

System programming mid-exam source code

2022.04.17

2010 mid-exam

1. double linked list - 시험 범위 아님

2. main 함수가 실행될 때 argv[2]의 파일을 argv[1] 파일로 복사하는 프로그램을 작성

```
jeekim@jeekim-System-Product-Name: ~  
#include <stdio.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <fcntl.h>  
#include <stdlib.h>  
#include <unistd.h>  
#define LEN 64  
  
int main(int argc, char** argv) {  
    int fd1, fd2, size=0;  
    char buf[LEN];  
  
    if (argc < 3) {  
        fprintf(stderr, "usage: %s dst src\n", argv[0]);  
        exit(-1);  
    }  
  
    if( (fd1 = open(argv[1], O_WRONLY)) > 0 ) {  
        fprintf(stderr, "%s: exist\n", argv[1]);  
        exit(-2);  
    }  
    else {  
        if( (fd1 = open(argv[1], O_WRONLY|O_CREAT, 0666)) < 0 ) {  
            perror(argv[1]);  
            exit(-3);  
        }  
    }  
  
    if( (fd2 = open(argv[2], O_RDONLY)) < 0 ) {  
        perror(argv[2]);  
        close(fd1);  
        exit(-3);  
    }  
  
    while( (size = read(fd2, buf, sizeof(buf))) > 0 ) {  
        write(fd1, buf, size);  
    }  
    close(fd1);  
    close(fd2);  
    exit(0);  
}
```

2011 mid-exam

1. standard input 으로부터 1줄 씩 입력 받아, 입력 받은 내용을 argv[1] 파일에 저장하는 프로그램 작성

```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <fcntl.h>  
#include <unistd.h>  
#define LEN 256  
  
int main(int argc, char **argv) {  
    int fd, size=0;  
    char buf[LEN];  
  
    if (argc < 2) {  
        fprintf(stderr, "Usage: %s filename.\n", argv[0]);  
        exit(-1);  
    }  
  
    if ( (fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0644)) < 0 ) {  
        perror(argv[1]);  
        exit(-2);  
    }  
  
    while(1) {  
        if ( (size = read(0, buf, sizeof(buf))) < 0 )  
            break;  
  
        if (buf[0] == '.')  
            break;  
  
        write(fd, buf, size);  
    }  
  
    close(fd);  
    exit(0);  
}
```

2. argv[1] 파일에 있는 각 바이트의 값에 3을 더해서 argv[2] 파일에 저장하는 프로그램 작성

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#define LEN 128

int main(int argc, char** argv) {
    int fd1, fd2, n, size=0;
    char buf[LEN], tmp[1];

    if (argc < 4) {
        fprintf(stderr, "usage: %s inputfile n outfile\n", argv[0]);
        exit(-1);
    }

    if ((fd1 = open(argv[1], O_RDONLY)) < 0 ) {
        perror(argv[1]);
        exit(-1);
    }

    if ((fd2 = open(argv[3], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0 ) {
        perror(argv[3]);
        exit(-2);
    }

    n = atoi(argv[2]);
    if (sizeof(n) == 0) {
        printf("error");
        perror(argv[2]);
        exit(-3);
    }

    while((size = read(fd1, buf, 1)) > 0) {
        if (buf[0] != '\n')
            buf[0] = (int)buf[0] + n;
        write(fd2, buf, sizeof(tmp));
    }

    close(fd1);
    close(fd2);
    exit(0);
}
~
```

3. child 가 argv[1] 파일을 실행하도록 한다.

```

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

int main(int argc, char** argv) {
    int i=0, pid;
    char** argv_c = NULL;

    argv_c = (char**)malloc(sizeof(char*) * argc);

    if (argc < 2) {
        fprintf(stderr, "usage: %s programFile argv...\n", argv[0]);
        exit(-1);
    }

    pid = fork();

    if (pid == 0) {
        for(i=1; i<argc; i++) {
            argv_c[i-1] = argv[i];
        }
        argv_c[i-1] = (char*) NULL;
        execv(argv[1], argv_c);
        exit(0);
    }
    else{
        wait((int*)0);
    }
    perror("execv(): ");
    exit(0);
}
~
```

4. 문제와 같은 프로그램 작성

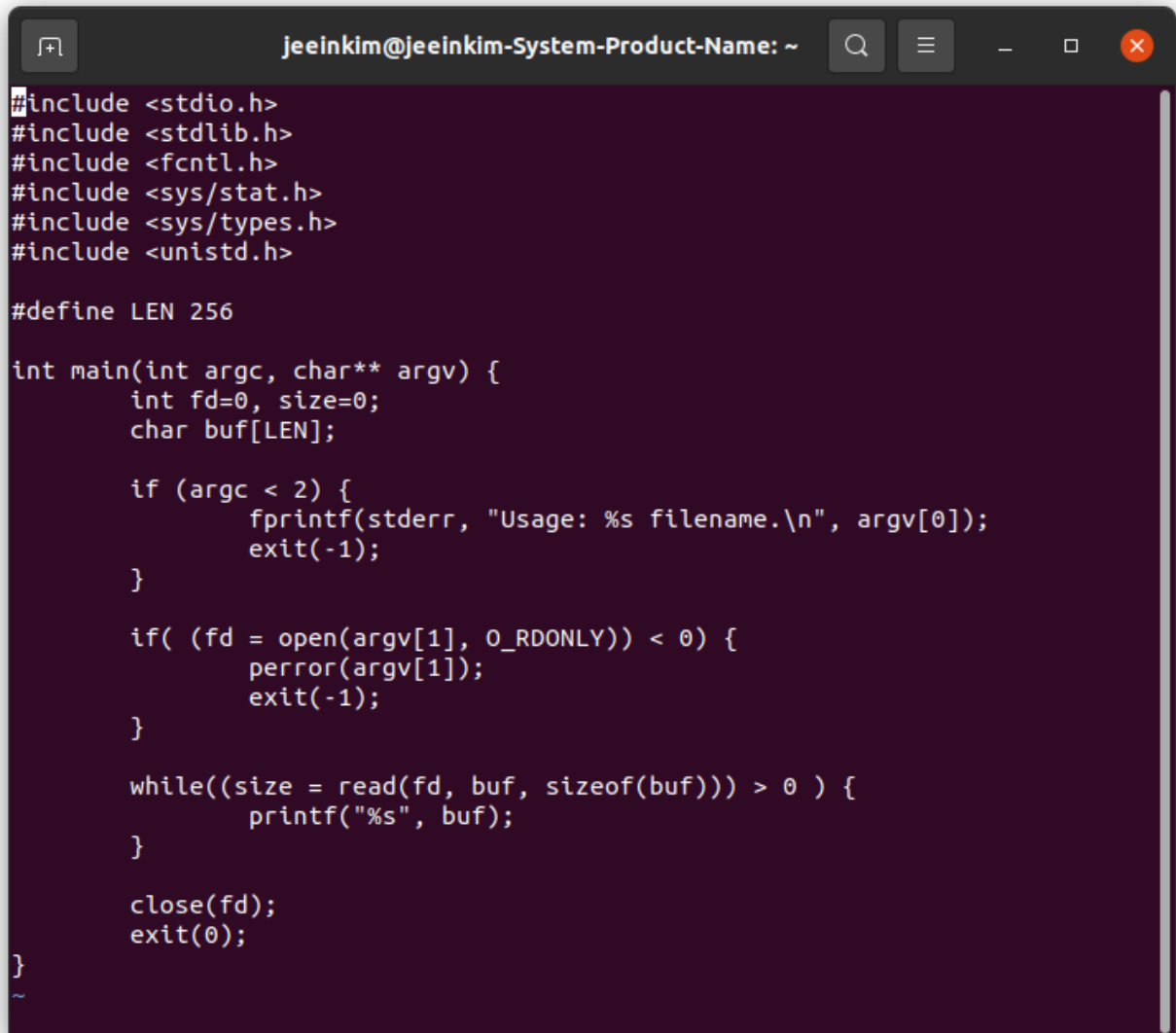
```

jееinkim@jeeinkim-System-Product-Name: ~
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/wait.h>

int main() {
    int pid, cid, childRet=0;
    pid = fork();

    if (pid == 0) {
        printf("Hello, World\n");
        exit(10);
    }
    else {
        cid = wait(&childRet);
        printf("child pid - [%d]\n", cid);
        childRet = childRet>>8;
        printf("exit code - [%d]\n", childRet);
    }
    exit(0);
}
~
```

1. argv[1]의 파일의 내용을 화면에 출력하는 프로그램을 작성

A terminal window with a dark purple background and white text. The window title is "jeeinkim@jeeinkim-System-Product-Name: ~". The code is a C program that reads a file specified by argv[1] and prints its contents. The code includes standard headers, defines a buffer size of 256, and uses open, read, and printf functions. The window has standard Linux terminal window controls (search, menu, zoom, close) in the top right corner.

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

#define LEN 256

int main(int argc, char** argv) {
    int fd=0, size=0;
    char buf[LEN];

    if (argc < 2) {
        fprintf(stderr, "Usage: %s filename.\n", argv[0]);
        exit(-1);
    }

    if( (fd = open(argv[1], O_RDONLY)) < 0) {
        perror(argv[1]);
        exit(-1);
    }

    while((size = read(fd, buf, sizeof(buf))) > 0 ) {
        printf("%s", buf);
    }

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

2. 프로그램에서 printf 로 출력하면 그 내용이 argv[1] 파일에 출력되도록 하는 프로그램

```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
  
#define LEN 512  
  
int main(int argc, char **argv) {  
    int count=0, fd1=0;  
    char buf[LEN];  
  
    if (argc < 2) {  
        fprintf(stderr, "Usage: %s filename.\n", argv[0]);  
        exit(-1);  
    }  
  
    if ((fd1 = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0644)) < 0) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    dup2(fd1, 1);  
    close(fd1);  
  
    while((count=read(0, buf, 512)) > 0 ) {  
        buf[count] = 0;  
        printf("%s", buf);  
    }  
  
    exit(0);  
}
```

```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
  
#define LEN 512  
  
int main(int argc, char **argv) {  
    int count=0, fd1=0;  
    char buf[LEN];  
  
    if (argc < 2) {  
        fprintf(stderr, "Usage: %s filename.\n", argv[0]);  
        exit(-1);  
    }  
  
    close(1);  
    if ((fd1 = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0644)) < 0) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    fd1 = dup(0);  
  
    while((count=read(0, buf, 512)) > 0 ) {  
        buf[count] = 0;  
        printf("%s", buf);  
    }  
  
    exit(0);  
}
```


3. 자식 프로세스 생성하여 자식 프로세스에게 “/bin/cp” 프로그램을 실행하여 파일을 복사하도록 하는 프로그램 작성

```

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

int main(int argc, char** argv) {
    int fd1, fd2, pid;

    if (argc < 3) {
        fprintf(stderr, "usage: a.out infile outfile\n");
        exit(-1);
    }

    if ((fd1 = open(argv[1], O_RDONLY)) < 0 ) {
        perror(argv[1]);
        exit(-1);
    }

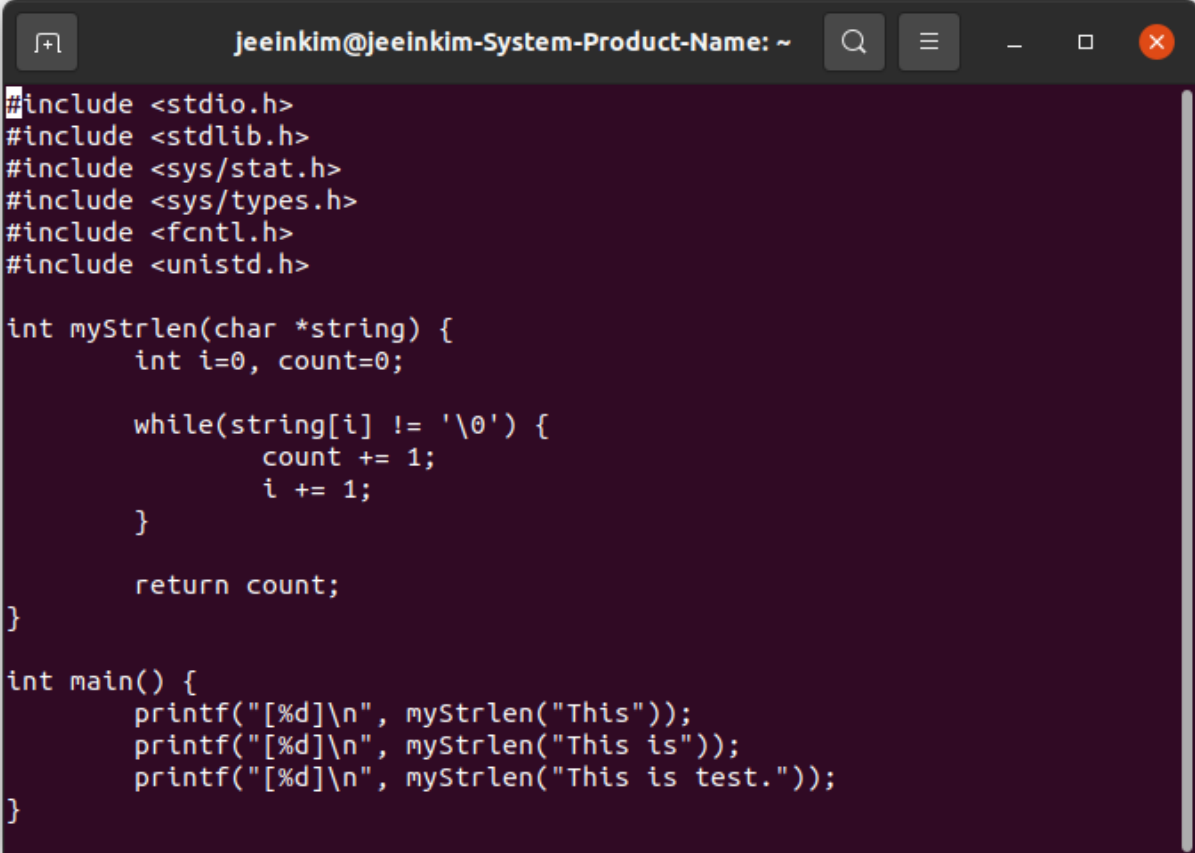
    if ((fd2 = open(argv[2], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0 ) {
        perror(argv[2]);
        close(fd1);
        exit(-1);
    }

    if ((pid = fork()) == -1) {
        perror("fork() : ");
        exit(-1);
    }

    if (pid == 0) {
        execl("/bin/cp", "interest", argv[1], argv[2], (char*) NULL);
        exit(0);
    }
    else {
        wait((int*)0);
        printf("child exited\n");
    }

    close(fd1);
    close(fd2);
    exit(0);
}
```

1. 문자열의 길이를 리턴하는 myStrlen(char *string) 함수를 완성

A terminal window with a dark purple background and light green text. The window title is "jeeinkim@jeeinkim-System-Product-Name: ~". The code is a C program that implements a custom strlen function named myStrlen. It includes standard headers and uses a while loop to count characters until a null terminator is reached. The main function tests the myStrlen function with three strings: "This", "This is", and "This is test.".

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

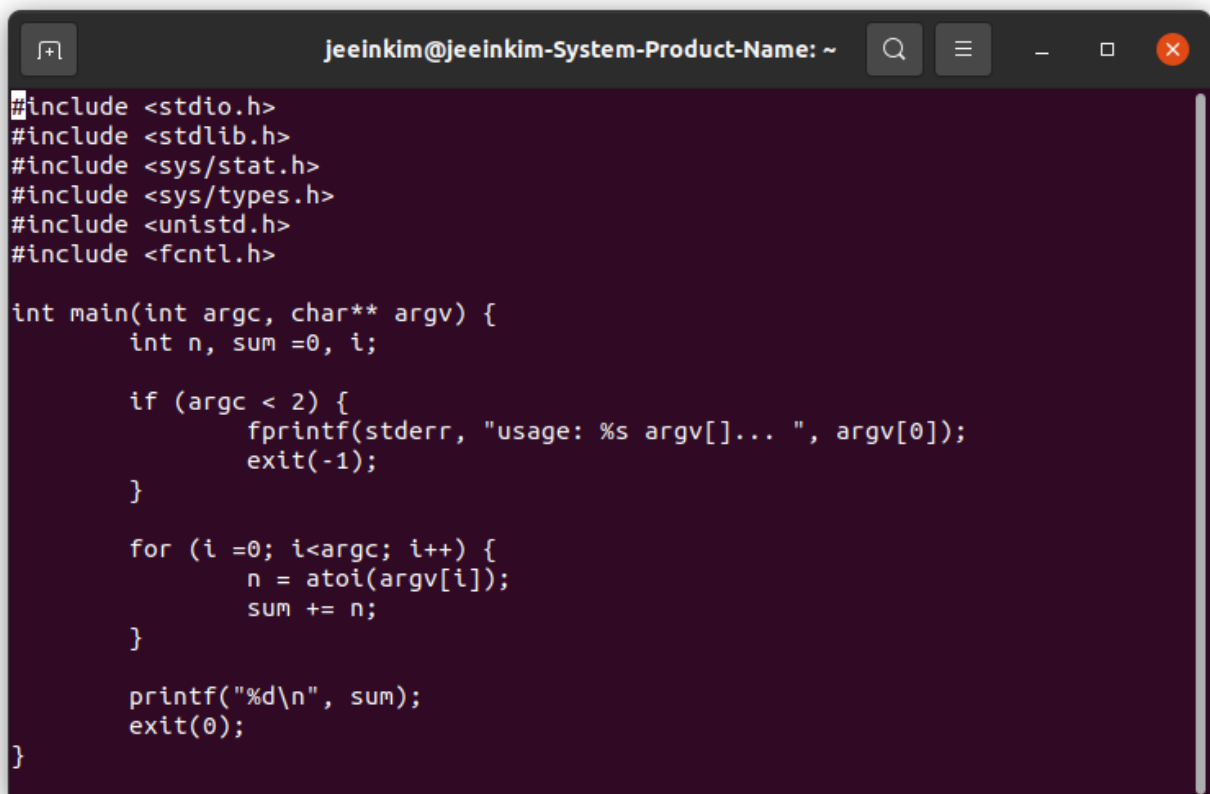
int myStrlen(char *string) {
    int i=0, count=0;

    while(string[i] != '\0') {
        count += 1;
        i += 1;
    }

    return count;
}

int main() {
    printf("[%d]\n", myStrlen("This"));
    printf("[%d]\n", myStrlen("This is"));
    printf("[%d]\n", myStrlen("This is test."));
}
```

2. argv[1] + argv[2] + ... + argv[n]의 값을 출력하는 프로그램 작성



```
jeekimkim@jeekimkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
  
int main(int argc, char** argv) {  
    int n, sum = 0, i;  
  
    if (argc < 2) {  
        fprintf(stderr, "usage: %s argv[]... ", argv[0]);  
        exit(-1);  
    }  
  
    for (i = 0; i < argc; i++) {  
        n = atoi(argv[i]);  
        sum += n;  
    }  
  
    printf("%d\n", sum);  
    exit(0);  
}
```

3. argv[1]의 파일에서 argv[2]의 문자열이 있는가를 검사하여, 해당 파일에서 문자가 발견되면 “Found” 라고 출력하고, 발견되지 않으면 “Not Found”라고 출력하는 프로그램을 작성

```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <fcntl.h>  
#include <unistd.h>  
#include <string.h>  
#define LEN 128  
  
int main(int argc, char** argv) {  
    int fd, size, i=0, flag=0;  
    char buf[LEN], word[LEN];  
  
    if (argc < 3) {  
        fprintf(stderr, "usage: %s filename word\n", argv[0]);  
        exit(-1);  
    }  
  
    if ((fd = open(argv[1], O_RDONLY)) < 0) {  
        perror(argv[1]);  
        exit(-2);  
    }  
  
    while((size = read(fd, buf, 1)) > 0) {  
        if(buf[0] == '\n' || buf[0] == '\t' || buf[0] == ' ') {  
            if(strcmp(argv[2], word) == 0) {  
                printf("Found\n");  
                flag = 1;  
                break;  
            }  
            memset(buf, 0x00, sizeof(buf));  
            memset(word, 0x00, sizeof(word));  
            i=0;  
        }  
        else{  
            word[i] = buf[0];  
            i += 1;  
        }  
    }  
  
    if (flag==0) {  
        printf("Not Found\n");  
    }  
  
    close(fd);  
    exit(0);  
}
```

4. main 프로그램에서 scanf 함수로 읽으면, argv[1] 파일의 내용이 읽혀질 수 있도록 하는 프로그램을 작성 (dup 사용은 선택)

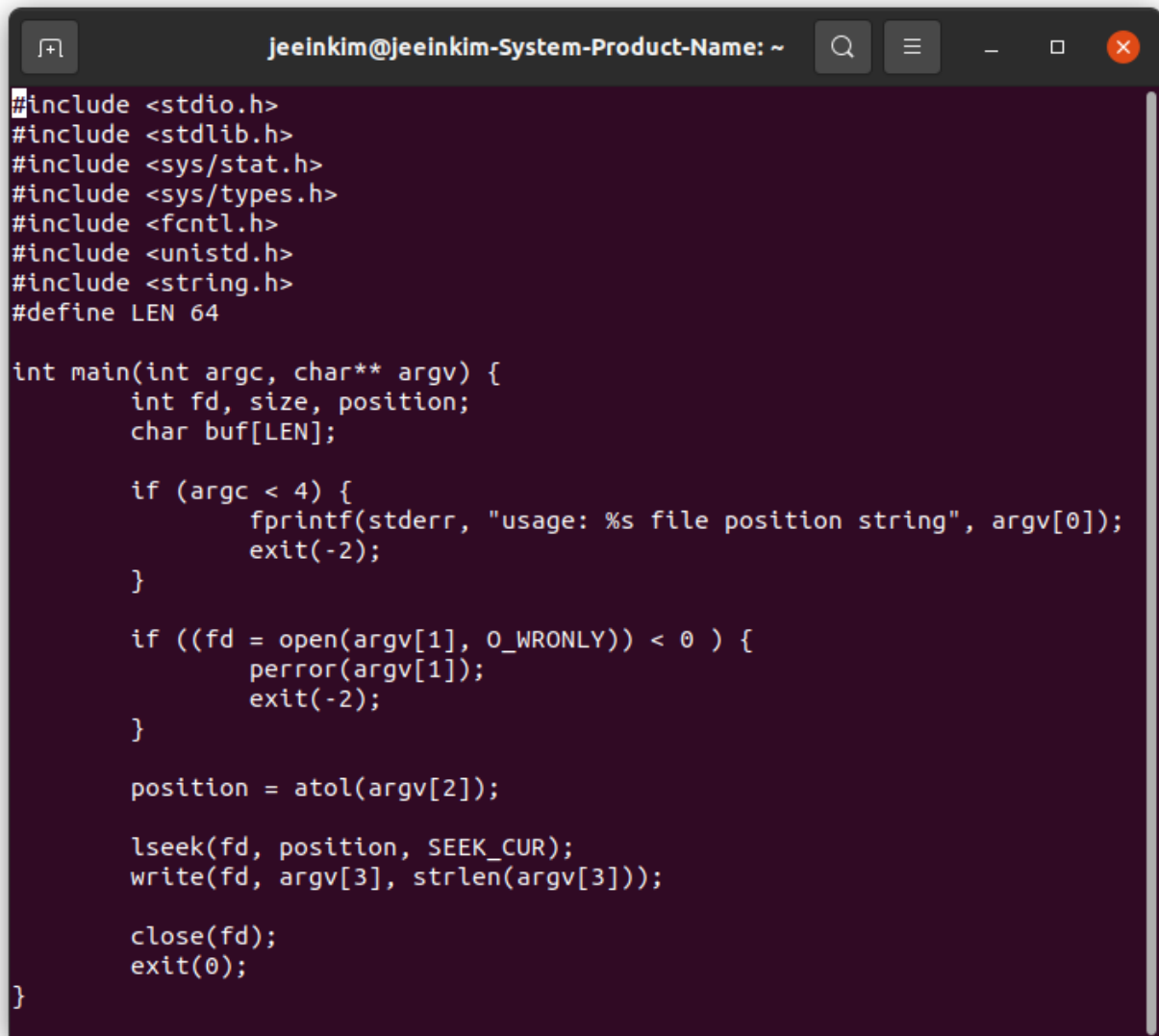
```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <fcntl.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int main(int argc, char** argv) {  
    int fd;  
    char buf[1];  
  
    if (argc < 2) {  
        fprintf(stderr, "usage: %s filename\n", argv[0]);  
        exit(-1);  
    }  
  
    close(0);  
    if ((fd = open(argv[1], O_RDONLY) ) < 0) {  
        perror(argv[1]);  
        exit(-2);  
    }  
  
    while(scanf("%c", buf) > 0 ) {  
        printf("%c", buf[0]);  
    }  
  
    close(fd);  
    exit(0);  
}
```

2014 mid-exam

1. 어떤 사람의 이름과 나이를 struct 구조에 읽어서 argv[1] 파일에 저장하는 프로그램을 작성

```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
#include <string.h>  
  
typedef struct {  
    char name[20];  
    int age;  
}PERSON;  
  
int main(int argc, char** argv) {  
    int fd, size=0;  
    PERSON p;  
  
    if (argc < 2) {  
        fprintf(stderr, "usage: %s filename\n", argv[0]);  
        exit(-1);  
    }  
  
    if( (fd = open(argv[1], O_WRONLY|O_CREAT|O_APPEND, 0600)) < 0 ) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    while(1) {  
        memset(&p, 0x00, sizeof(PERSON));  
        fprintf(stderr, "Insert Person name: ");  
        fflush(stderr);  
  
        if( (size = read(0, p.name, sizeof(p.name))) < 0 )  
            break;  
  
        if (p.name[0] == '-')  
            break;  
  
        fprintf(stderr, "Insert Person age: ");  
        fflush(stderr);  
  
        if( (size = read(0, &p.age, sizeof(p.age))) < 0 )  
            break;  
  
        write(fd, &p, sizeof(PERSON));  
    }  
  
    close(fd);  
    exit(0);  
}
```

2. 사용자가 shell 에서 “% a.out file_name position string” 이라고 입력하면 기존에 존재하는 “file_name”의 파일 처음부터 “position” 바이트만큼 떨어진 위치에 “string”을 쓰는 프로그램 작성

A terminal window with a dark purple background and white text. The window title is "jeeinkim@jeeinkim-System-Product-Name: ~". The code is a C program that takes four arguments: file name, position, and string. It opens the file in write-only mode, seeks to the specified position, and writes the string. The code is as follows:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#define LEN 64

int main(int argc, char** argv) {
    int fd, size, position;
    char buf[LEN];

    if (argc < 4) {
        fprintf(stderr, "usage: %s file position string", argv[0]);
        exit(-2);
    }

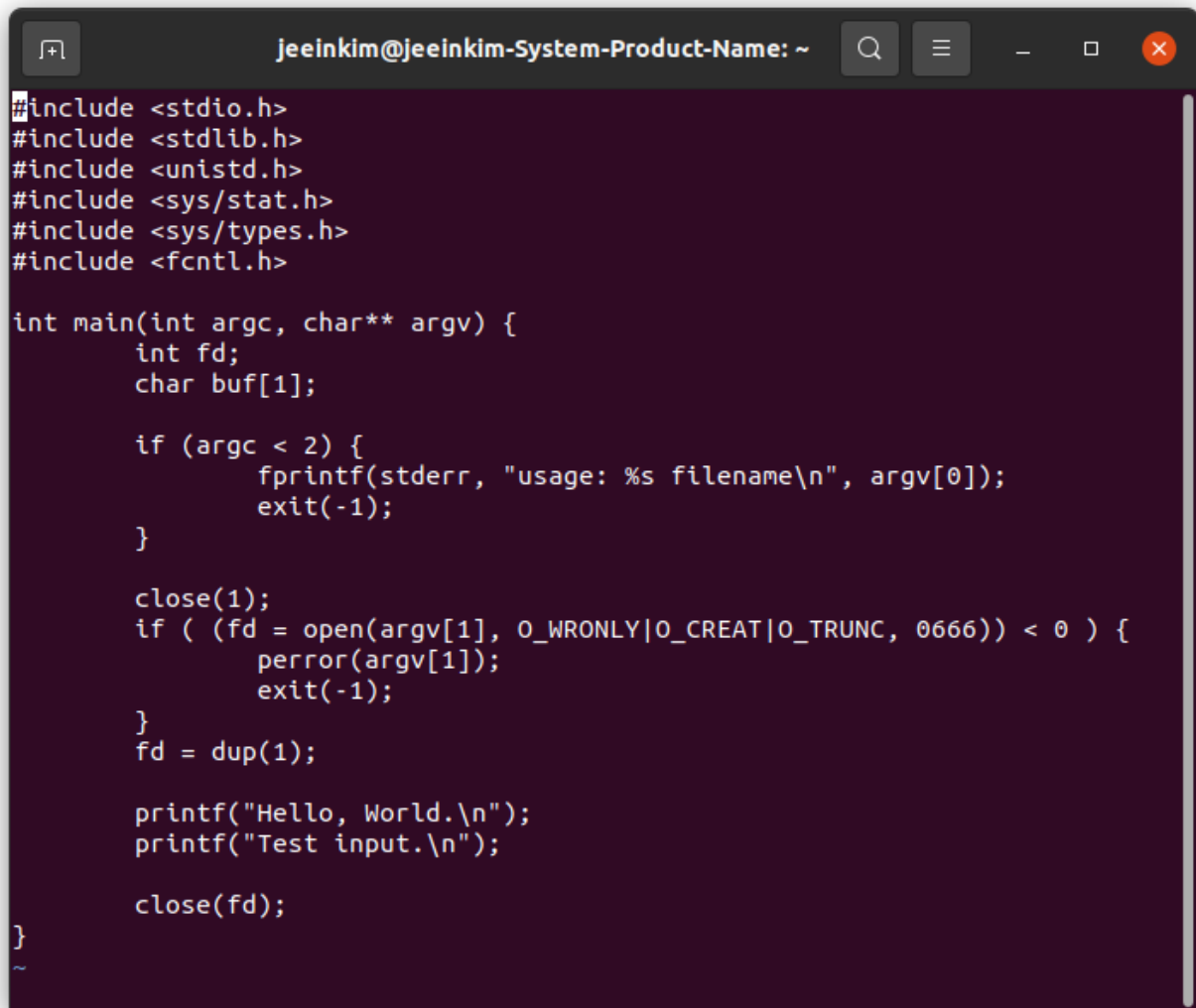
    if ((fd = open(argv[1], O_WRONLY)) < 0 ) {
        perror(argv[1]);
        exit(-2);
    }

    position = atol(argv[2]);

    lseek(fd, position, SEEK_CUR);
    write(fd, argv[3], strlen(argv[3]));

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

3. main 프로그램에서 printf 함수로 출력하면 , argv[1] 파일로 출력되도록 하는 프로그램을 작성. (dup 호출을 사용할 수도 있고, 그렇지 않을 수도 있다.)

A terminal window with a dark purple background and white text. The window title is 'jeeinkim@jeeinkim-System-Product-Name: ~'. The code is a C program that takes a filename as an argument and prints to that file using dup. The code is as follows:

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

int main(int argc, char** argv) {
    int fd;
    char buf[1];

    if (argc < 2) {
        fprintf(stderr, "usage: %s filename\n", argv[0]);
        exit(-1);
    }

    close(1);
    if ( (fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0 ) {
        perror(argv[1]);
        exit(-1);
    }
    fd = dup(1);

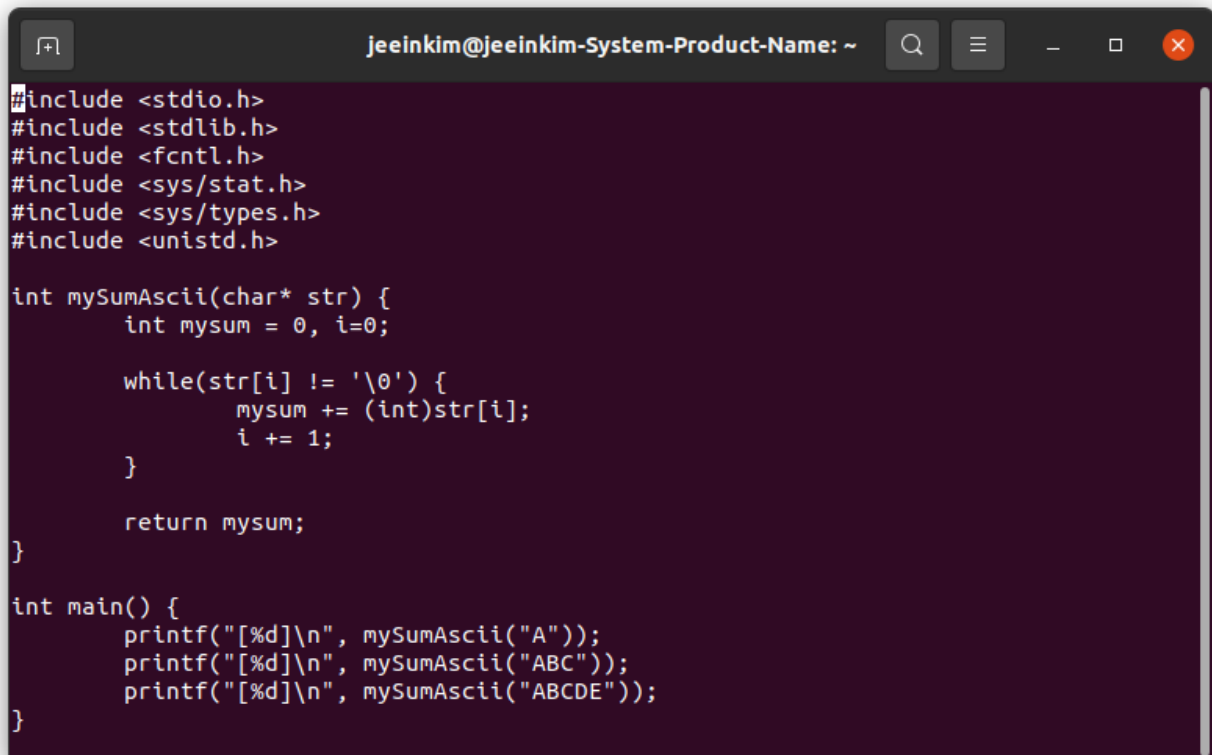
    printf("Hello, World.\n");
    printf("Test input.\n");

    close(fd);
}
```


4. argv[1]의 파일에서 argv[2] 문자열이 있는가를 검사하여, 해당 파일에서 문자가 발견되면 “Found”라고 출력하고, 발견되지 않으면 “Not Found”라고 출력하는 프로그램 작성 (command 로 인식하는 오류가 생긴다.)

2015 mid-exam

1. 문자열의 아스키코드 값의 합을 리턴해주는 mySumAscii(char *string) 함수 작성

A terminal window with a dark purple background and light green text. The window title is "jeeinkim@jeeinkim-System-Product-Name: ~". The code is as follows:

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

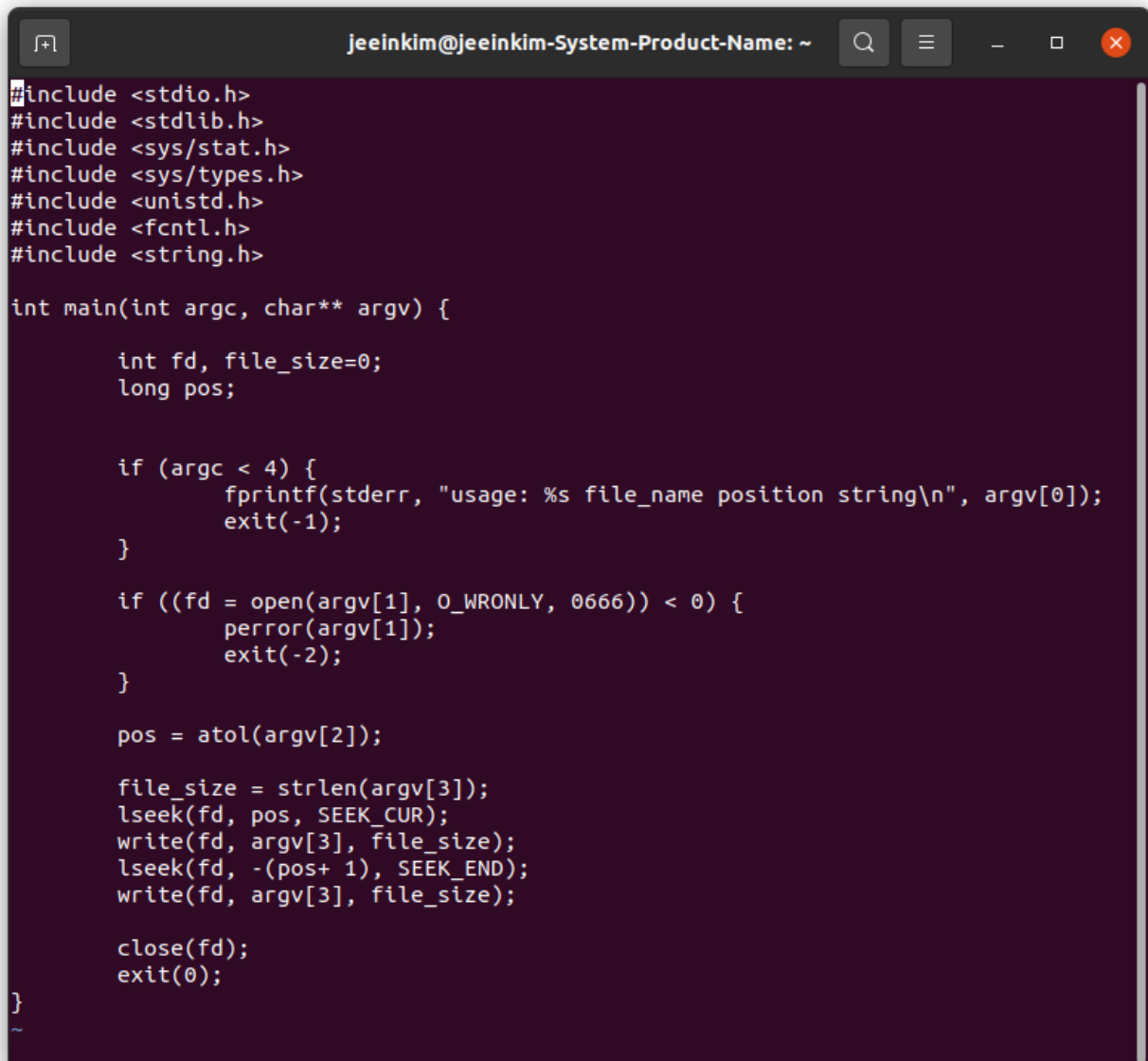
int mySumAscii(char* str) {
    int mysum = 0, i=0;

    while(str[i] != '\0') {
        mysum += (int)str[i];
        i += 1;
    }

    return mysum;
}

int main() {
    printf("[%d]\n", mySumAscii("A"));
    printf("[%d]\n", mySumAscii("ABC"));
    printf("[%d]\n", mySumAscii("ABCDE"));
}
```

2. 사용자가 shell 에서 “% ./a.out file_name position string” 이라고 입력하면 기존에 존재하는 “file_name” 파일의 처음부터 “position” 바이트만큼 떨어진 위치에 “string”을 입력하고, 파일의 끝에서부터 “position” 바이트만큼 떨어진 위치에 “string”을 쓰는 프로그램 작성

A terminal window with a dark purple background and light green text. The window title is "jeeinkim@jeeinkim-System-Product-Name: ~". The code is a C program that takes four arguments: file name, position, and string. It opens the file in write-only mode, seeks to the specified position, and writes the string. It also has a fallback to write at the end of the file if the position is not found. The code is as follows:

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

int main(int argc, char** argv) {
    int fd, file_size=0;
    long pos;

    if (argc < 4) {
        fprintf(stderr, "usage: %s file_name position string\n", argv[0]);
        exit(-1);
    }

    if ((fd = open(argv[1], O_WRONLY, 0666)) < 0) {
        perror(argv[1]);
        exit(-2);
    }

    pos = atol(argv[2]);

    file_size = strlen(argv[3]);
    lseek(fd, pos, SEEK_CUR);
    write(fd, argv[3], file_size);
    lseek(fd, -(pos+ 1), SEEK_END);
    write(fd, argv[3], file_size);

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

3. 자식 프로세스를 생성하여 부모 프로세스는 출력 값을 화면에 출력하고, 자식 프로세스는 출력 값을 argv[1]의 파일에 출력하는 프로그램을 작성하라.

```

jeeeinkim@jeeinkim-System-Product-Name: ~
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <sys/wait.h>
#define LEN 128

int main(int argc, char** argv) {
    int fd=0, fdd=0, ret;
    int stdout_fd = dup(1);
    char buf[LEN];

    if (argc < 2) {
        fprintf(stderr, "Usage: %s filename.\n", argv[0]);
        exit(-1);
    }

    close(1);
    if ((fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0) {
        perror(argv[1]);
        exit(-1);
    }
    ret = fork();

    if (ret != 0) {
        wait((int *)0);

        dup2(stdout_fd, 1);

        printf("[Parent]\n");
        printf("getpid() - %d\n", getpid());
        printf("getuid() - %d\n", getuid());
        printf("getgid() - %d\n", getgid());
    }
    else {
        fd = dup(0);
        printf("[Child]\n");
        printf("getppid - %d\n", getppid());
        printf("geteuid - %d\n", geteuid());
        printf("getegid - %d\n", getegid());
        exit(0);
    }

    close(fd);
    exit(0);
}

"mid2015_3.c" 48 lines, 888 characters
```

4. argv[1] 파일의 내용을 argv[2] 파일로 복사하는 프로그램을 작성 (dup 사용)

```
jeekimkim@jeekimkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
#define LEN 245  
  
int main(int argc, char** argv) {  
    int fd1, fd2, length=0, size=0;  
    char buf[LEN];  
  
    if (argc < 3) {  
        fprintf(stderr, "usage: %s srcfile dstfile\n", argv[0]);  
        exit(-1);  
    }  
  
    close(0);  
    if ((fd1 = open(argv[1], O_RDONLY)) < 0) {  
        perror(argv[1]);  
        exit(-2);  
    }  
  
    close(1);  
    if ((fd2 = open(argv[2], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0) {  
        perror(argv[2]);  
        exit(-3);  
    }  
  
    length = sizeof(buf);  
    while((size = read(0, buf, length)) > 0) {  
        write(1, buf, size);  
    }  
    close(fd1);  
    close(fd2);  
    exit(0);  
}
```

2016 mid-exam

1. 문자열이 몇 개나 있는지 검사하여, 발견된 문자열의 개수를 출력하는 프로그램을 작성

```
jeekimkim@jeekimkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
#include <string.h>  
#define LEN 256  
  
int main(int argc, char** argv) {  
    int fd1, size, count=0, i=0;  
    char word[LEN];  
    char buf[1];  
    char target[LEN];  
  
    if (argc < 3) {  
        fprintf(stderr, "Usage: %s filename\n", argv[0]);  
        exit(-1);  
    }  
  
    if ((fd1 = open(argv[1], O_RDONLY)) < 0) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    strcpy(target, argv[2]);  
    while((size = read(fd1, buf, 1)) > 0) {  
        if (buf[0] == '\n' || buf[0] == ' ' || buf[0] == '\t') {  
            if (strcmp(target, word) == 0) {  
                count += 1;  
            }  
            memset(buf, 0x00, sizeof(buf));  
            memset(word, 0x00, sizeof(word));  
            i = 0;  
        }  
        else {  
            word[i] = buf[0];  
            i += 1;  
        }  
    }  
  
    printf("count - [%d]\n", count);  
  
    close(fd1);  
    exit(0);  
}
```

~
"mid2016_1.c" 47 lines, 824 characters

2. 어떤 사람의 이름과 나이를 struct 구조에 읽어서 argv[1] 파일에 저장하는 프로그램 작성

```
jeekimkim@jeekimkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <fcntl.h>  
#include <unistd.h>  
#include <string.h>  
  
typedef struct  
{  
    char name[20];  
    int age;  
} Person;  
  
int main(int argc, char **argv) {  
    int fd=0, size=0;  
    Person p;  
  
    if(argc < 2) {  
        fprintf(stderr, "Usage: %s filename\n", argv[0]);  
        exit(-1);  
    }  
  
    if((fd = open(argv[1], O_WRONLY|O_CREAT|O_EXCL, 0640)) < 0 ) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    while(1) {  
        memset(&p, 0x00, sizeof(Person));  
        fprintf(stderr, "Inset Person Name: ");  
        fflush(stderr);  
  
        if((size=read(0, p.name, sizeof(p.name))) < 0 )  
            break;  
  
        if (p.name[0] == '*')  
            break;  
  
        fprintf(stderr, "Insert Person's age: ");  
        fflush(stderr);  
  
        if((size=read(0, &p.age, sizeof(p.age))) < 0 )  
            break;  
  
        write(fd, (char *)&p, sizeof(Person));  
    }  
    close(fd);  
    exit(0);  
}
```

3. main 프로그램에서 printf 로 출력하면 그 내용이 argv[1] 파일에 출력하는 프로그램 작성

```
jeekim@jeekim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
#define LEN 256  
  
int main(int argc, char** argv) {  
    int fd=0, count;  
    char buf[LEN];  
  
    if (argc < 2) {  
        fprintf(stderr, "Usage: %s filename\n", argv[0]);  
        exit(-1);  
    }  
  
    close(1);  
    if ((fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0 ) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    fd = dup(1);  
    while((count = read(0, buf, 512)) > 0 ) {  
        buf[count] = 0;  
        printf("%s", buf);  
    }  
  
    close(fd);  
}
```

1. argv[1] 파일의 처음부터 n 바이트 떨어진 위치부터 시작하는 m 개의 글자를 c 로 바꾸는 프로그램 작성

```
jeekimkim@jeekimkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <fcntl.h>  
#include <string.h>  
  
#define LEN 128  
  
int main(int argc, char** argv) {  
    int fd, n, m, size=0, i;  
    char buf[LEN], c[1];  
    int end_pos=0, start_pos =0;  
  
    if (argc < 5) {  
        fprintf(stderr, "Usage: %s fileName n m c\n", argv[0]);  
        exit(-1);  
    }  
  
    if ((fd = open(argv[1], O_WRONLY)) < 0 ) {  
        perror(argv[1]);  
        exit(-1);  
    }  
  
    n = atoi(argv[2]);  
    m = atoi(argv[3]);  
  
    end_pos = lseek(fd, 0L, SEEK_END);  
    start_pos = lseek(fd, 0L, SEEK_SET);  
  
    size = end_pos - start_pos - 1;  
  
    if( (n+m) > size) {  
        fprintf(stderr, "overflow\n");  
        exit(-2);  
    }  
  
    lseek(fd, n, SEEK_CUR);  
  
    for(i=0; i<m; i++)  
        write(fd, argv[4], 1);  
  
    close(fd);  
    exit(0);  
}
```


2. argv[1] 파일의 각 바이트에 argv[2] 값을 더한 결과를 argv[3] 파일에 저장하는 프로그램

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#define LEN 128

int main(int argc, char** argv) {
    int fd1, fd2, n, size=0;
    char buf[LEN], tmp[1];

    if (argc < 4) {
        fprintf(stderr, "usage: %s inputfile n outfile\n", argv[0]);
        exit(-1);
    }

    if ((fd1 = open(argv[1], O_RDONLY)) < 0 ) {
        perror(argv[1]);
        exit(-1);
    }

    if ((fd2 = open(argv[3], O_WRONLY|O_CREAT|O_TRUNC, 0666)) < 0 ) {
        perror(argv[3]);
        exit(-2);
    }

    n = atoi(argv[2]);
    if (sizeof(n) == 0) {
        printf("error");
        perror(argv[2]);
        exit(-3);
    }

    while((size = read(fd1, buf, 1)) > 0) {
        if (buf[0] != '\n')
            buf[0] = (int)buf[0] + n;
        write(fd2, buf, sizeof(tmp));
    }

    close(fd1);
    close(fd2);
    exit(0);
}
~
```

3. 자식 프로세스가 “/usr/bin/cmp” 프로그램을 실행하여 파일을 비교하도록 하는 프로그램 작성

```
jeeinkim@jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <sys/stat.h>  
#include <sys/types.h>  
#include <fcntl.h>  
#include <sys/wait.h>  
  
int main(int argc, char** argv) {  
    int i=0, pid;  
    char** new_argv;  
  
    new_argv = calloc(argc, sizeof(char*));  
  
    if (argc < 2) {  
        fprintf(stderr, "usage: %s childname argv...\n", argv[0]);  
        exit(-1);  
    }  
  
    pid = fork();  
  
    if (pid == 0) {  
        for(i=1; i<argc; i++) {  
            new_argv[i-1] = argv[i];  
        }  
        new_argv[i-1] = (char*)NULL;  
        execv("/usr/bin/cmp", new_argv);  
        exit(0);  
    }  
    else {  
        wait((int *) 0);  
        printf("child exited.\n");  
    }  
  
    exit(0);  
}
```

2018 mid-exam

1. 물음에 답하기

(가) System call

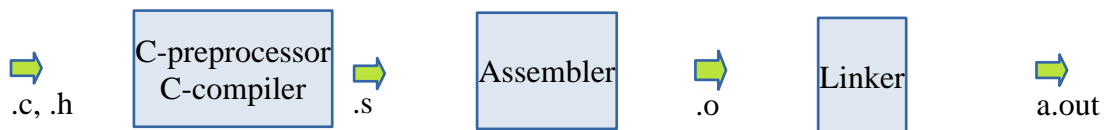
[특징]

- * 함수와 형태와 동작 방법이 비슷하다.
- * 정보를 제공하거나 특정 동작을 수행한다.
- * 에러발생 -1 리턴
- * 시스템 호출 코드는 운영체제 내에 존재한다.

[종류]

- * IO – read, write, sendto, recvfrom,...
- * File Management/Control – open, close, mknod,...
- * Access to System Data – getpid, getuid, gettimeofday,...
- * Process Management and Synchronization – fork, exec, signal, IPC
- * Storage Management – mmap, munmap (for malloc)

(나) C 컴파일 과정의 최종 단계인 linking 과정



- * Linkage
- * Adjust function calls

2. main 프로그램에서 scanf 함수로 읽으면 argv[1] 파일에서 읽혀지고, printf 함수로 출력하면 ,argv[2] 파일로 출력되도록 하는 프로그램 작성

```
Jeeinkim@Jeeinkim-System-Product-Name: ~  
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
  
int main(int argc, char *argv[]) {  
    char ch;  
    int fd1, fd2;  
  
    if (argc < 3) {  
        fprintf(stderr, "Usage: %s infile outfile.\n", argv[0]);  
        exit(-1);  
    }  
    close(0);  
    if ( (fd1 = open(argv[1], O_RDONLY)) < 0 ) {  
        perror(argv[1]);  
        exit(-2);  
    }  
    fd1 = dup(0);  
  
    close(1);  
    if ( (fd2 = open(argv[2], O_WRONLY|O_CREAT|O_EXCL, 0666)) < 0 ) {  
        perror(argv[2]);  
        close(fd1);  
        exit(-3);  
    }  
  
    fd2 = dup(1);  
    while(scanf("%c", &ch) > 0 ) {  
        printf("%c", ch);  
    }  
  
    close(fd1);  
    close(fd2);  
    exit(0);  
}
```

3. 문제와 같이 동작하는 프로그램 작성

```

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

int main(int argc, char** argv) {
    int pid, i;
    char **new_argv;

    new_argv = calloc(argc, sizeof(char*));

    if(argc < 2) {
        fprintf(stderr, "usage: %s program argv...\n", argv[0]);
        exit(-1);
    }

    if( (pid = fork()) == -1) {
        perror("fork() :");
        exit(-3);
    }

    if (pid == 0) {
        for (i =1; i<argc; i++)
            new_argv[i-1] = argv[i];
        new_argv[i-1] = (char*)NULL;

        execv(argv[1], new_argv);
        exit(0);
    }
    else {
        wait((int *)0);
        printf("parent exit\n");
    }

    exit(0);
}
~
```

참고 사항

- 2019, 2020, 2021년도 기출 문제는 제공 받지 않았음.
- 꼭 다 풀어보기를 추천함.
- man command 활용하면 좋다.