1;
                break;
            default:
                fprintf(stderr, "사용법: %s [-v] directory_name₩n", argv[0]);
                exit(-1);
        }
    }
    int status;

    status = rmdir(argv[optind]);

    if (status == -1) {
        perror("rmdir");
        exit(-1);
    } else if (verbose) {
        printf("'%s' 디렉터리가 성공적으로 삭제되었습니다.₩n", argv[optind]);
    }

    return 0;
}
```

6. rmdir

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprmdir -a
./sprmdir: invalid option -- 'a'
사용법: ./sprmdir [-v] directory_name
```

[sprmdir]

옵션 종류

--verbose / -v : 실행결과 출력

6. rmdir

./ rmdir [폴더명]

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprmdir hello5
root@DESKTOP-V2OMIQ7:~/testfinal# ll
total 288
drwxr-xr-x  6 root root  4096 Jun 12 22:36 ./
drwx------ 15 root root  4096 Jun 12 00:40 ../
drwxr-xr-x  2 root root  4096 Jun 12 22:28 hello1/
drwxr-xr-x  2 root root  4096 Jun 12 22:29 hello2/
drwxr-xr-x  2 root root  4096 Jun 12 22:29 hello3/
dr-xr-xr-x  2 root root  4096 Jun 12 22:29 hello4/
-rwxr-xr-x  1 root root 16560 Jun 12 00:33 spcat*
```

./ rmdir [폴더명] -v

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./sprmdir hello4 -v
'hello4' 디렉터리가 성공적으로 삭제되었습니다.
root@DESKTOP-V2OMIQ7:~/testfinal#
```

7. chmod

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <getopt.h>
#include <string.h>
#include <stdbool.h>

int main(int argc, char *argv[])
{
    long strtol();
    struct option long_options[] = {
        {"verbose", no_argument, NULL, 'v'},
        {0, 0, 0, 0}};

    int option_index = 0;
    int c;
    bool verbose_output = false;
    while ((c = getopt_long(argc, argv, "v", long_options, &option_index)) != -1)
    {
        switch (c)
        {
        case 'v':
            verbose_output = true;
            break;
        default:
            printf("사용법: %s [-v|--verbose] MODE FILE₩n", argv[0]);
            return 1;
        }
    }

    int newmode = (int)strtol(argv[optind], (char **)NULL, 8);
    struct stat st;
    if (stat(argv[optind + 1], &st) == -1)
    {
        perror(argv[optind + 1]);
        return 1;
    }

    mode_t oldmode = st.st_mode;
    if (chmod(argv[optind + 1], newmode) == -1)
    {
        perror(argv[optind + 1]);
        return 1;
    }

    if (verbose_output)
    {
        printf("mode of '%s' changed from %o to %o₩n", argv[optind + 1], oldmode & 07777,
newmode);
    }

    return 0;
}
```

7. chmod

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spchmod -a
./spchmod: invalid option -- 'a'
사용법: ./spchmod [-v|--verbose] MODE FILE
```

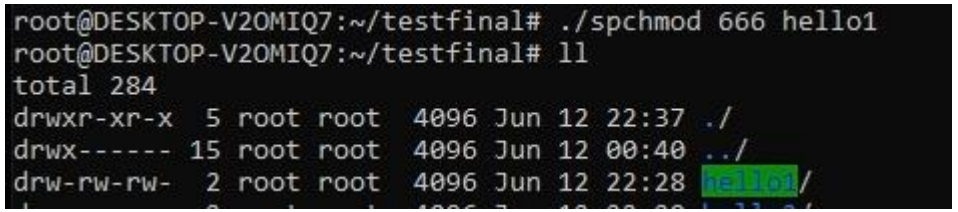
[spchmod]

옵션 종류

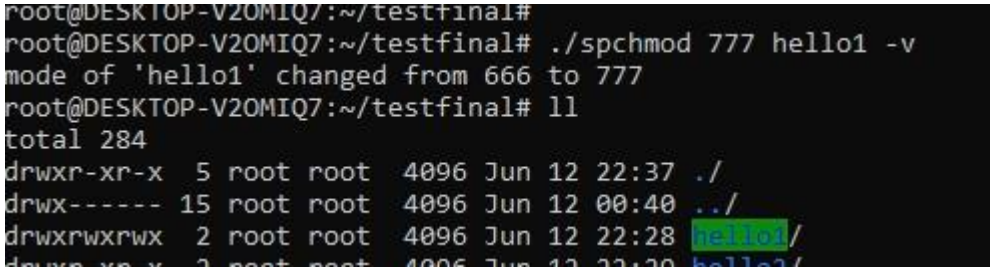
--verbose / -v : 실행결과 출력

7. chmod

./spchmod [파일명]



./spchmod [파일명] -v

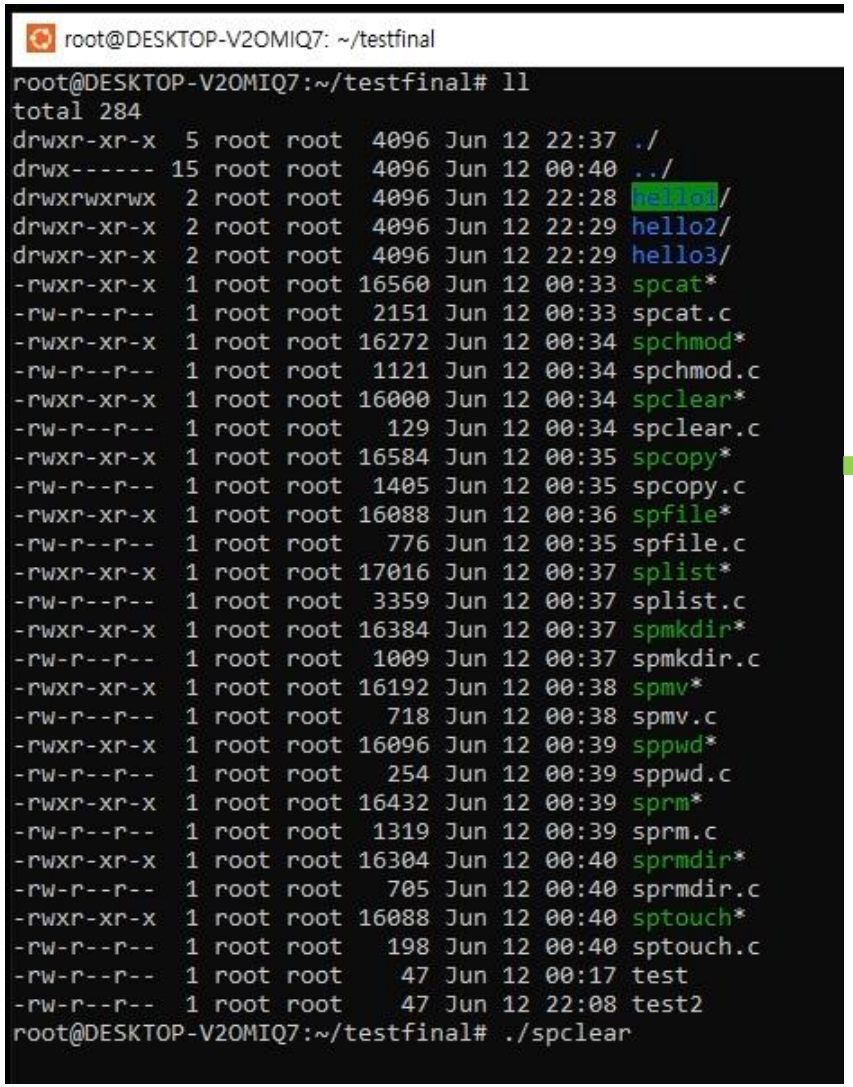


8. clear

```
#include <stdio.h>

void clear_screen()
{
    printf("₩033[H₩033[J");
}

int main()
{
    clear_screen();
    return 0;
}
```

8. clear

./spclear

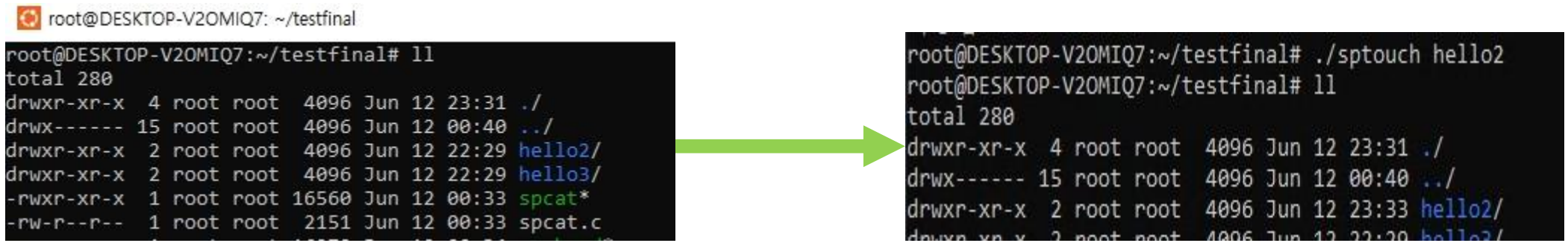


9. touch

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

int main(int argc, char *argv[])

{
        if (argc < 2) {
        fprintf(stderr, "사용법: touch file1 ₩n");
        exit(-1);
        }

        utime(argv[1], NULL);
}
```

9. touch

./sptouch [파일명]



10. file

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

int main(int argc, char * argv[]) {
    int i;
    struct stat buf;
    for (i = 1; i < argc; i++) {
        printf("%s: ", argv[i]);
        if (lstat(argv[i], & buf) < 0) {
            perror("lstat()");
            continue;
        }
        if (S_ISREG(buf.st_mode))
            printf("%s ₩n", "일반 파일");
        if (S_ISDIR(buf.st_mode))
            printf("%s ₩n", "디렉터리");
        if (S_ISCHR(buf.st_mode))
            printf("%s ₩n", "문자 장치 파일");
        if (S_ISBLK(buf.st_mode))
            printf("%s ₩n", "블록 장치 파일");
        if (S_ISFIFO(buf.st_mode))
            printf("%s ₩n", "FIFO 파일");
        if (S_ISLNK(buf.st_mode))
            printf("%s ₩n", "심볼릭 링크");
        if (S_ISSOCK(buf.st_mode))
            printf("%s ₩n", "소켓");
    }
    exit(0);
}
```

10. file

./spfile [파일명]

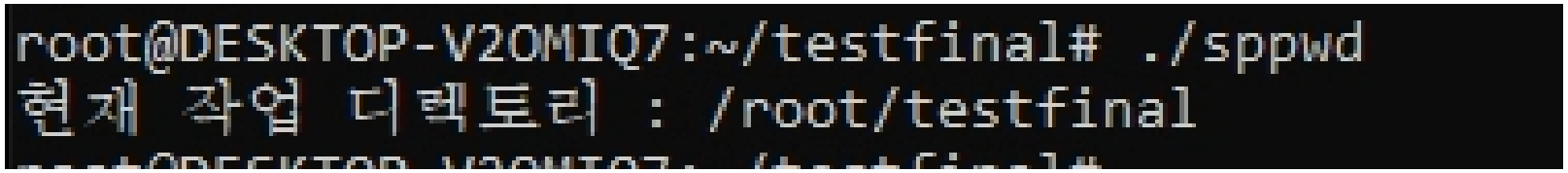
```
root@DESKTOP-V2OMIQ7:~/testfinal# ./spfile test
test: 일반 파일
root@DESKTOP-V2OMIQ7:~/testfinal# ./spfile hello2
hello2: 디렉터리
```

11. pwd

```
#include <stdio.h>
#include <unistd.h>
#include <limits.h>

int main() {
    char cwd[PATH_MAX];
    if (getcwd(cwd, sizeof(cwd)) != NULL) {
        printf("현재 작업 디렉토리 : %s₩n", cwd);
    } else {
        perror("getcwd() 에러");
        return 1;
    }
    return 0;
}
```

11. pwd

./sppwd


root@DESKTOP-V2OMIQ7:~/testfinal# ./sppwd
현재 작업 디렉토리 : /root/testfinal

12. list

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>
#include <string.h>

void display_file_properties(struct stat file_stat);

int compare(const void *a, const void *b) {
    return strcmp(*(const char **)a, *(const char **)b);
}

int main(int argc, char **argv) {
    DIR *dp;
    char *dir;
    struct dirent *d;
    struct stat st;
    char path[BUFSIZ+1];
    int long_option;
    int l_flag = 0;
    int b_flag = 0;
    int a_flag = 0;
    int i_flag = 0;
    int file_count = 0;
    int i;

    static struct option long_options[] = {
        {"long", no_argument, 0, 'l'},
        {"alpha", no_argument, 0, 'b'},
        {"almost-all", no_argument, 0, 'A'},
        {"inode", no_argument, 0, 'i'},
        {0, 0, 0, 0}
    };

    while ((long_option = getopt_long(argc, argv, "lbAi", long_options, 0)) != -1) {
        switch (long_option) {
            case 'l':
                l_flag = 1;
                break;
            case 'b':
                b_flag = 1;
                break;
            case 'A':
                a_flag = 1;
                break;
            case 'i':
                i_flag = 1;
                break;
            default:
                printf("사용법: %s [options] <directory>₩n", argv[0]);
                printf("₩n-l, --long₩t 파일의 자세한 정보를 출력한다.₩n");
                printf("-b, --alpha₩t 파일 이름을 알파벳 순서대로 정렬하여 출력한다.₩n");
                printf("-A, --almost-all₩t .(현재 디렉터리)와 ..(상위 디렉터리)를 제외한 거의 모든 항목을 출력한다.₩n");
                printf("-i, --inode₩t 각 파일의 inode 번호를 함께 출력한다.₩n");
        }
    }

    if (argc - optind == 0) {
        dir = ".";
    } else {
        dir = argv[optind];
    }

    if ((dp = opendir(dir)) == NULL) {
        perror(dir);
    }

    while ((d = readdir(dp)) != NULL) {
        if (a_flag && (strcmp(d->d_name, ".") == 0 || strcmp(d->d_name, "..") == 0)) {
            continue;
        }
        file_count++;
    }

    char **file_list = malloc(file_count * sizeof(char *));
    rewinddir(dp);
    i = 0;

    while ((d = readdir(dp)) != NULL) {
        if (a_flag && (strcmp(d->d_name, ".") == 0 || strcmp(d->d_name, "..") == 0)) {
            continue;
        }
        file_list[i] = malloc(strlen(d->d_name) + 1);
        strcpy(file_list[i], d->d_name);
        i++;
    }

    if (b_flag) {
        qsort(file_list, file_count, sizeof(char *), compare);
    }

    for (i = 0; i < file_count; i++) {
        if (l_flag) {
            sprintf(path, "%s/%s", dir, file_list[i]);
            if (stat(path, &st) == -1) {
                perror(path);
            } else {
                display_file_properties(st);
            }
        }
        if (i_flag) {
            sprintf(path, "%s/%s", dir, file_list[i]);
            if (stat(path, &st) == -1) {
                perror(path);
            } else {
                printf("%lu ", (unsigned long)st.st_ino);
            }
        }
        printf("%s₩n", file_list[i]);
    }

    for (i = 0; i < file_count; i++) {
        free(file_list[i]);
    }
    free(file_list);
    closedir(dp);
    exit(0);
}

void display_file_properties(struct stat file_stat) {
    printf((file_stat.st_mode & S_IRUSR) ? "r" : "-");
    printf((file_stat.st_mode & S_IWUSR) ? "w" : "-");
    printf((file_stat.st_mode & S_IXUSR) ? "x" : "-");
    printf((file_stat.st_mode & S_IRGRP) ? "r" : "-");
    printf((file_stat.st_mode & S_IWGRP) ? "w" : "-");
    printf((file_stat.st_mode & S_IXGRP) ? "x" : "-");
    printf((file_stat.st_mode & S_IROTH) ? "r" : "-");
    printf((file_stat.st_mode & S_IWOTH) ? "w" : "-");
    printf((file_stat.st_mode & S_IXOTH) ? "x" : "-");

    printf(" %ld ", file_stat.st_nlink);
    printf("%d %d ", file_stat.st_uid, file_stat.st_gid);
    printf("%ld ", file_stat.st_size);

    printf(" ");
}
```

12. list

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist -Aadfads
./splist: invalid option -- 'a'
사용법: ./splist [options] <directory>

-l, --long       파일의 자세한 정보를 출력한다.
-b, --alpha      파일 이름을 알파벳 순서대로 정렬하여 출력한다.
-A, --almost-all         .(현재 디렉터리)와 ..(상위 디렉터리)를 제외한 거의 모든 항목을 출력한다.
-i, --inode      각 파일의 inode 번호를 함께 출력한다.
./splist: invalid option -- 'd'
사용법: ./splist [options] <directory>
```

[splist]

옵션 종류

--long / -l : 자세한 정보 출력

--alpha / -b : 알파벳 순으로 출력

--almost-all / -A : .(현재 디렉토리)와 ..(상위디렉토리)를 제외하여 출력

--inode / -I : 각 파일의 inode 번호를 함께 출력

12. list

./splist

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist
spfile.c
spcat
spcopy.c
splist.c
spcopy
spchmod.c
spmkdir
hello2
splist
hello3
sprm.c
sptouch
sprm
sppwd.c
spmv
spclear
spmv.c
test
spcat.c
sptouch.c
sppwd
spchmod
spmkdir.c
spclear.c
..
sprmdir
.
spfile
test2
sprmdir.c
root@DESKTOP-V2OMIQ7:~/testfinal#
```

./splist -l

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist -l
rw-r--r-- 1 0 0 776  spfile.c
rwxr-xr-x 1 0 0 16560  spcat
rw-r--r-- 1 0 0 1405  spcopy.c
rw-r--r-- 1 0 0 3359  splist.c
rwxr-xr-x 1 0 0 16584  spcopy
rw-r--r-- 1 0 0 1121  spchmod.c
rwxr-xr-x 1 0 0 16384  spmkdir
rwxr-xr-x 2 0 0 4096  hello2
rwxr-xr-x 1 0 0 17016  splist
rwxr-xr-x 2 0 0 4096  hello3
rw-r--r-- 1 0 0 1319  sprm.c
rwxr-xr-x 1 0 0 16088  sptouch
rwxr-xr-x 1 0 0 16432  sprm
rw-r--r-- 1 0 0 254  sppwd.c
rwxr-xr-x 1 0 0 16192  spmv
rwxr-xr-x 1 0 0 16000  spclear
rw-r--r-- 1 0 0 718  spmv.c
rw-r--r-- 1 0 0 47  test
rw-r--r-- 1 0 0 2151  spcat.c
rw-r--r-- 1 0 0 198  sptouch.c
rwxr-xr-x 1 0 0 16096  sppwd
rwxr-xr-x 1 0 0 16272  spchmod
rw-r--r-- 1 0 0 1009  spmkdir.c
rw-r--r-- 1 0 0 129  spclear.c
rwx------ 15 0 0 4096  ..
rwxr-xr-x 1 0 0 16304  sprmdir
rwxr-xr-x 4 0 0 4096  .
rwxr-xr-x 1 0 0 16088  spfile
rw-r--r-- 1 0 0 47  test2
rw-r--r-- 1 0 0 705  sprmdir.c
```

12. list

./splist -b

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist -b
.
..
hello2
hello3
spcat
spcat.c
spchmod
spchmod.c
spclear
spclear.c
spcopy
spcopy.c
spfile
spfile.c
splist
splist.c
spmkdir
spmkdir.c
spmv
spmv.c
sppwd
sppwd.c
sprm
sprm.c
sprmdir
sprmdir.c
sptouch
sptouch.c
test
test2
```

./splist -i

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist -i
34653 spfile.c
34655 spcat
34652 spcopy.c
34654 splist.c
34661 spcopy
34650 spchmod.c
34667 spmkdir
34672 hello2
34665 splist
34674 hello3
34662 sprm.c
34677 sptouch
34673 sprm
34660 sppwd.c
34669 spmv
34659 spclear
34658 spmv.c
34646 test
34644 spcat.c
34666 sptouch.c
34671 sppwd
34657 spchmod
34656 spmkdir.c
34648 spclear.c
40961 ..
34675 sprmdir
34622 .
34663 spfile
34513 test2
34664 sprmdir.c
```

12. list

./splist -A

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist -A
spfile.c
spcat
spcopy.c
splist.c
spcopy
spchmod.c
spmkdir
hello2
splist
hello3
sprm.c
sptouch
sprm
sppwd.c
spmv
spclear
spmv.c
test
spcat.c
sptouch.c
sppwd
spchmod
spmkdir.c
spclear.c
sprmdir
spfile
test2
sprmdir.c
root@DESKTOP-V2OMIQ7:~/testfinal#
```

./splist -lbAi

```
root@DESKTOP-V2OMIQ7:~/testfinal# ./splist -lbAi
rwxr-xr-x 2 0 0 4096  34672 hello2
rwxr-xr-x 2 0 0 4096  34674 hello3
rwxr-xr-x 1 0 0 16560  34655 spcat
rw-r--r-- 1 0 0 2151  34644 spcat.c
rwxr-xr-x 1 0 0 16272  34657 spchmod
rw-r--r-- 1 0 0 1121  34650 spchmod.c
rwxr-xr-x 1 0 0 16000  34659 spclear
rw-r--r-- 1 0 0 129  34648 spclear.c
rwxr-xr-x 1 0 0 16584  34661 spcopy
rw-r--r-- 1 0 0 1405  34652 spcopy.c
rwxr-xr-x 1 0 0 16088  34663 spfile
rw-r--r-- 1 0 0 776  34653 spfile.c
rwxr-xr-x 1 0 0 17016  34665 splist
rw-r--r-- 1 0 0 3359  34654 splist.c
rwxr-xr-x 1 0 0 16384  34667 spmkdir
rw-r--r-- 1 0 0 1009  34656 spmkdir.c
rwxr-xr-x 1 0 0 16192  34669 spmv
rw-r--r-- 1 0 0 718  34658 spmv.c
rwxr-xr-x 1 0 0 16096  34671 sppwd
rw-r--r-- 1 0 0 254  34660 sppwd.c
rwxr-xr-x 1 0 0 16432  34673 sprm
rw-r--r-- 1 0 0 1319  34662 sprm.c
rwxr-xr-x 1 0 0 16304  34675 sprmdir
rw-r--r-- 1 0 0 705  34664 sprmdir.c
rwxr-xr-x 1 0 0 16088  34677 sptouch
rw-r--r-- 1 0 0 198  34666 sptouch.c
rw-r--r-- 1 0 0 47  34646 test
rw-r--r-- 1 0 0 47  34513 test2
root@DESKTOP-V2OMIQ7:~/testfinal#
```

4. 출처

ai.com

https://wrtn.ai/

교수님강의자료 및 학우님들 발표자료

https://server-talk.tistory.com/394

https://coding-factory.tistory.com/748

https://coding-factory.tistory.com/751

https://shaeod.tistory.com/663

https://iksflow.tistory.com/83

https://rhrhth23.tistory.com/47

https://www.lesstif.com/lpt/directory-linux-mkdir-95880384.html

https://coding-factory.tistory.com/753

https://rhrhth23.tistory.com/19

https://junshim.github.io/linux%20kernel%20study/Add_a_New_System_Call/

https://leeeeye321.tistory.com/222

https://jframework.tistory.com/6

목차