

# 시스템 프로그래밍

2019975034 신기윤

# Cat 명령어 ( -E, -T, -n)

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./cat a.txt  
1  
2  
3  
4  
5
```

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./cat -E a.txt  
1  
$2  
$3  
$4  
$5
```

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./cat -T b.txt  
1^2^3  
4^5^6
```

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./cat -n a.txt  
1 1  
2 2  
3 3  
4 4  
5 5
```

```

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

int main(int argc, char *argv[]) {
    int option;
    int showEnds = 0;
    int showTabs = 0;
    int numberNonEmpty = 0;
    int numberAll = 0;

    while ((option = getopt(argc, argv, "ETbn")) != -1) {
        switch (option) {
            case 'E':
                showEnds = 1;
                break;
            case 'T':
                showTabs = 1;
                break;
            case 'b':
                numberNonEmpty = 1;
                break;
            case 'n':
                numberAll = 1;
                break;
            case '?':
                if (optopt == 'E' || optopt == 'T' || optopt == 'b' || optopt == 'n') {
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }

    // Process remaining non-option arguments (input files)
    for (int i = optind; i < argc; i++) {
        FILE *file = fopen(argv[i], "r");
        if (file == NULL) {
            fprintf(stderr, "Cannot open file: %s\n", argv[i]);
            continue;
        }

        int lineNumber = 1;
        char line[256];
        while (fgets(line, sizeof(line), file)) {
            if (showTabs) {
                int j = 0;
                while (line[j] != '\0') {
                    if (line[j] == '\t') {
                        line[j] = '^';
                    }
                    j++;
                }
            }

            if (numberNonEmpty && line[0] == '\n') {
                printf("%s", line);
                continue;
            }

            if (numberAll) {
                printf("%6d  ", lineNumber);
            }

            printf("%s", line);

            if (showEnds && line[0] != '\n') {
                printf("$");
            }

            lineNumber++;
        }

        fclose(file);
    }

    return 0;
}

```

# Chmod 명령어

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./chmod 777 b.txt
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ll
total 372
drwxr-xr-x  3 shingiyoun shingiyoun 4096 Jun 13 20:40 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 20:40 ../
-rw-r--r--  1 shingiyoun shingiyoun   21 Jun 13 20:05 a.txt
drwxr-xr-x  3 shingiyoun shingiyoun 4096 Jun 13 18:58 abc/
-rw-r--r--  1 shingiyoun shingiyoun    0 Jun 13 19:34 aqw
-rwxrwxrwx  1 shingiyoun shingiyoun    0 Jun 13 19:20 b.txt*
```

```

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

int main(int argc, char *argv[]) {
    int option;
    mode_t permissions = 0;
    int recursive = 0;

    while ((option = getopt(argc, argv, "R")) != -1) {
        switch (option) {
            case 'R':
                recursive = 1;
                break;
            case '?':
                if (optopt == 'R') {
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }
}

```

```

// Process remaining non-option arguments (permissions and files/directories)
int numArgs = argc - optind;
if (numArgs < 2) {
    fprintf(stderr, "Usage: %s [-R] <permissions> <file1> [<file2>...]\n", argv[0]);
    return 1;
}

// Parse permissions string
if (sscanf(argv[optind], "%o", &permissions) != 1) {
    fprintf(stderr, "Invalid permissions: %s\n", argv[optind]);
    return 1;
}

// Apply permissions to files/directories
for (int i = optind + 1; i < argc; i++) {
    if (recursive) {
        if (chmod(argv[i], permissions) != 0) {
            fprintf(stderr, "Failed to change permissions of: %s\n", argv[i]);
        }
    }
    else {
        struct stat st;
        if (stat(argv[i], &st) == 0) {
            mode_t newPermissions = (st.st_mode & ~0777) | permissions;
            if (chmod(argv[i], newPermissions) != 0) {
                fprintf(stderr, "Failed to change permissions of: %s\n", argv[i]);
            }
        }
        else {
            fprintf(stderr, "Failed to get file/directory information: %s\n", argv[i]);
        }
    }
}

```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll
total 92
drwxr-xr-x  2 shingiyoun shingiyoun 4096 Jun 13 18:27 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 18:27 ../
-rwxrwxrwx  1 shingiyoun shingiyoun  10 Jun 13 18:07 a.txt*
-rwxrwxrwx  1 shingiyoun shingiyoun  12 Jun 13 18:15 b.txt*
-rw-r--r--  1 shingiyoun shingiyoun  15 Jun 13 18:18 c.txt
-rwxr-xr-x  1 shingiyoun shingiyoun 16464 Jun 13 18:07 cat*
-rw-r--r--  1 shingiyoun shingiyoun  2232 Jun 13 18:06 cat.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16408 Jun 13 18:23 chmod*
-rw-r--r--  1 shingiyoun shingiyoun  2097 Jun 13 18:23 chmod.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16208 Jun 13 18:27 clear*
-rw-r--r--  1 shingiyoun shingiyoun  1086 Jun 13 18:27 clear.c
-rw-r--r--  1 shingiyoun shingiyoun   10 Jun 13 18:19 d.txt
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./clear
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ |
```

# Clear 명령어

```

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

int main(int argc, char *argv[]) {
    int option;
    int preserveScreen = 0;

    while ((option = getopt(argc, argv, "x")) != -1) {
        switch (option) {
            case 'x':
                preserveScreen = 1;
                break;
            case '?':
                if (optopt == 'x') {
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }

    if (preserveScreen) {
        printf("\e[2J"); // Clear entire screen and move cursor to top-left corner
    }
    else {
        printf("\e[H\e[2J"); // Clear entire screen and move cursor to top-left corner
    }

    return 0;
}

```

# Cp (-i, -f, -v,)

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./cp a.txt c.txt
shingiyoun@DESKTOP-7SFPOVE:~/exam$ cat a.txt
1
2
3
4
5
shingiyoun@DESKTOP-7SFPOVE:~/exam$ cat c.txt
1
2
3
4
5
shingiyoun@DESKTOP-7SFPOVE:~/exam$
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./cp -i a.txt b.txt
cp: overwrite 'b.txt'?y
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./cp -f a.txt b.txt
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./cp -v a.txt b.txt
'a.txt' -> 'b.txt'
```



```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>
#include <fcntl.h>
#include <ctype.h>

int main(int argc, char *argv[]) {
    int option;
    int force = 0;
    int interactive = 0;
    int verbose = 0;

    while ((option = getopt(argc, argv, "fiV")) != -1) {
        switch (option) {
            case 'f':
                force = 1;
                break;
            case 'i':
                interactive = 1;
                break;
            case 'v':
                verbose = 1;
                break;
            case '?':
                if (optopt == 'f' || optopt == 'i' || optopt == 'v') {
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }

    // Process remaining non-option arguments (source and destination files)
    if (argc - optind != 2) {
        fprintf(stderr, "Usage: %s [-f] [-i] [-v] <source> <destination>\n", argv[0]);
        return 1;
    }

    char *source = argv[optind];
    char *destination = argv[optind + 1];

    if (force || interactive) {
        if (interactive) {
            printf("cp: overwrite '%s'?", destination);
            char response;
            scanf("%c", &response);
            if (response != 'y' && response != 'Y') {
                printf("cp: not overwritten\n");
                return 0;
            }
        }
    }

    if (verbose) {
        printf("'%s' -> '%s'\n", source, destination);
    }
    else {
        FILE *fileSource = fopen(source, "r");
        if (fileSource == NULL) {
            fprintf(stderr, "Failed to open source file: %s\n", source);
            return 1;
        }

        FILE *fileDest = fopen(destination, "w");
        if (fileDest == NULL) {
            fprintf(stderr, "Failed to open destination file: %s\n", destination);
            return 1;
        }

        int ch;
        while ((ch = fgetc(fileSource)) != EOF) {
            fputc(ch, fileDest);
        }

        fclose(fileSource);
        fclose(fileDest);

        if (verbose) {
            printf("'%s' -> '%s'\n", source, destination);
        }
    }

    return 0;
}

```

# Echo 명령어

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./echo "안녕하세요"
```

안녕하세요

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>
#include <ctype.h>
int main(int argc, char *argv[]) {
    int option;
    int noNewLine = 0;

    while ((option = getopt(argc, argv, "n")) != -1) {
        switch (option) {
            case 'n':
                noNewLine = 1;
                break;
            case '?':
                if (optopt == 'n') {
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }

    for (int i = optind; i < argc; i++) {
        printf("%s ", argv[i]);
    }

    if (!noNewLine) {
        printf("\n");
    }

    return 0;
}

```

# Head (-n) 명령어

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./head a.txt  
==> a.txt <==  
1  
2  
3  
4  
5
```

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./head -n 4 a.txt  
==> a.txt <==  
1  
2  
3  
4
```

```

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

int main(int argc, char *argv[]) {
    int option;
    int lines = 10; // 기본적으로 10줄만 출력

    while ((option = getopt(argc, argv, "n:")) != -1) {
        switch (option) {
            case 'n':
                lines = atoi(optarg);
                break;
            case '?':
                if (optopt == 'n') {
                    fprintf(stderr, "Option -%c requires an argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }

    // Process remaining non-option arguments (file names)
    int fileIndex = optind;
    for (int i = fileIndex; i < argc; i++) {
        FILE *file = fopen(argv[i], "r");
        if (file == NULL) {
            fprintf(stderr, "Failed to open file: %s\n", argv[i]);
            continue;
        }
    }
}

```

```

        printf("==> %s <==\n", argv[i]);

        int lineCount = 0;
        int ch;
        while ((ch = fgetc(file)) != EOF) {
            putchar(ch);
            if (ch == '\n') {
                lineCount++;
                if (lineCount >= lines) {
                    break;
                }
            }
        }

        fclose(file);
        printf("\n");
    }

    return 0;
}

```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./mkdir -v abb
Created directory: abb
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll
total 196
drwxr-xr-x  6 shingiyoun shingiyoun 4096 Jun 13 18:54 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 18:53 ../
-rwxrwxrwx  1 shingiyoun shingiyoun   10 Jun 13 18:07 a.txt*
drwxr-xr-x  2 shingiyoun shingiyoun 4096 Jun 13 18:54 aaa/
drwxr-xr-x  2 shingiyoun shingiyoun 4096 Jun 13 18:54 abb/
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./mkdir aaa
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll
total 192
drwxr-xr-x  5 shingiyoun shingiyoun 4096 Jun 13 18:54 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 18:53 ../
-rwxrwxrwx  1 shingiyoun shingiyoun   10 Jun 13 18:07 a.txt*
drwxr-xr-x  2 shingiyoun shingiyoun 4096 Jun 13 18:54 aaa/
```

# Mkdir(-v)명령어

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>
#include <sys/stat.h>
#include <ctype.h>
int main(int argc, char *argv[]) {
    int option;
    int verbose = 0; // 기본적으로는 verbose 모드 비활성화

    while ((option = getopt(argc, argv, "v")) != -1) {
        switch (option) {
            case 'v':
                verbose = 1; // verbose 모드 활성화
                break;
            case '?':
                if (optopt == 'v') {
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                }
                else if (isprint(optopt)) {
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                }
                else {
                    fprintf(stderr, "Unknown option character `\\%c'.\n", optopt);
                }
                return 1;
            default:
                abort();
        }
    }

    // Process remaining non-option arguments (directory names)
    int dirIndex = optind;
    for (int i = dirIndex; i < argc; i++) {
        if (mkdir(argv[i], 0777) != 0) {

            perror("Failed to create directory");
        }
        else if (verbose) {
            printf("Created directory: %s\n", argv[i]);
        }
    }

    return 0;
}

```

# Pwd 명령어

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./pwd  
Current working directory: /home/shingiyoun/exam
```



```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <ctype.h>
int main(int argc, char *argv[]) {
    int c;
    char *path = NULL;

    while ((c = getopt(argc, argv, "p:")) != -1) {
        switch (c) {
            case 'p':
                path = optarg;
                break;
            case '?':
                if (optopt == 'p')
                    fprintf(stderr, "Option -%c requires an argument.\n", optopt);
                else if (isprint(optopt))
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                else
                    fprintf(stderr, "Unknown option character `\\x%x'.\n", optopt);
                return 1;
            default:
                abort();
        }
    }

    if (path != NULL) {
        printf("Path: %s\n", path);
        // 여기에서 해당 경로에 대한 작업을 수행합니다.
    } else {
        // 현재 작업 디렉토리를 얻어와서 출력합니다.
        char cwd[1024];
        if (getcwd(cwd, sizeof(cwd)) != NULL) {
            printf("Current working directory: %s\n", cwd);
        } else {
            perror("getcwd() error");
            return 1;
        }
    }

    return 0;
}

```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./rm -v a.txt
```

```
Removed file: a.txt
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./rm -i c.txt
```

```
Remove file 'c.txt'? (y/n): y
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll
```

```
total 240
```

```
drwxr-xr-x  5 shingiyoun shingiyoun 4096 Jun 13 19:20 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 19:18 ../
-rw-r--r--  1 shingiyoun shingiyoun   0 Jun 13 19:20 a.txt
drwxr-xr-x  2 shingiyoun shingiyoun 4096 Jun 13 18:54 aaa/
drwxr-xr-x  3 shingiyoun shingiyoun 4096 Jun 13 18:58 abc/
-rw-r--r--  1 shingiyoun shingiyoun   0 Jun 13 19:20 b.txt
-rwxr-xr-x  1 shingiyoun shingiyoun 16464 Jun 13 18:07 cat*
-rw-r--r--  1 shingiyoun shingiyoun  2232 Jun 13 18:06 cat.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16408 Jun 13 18:23 chmod*
-rw-r--r--  1 shingiyoun shingiyoun  2097 Jun 13 18:23 chmod.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16208 Jun 13 18:27 clear*
-rw-r--r--  1 shingiyoun shingiyoun  1086 Jun 13 18:27 clear.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16688 Jun 13 18:36 cp*
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./rm -i c.txt
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll
```

```
total 240
```

```
drwxr-xr-x  5 shingiyoun shingiyoun 4096 Jun 13 19:19 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 19:18 ../
drwxr-xr-x  2 shingiyoun shingiyoun 4096 Jun 13 18:54 aaa/
drwxr-xr-x  3 shingiyoun shingiyoun 4096 Jun 13 18:58 abc/
-rwxr-xr-x  1 shingiyoun shingiyoun 16464 Jun 13 18:07 cat*
-rw-r--r--  1 shingiyoun shingiyoun  2232 Jun 13 18:06 cat.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16408 Jun 13 18:23 chmod*
-rw-r--r--  1 shingiyoun shingiyoun  2097 Jun 13 18:23 chmod.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16208 Jun 13 18:27 clear*
-rw-r--r--  1 shingiyoun shingiyoun  1086 Jun 13 18:27 clear.c
-rwxr-xr-x  1 shingiyoun shingiyoun 16688 Jun 13 18:36 cp*
-rw-r--r--  1 shingiyoun shingiyoun  3272 Jun 13 18:36 cp.c
-rw-r--r--  1 shingiyoun shingiyoun   10 Jun 13 18:19 d.txt
```

# Rm (-i, -v) 명령어

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>
int main(int argc, char *argv[]) {
    int c;
    int interactive = 0;
    int verbose = 0;

    while ((c = getopt(argc, argv, "iv")) != -1) {
        switch (c) {
            case 'i':
                interactive = 1;
                break;
            case 'v':
                verbose = 1;
                break;
            case '?':
                if (optopt == 'i' || optopt == 'v')
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                else
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                return 1;
            default:
                abort();
        }
    }
}

```

```

// getopt가 분석한 옵션들을 제외한 나머지 인자들을 출력합니다.
for (int i = optind; i < argc; ++i) {
    if (interactive) {
        char response;
        printf("Remove file '%s'? (y/n): ", argv[i]);
        scanf(" %c", &response);
        if (response == 'y' || response == 'Y') {
            if (remove(argv[i]) == 0 && verbose)
                printf("Removed file: %s\n", argv[i]);
            else if (verbose)
                printf("Failed to remove file: %s\n", argv[i]);
        }
    } else {
        if (remove(argv[i]) == 0 && verbose)
            printf("Removed file: %s\n", argv[i]);
        else if (verbose)
            printf("Failed to remove file: %s\n", argv[i]);
    }
}

return 0;
}

```

# rmmdir (-i, -v) 명령어

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./rmmdir aaa
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll
total 256
drwxr-xr-x  4 shingiyoun shingiyoun 4096 Jun 13 19:24 ./
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 19:23 ../
drwxr-xr-x  3 shingiyoun shingiyoun 4096 Jun 13 18:58 abc/
-rw-r--r--  1 shingiyoun shingiyoun   0 Jun 13 19:20 b.txt
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./rmmdir -i qwer
Remove directory 'qwer'? (y/n): y
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./rmmdir -v qwe
Removed directory: qwe
```

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <ctype.h>
int main(int argc, char *argv[]) {
    int c;
    int interactive = 0;
    int verbose = 0;

    while ((c = getopt(argc, argv, "iv")) != -1) {
        switch (c) {
            case 'i':
                interactive = 1;
                break;
            case 'v':
                verbose = 1;
                break;
            case '?':
                if (optopt == 'i' || optopt == 'v')
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                else
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                return 1;
            default:
                abort();
        }
    }
}

```

```

for (int i = optind; i < argc; ++i) {
    if (interactive) {
        char response;
        printf("Remove directory '%s'? (y/n): ", argv[i]);
        scanf(" %c", &response);
        if (response == 'y' || response == 'Y') {
            if (rmdir(argv[i]) == 0 && verbose)
                printf("Removed directory: %s\n", argv[i]);
            else if (verbose)
                printf("Failed to remove directory: %s\n", argv[i]);
        }
    } else {
        if (rmdir(argv[i]) == 0 && verbose)
            printf("Removed directory: %s\n", argv[i]);
        else if (verbose)
            printf("Failed to remove directory: %s\n", argv[i]);
    }
}


return 0;
}

```

## Touch (-m) 명령어

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./touch aqw  
Created file: aqw
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./touch -m a.txt  
Modified file: a.txt  
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ll  
total 276  
drwxr-xr-x  3 shingiyoun shingiyoun 4096 Jun 13 19:32 ./  
drwxr-x--- 15 shingiyoun shingiyoun 4096 Jun 13 19:32 ../  
-rw-r--r--  1 shingiyoun shingiyoun   0 Jun 13 19:35 a.txt  
l  2 shingiyoun shingiyoun 4096 Jun 13 19:50 a.txt
```

A Linux desktop taskbar at the bottom of the terminal window. It contains several application icons: a web browser, a file manager, a terminal, and a system monitor. On the right side of the taskbar, there is a clock showing '오후 7:35' (7:35 PM) and a battery status icon.

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/stat.h>
#include <utime.h>
#include <ctype.h>

int main(int argc, char *argv[]) {
    int c;
    int modify = 0;

    while ((c = getopt(argc, argv, "m")) != -1) {
        switch (c) {
            case 'm':
                modify = 1;
                break;
            case '?':
                if (optopt == 'm')
                    fprintf(stderr, "Option -%c requires no argument.\n", optopt);
                else
                    fprintf(stderr, "Unknown option `-%c'.\n", optopt);
                return 1;
            default:
                abort();
        }
    }
}

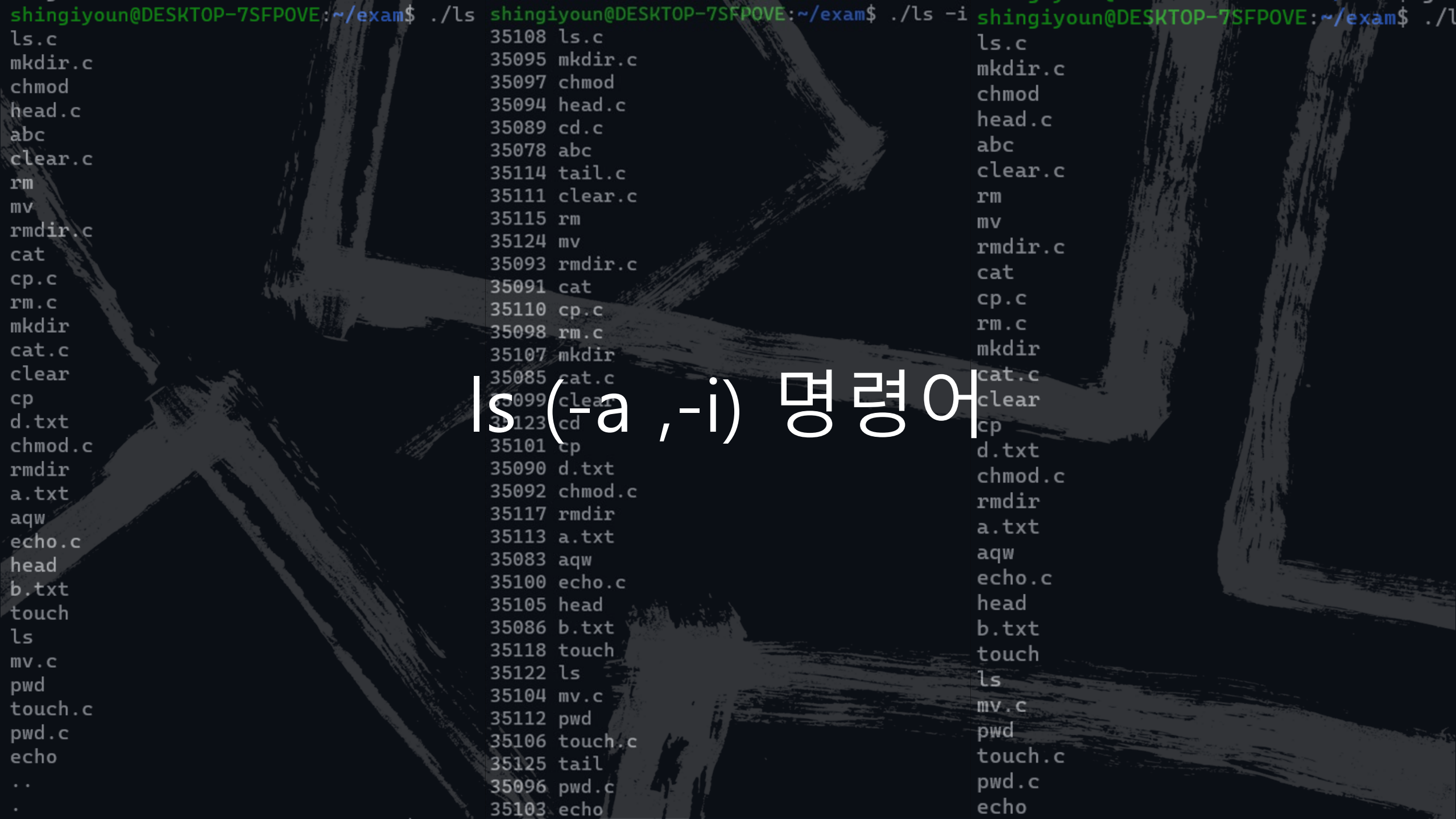
```

```

for (int i = optind; i < argc; ++i) {
    struct stat st;
    if (stat(argv[i], &st) == 0) {
        // 파일이 이미 존재하는 경우
        if (modify) {
            // -m 옵션이 지정된 경우 파일의 수정 시간을 현재 시간으로 변경합니다.
            struct utimbuf ut;
            ut.actime = st.st_atime;
            ut.modtime = time(NULL);
            if (utime(argv[i], &ut) == 0)
                printf("Modified file: %s\n", argv[i]);
            else
                printf("Failed to modify file: %s\n", argv[i]);
        }
    } else {
        // 파일이 존재하지 않는 경우
        FILE *file = fopen(argv[i], "w");
        if (file != NULL) {
            // 파일을 생성합니다.
            fclose(file);
            printf("Created file: %s\n", argv[i]);
        } else {
            printf("Failed to create file: %s\n", argv[i]);
        }
    }
}

return 0;
}

```



shingiyoun@DESKTOP-7SFPOVE:~/exam\$ ./ls shingiyoun@DESKTOP-7SFPOVE:~/exam\$ ./ls -i shingiyoun@DESKTOP-7SFPOVE:~/exam\$ ./ls

ls.c	35108 ls.c	ls.c
mkdir.c	35095 mkdir.c	mkdir.c
chmod	35097 chmod	chmod
head.c	35094 head.c	head.c
abc	35089 cd.c	abc
clear.c	35078 abc	clear.c
rm	35114 tail.c	rm
mv	35111 clear.c	mv
rmdir.c	35115 rm	rmdir.c
cat	35124 mv	cat
cp.c	35093 rmdir.c	cp.c
rm.c	35091 cat	rm.c
mkdir	35110 cp.c	mkdir
cat.c	35098 rm.c	cat.c
clear	35107 mkdir	clear
cp	35085 cat.c	cp
d.txt	35099 clear	d.txt
chmod.c	35123 cd	chmod.c
rmdir	35101 cp	rmdir
a.txt	35090 d.txt	a.txt
aqw	35092 chmod.c	aqw
echo.c	35117 rmdir	echo.c
head	35113 a.txt	head
b.txt	35083 aqw	b.txt
touch	35100 echo.c	touch
ls	35105 head	ls
mv.c	35086 b.txt	mv.c
pwd	35118 touch	pwd
touch.c	35122 ls	touch.c
pwd.c	35104 mv.c	pwd.c
echo	35112 pwd	echo
..	35106 touch.c	..
.	35125 tail	.
	35096 pwd.c	
	35103 echo	

# ls (-a , -i) 명령어



```

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

void list_files(int show_hidden, int show_inode) {
    DIR *dir;
    struct dirent *entry;
    struct stat file_stat;

    dir = opendir(".");
    if (dir == NULL) {
        perror("opendir");
        exit(EXIT_FAILURE);
    }

    while ((entry = readdir(dir)) != NULL) {
        if (!show_hidden && entry->d_name[0] == '.') {
            continue; // Skip hidden files
        }

        if (show_inode) {
            if (stat(entry->d_name, &file_stat) == -1) {
                perror("stat");
                exit(EXIT_FAILURE);
            }
            printf("%ld ", file_stat.st_ino);
        }

        printf("%s\n", entry->d_name);
    }
}

```

```

        closedir(dir);
    }

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

        while ((opt = getopt(argc, argv, "ai")) != -1) {
            switch (opt) {
                case 'a':
                    show_hidden = 1;
                    break;
                case 'i':
                    show_inode = 1;
                    break;
                default:
                    fprintf(stderr, "Usage: %s [-a] [-i]\n", argv[0]);
                    exit(EXIT_FAILURE);
            }
        }

        list_files(show_hidden, show_inode);

        return 0;
    }
}

```

# Tail(-n) 명령어

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./tail a.txt  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

```
shingiyoun@DESKTOP-7SFPOVE:~/exam$ ./tail -n 1 a.txt  
10
```

```

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

#define DEFAULT_N_LINES 10

void tail(FILE *file, int n_lines) {
    char **lines = (char **)malloc(n_lines * sizeof(char *));
    if (lines == NULL) {
        perror("malloc");
        exit(EXIT_FAILURE);
    }

    int line_count = 0;
    int curr_line = 0;

    char *buffer = NULL;
    size_t buffer_size = 0;
    ssize_t bytes_read;

    while ((bytes_read = getline(&buffer, &buffer_size, file)) != -1) {
        if (lines[curr_line] != NULL) {
            free(lines[curr_line]);
        }

        lines[curr_line] = buffer;
        buffer = NULL;
        buffer_size = 0;
        curr_line = (curr_line + 1) % n_lines;
        line_count++;
    }
}

```

```

if (bytes_read == -1 && ferror(file)) {
    perror("getline");
    exit(EXIT_FAILURE);
}

int start_line = line_count > n_lines ? (curr_line + 1) % n_lines : 0;
int count = line_count > n_lines ? n_lines : line_count;

for (int i = 0; i < count; i++) {
    int line_index = (start_line + i) % n_lines;
    printf("%s", lines[line_index]);
}

for (int i = 0; i < n_lines; i++) {
    free(lines[i]);
}
free(lines);

int main(int argc, char *argv[]) {
    int opt;
    int n_lines = DEFAULT_N_LINES;

    while ((opt = getopt(argc, argv, "n:")) != -1) {
        switch (opt) {
            case 'n':
                n_lines = atoi(optarg);
                if (n_lines <= 0) {
                    fprintf(stderr, "Invalid number of lines: %s\n", optarg);
                    exit(EXIT_FAILURE);
                }
                break;
            default:
                fprintf(stderr, "Usage: %s [-n num_lines] [file]\n", argv[0]);
                exit(EXIT_FAILURE);
        }
    }

    FILE *file;
    if (optind < argc) {
        file = fopen(argv[optind], "r");
        if (file == NULL) {

```

```

                return 0;
            }
        }
    }

    if (file == NULL) {
        perror("fopen");
        exit(EXIT_FAILURE);
    }

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

    tail(file, n_lines);
    if (file != stdin) {
        fclose(file);
    }

    return 0;
}

```

# More (-n) 명령어

```
shingiyoun@DESKTOP-7SFP0VE:~/exam$ ./more -n 3 a.txt
```

1

2

3

```

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

#define MAX_LINE_LENGTH 1024

void display_file(const char *filename, int num_lines) {
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        perror("fopen");
        exit(EXIT_FAILURE);
    }

    char line[MAX_LINE_LENGTH];
    int line_count = 0;

    while (fgets(line, MAX_LINE_LENGTH, file) != NULL) {
        fputs(line, stdout);
        line_count++;

        if (line_count == num_lines) {
            int c = getchar();
            if (c == 'q') {
                break;
            }
            line_count = 0;
        }
    }

    fclose(file);
}

```

```

    fclose(file);
}

int main(int argc, char *argv[]) {
    int opt;
    int num_lines = 10; // Default number of lines per page

    while ((opt = getopt(argc, argv, "n:")) != -1) {
        switch (opt) {
            case 'n':
                num_lines = atoi(optarg);
                break;
            default:
                fprintf(stderr, "Usage: %s [-n NUM_LINES] [FILE]\n", argv[0]);
                exit(EXIT_FAILURE);
        }
    }

    // Process the remaining arguments after options
    if (optind < argc) {
        for (int i = optind; i < argc; i++) {
            display_file(argv[i], num_lines);
        }
    } else {
        display_file("stdin", num_lines);
    }

    return 0;
}

```

# 출처

- <https://chat.openai.com/>