

응용과제 4

<소스코드>

```
#include<sys/types.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdlib.h>
#include<stdio.h>
#include<sys/errno.h>
#include<string.h>
#define BUF_SIZE 512

int main(int argc, char *argv[])
{
    int fd1,fd2, rd_cnt;
    int opt;
    char alast_opt;
    int check_argc;
    char answer;
    char buf[BUF_SIZE];
    extern char *optarg;
    extern int optind;
    char *file1,*file2;
    extern int opterr;
    opterr=1;

    //printf("argc %d optind %d",argc,optind);

    if(argc-optind<2)
    {
        fprintf(stderr,"파일2개가 선택되지 않았습니다.\n사용법 : %s -옵션 파일명1 파일명2\n-x : 복사
할 파일이 존재하는 경우 오류메세지 출력후 종료\n-t : 복사할 파일이 존재하는 경우 무조건 지우고 복사\n-y : 복
사할 파일이 존재하는 경우 사용자에게 메세지 출력, 사용자가 y를 답하는경우 지우고 복사. 그렇지 않는경우 종
료.\n\n옵션이 지정되지 않으면 옵션-x로 동작합니다\n여러옵션을 지정한경우 마지막 설정한 옵션으로 동작합니
다.\n",argv[0]);
        exit(1);
    }

    while((opt=getopt(argc,argv,"x:t:y:"))!=-1)
    {
        //if(argc-optind<3)
        //    opt='x';
        //printf("현재opt %c",opt);
        file1=argv[argc-optind+1]; //원본파일
        file2=argv[argc-optind+2]; //대상파일

        switch(opt)
        {
            case '?' :
                printf("x로 동작");
            case 'x' :
                if(argc-optind>1)
                //if(argv[optind]==file1)
                {
                    printf("마지막 옵션으로 진행합니다\n");
                    break;
                }
                else
                {
                    fd1=open(file1, O_RDONLY);
                    if(fd1<0)
                    {
                        perror("여는데 오류 발생\n");
                    }
                }
            }
```

```

        if(errno==ENOENT)
            fprintf(stderr,"복사할 소스파일(%s)가 존재하지 않음.\n",file1);
        exit(2);
    }
    fd2=open(file2, O_CREAT | O_RDWR | O_EXCL , 0644);
    if(fd2<0)
    {
        perror("여는데 오류 발생\n");
        if(errno==EEXIST)
            fprintf(stderr,"복사할 대상파일(%s)이 이미 존재함.\n",file2);
        exit(2);
    }
    while((rd_cnt=read(fd1,buf,BUF_SIZE))>0)
    {
        if(write(fd2,buf,rd_cnt)!=rd_cnt)
        {
            perror("파일을 쓰는중 오류 발생.\n");
            exit(3);
        }
    }
    if(rd_cnt<0)
    {
        perror("파일을 읽는중 오류 발생.\n");
        exit(4);
    }
    break;
}
case 't' :
    if(argc-optind>1)
        //if(argv[optind]==file1)
        {
            printf("마지막옵션으로 진행합니다\n");
            break;
        }
    else
    {
        fd1=open(file1, O_RDONLY);
        if(fd1<0)
        {
            printf("%s\n",file1);
            perror("여는데 오류 발생\n");
            if(errno==ENOENT)
                fprintf(stderr,"복사할 소스파일(%s)가 존재하지 않음.\n",file1);
            exit(2);
        }
        fd2=open(file2, O_CREAT | O_RDWR | O_TRUNC , 0644);
        if(fd2<0)
        {
            perror("여는데 오류 발생\n");
            exit(2);
        }
        while((rd_cnt=read(fd1,buf,BUF_SIZE))>0)
        {
            if(write(fd2,buf,rd_cnt)!=rd_cnt)
            {
                perror("파일을 쓰는중 오류 발생.\n");
                exit(3);
            }
        }
        if(rd_cnt<0)
        {
            perror("파일을 읽는중 오류 발생.\n");
            exit(4);
        }
    }
}

```

```

        }
        break;
    }
case 'y' :
    if(argc-optind>1)
        //if(argv[optind]==file1)
        {
            printf("마지막옵션으로 진행합니다\n");
            break;
        }
    else
    {
        fd1=open(file1, O_RDONLY);
        if(fd1<0)
        {
            perror("여는데 오류 발생\n");
            if(errno==ENOENT)
                fprintf(stderr,"복사할 소스파일(%s)가 존재하지 않음.\n",file1);
            exit(2);
        }
        fd2=open(file2, O_CREAT | O_RDWR | O_EXCL , 0644);
        if(fd2<0)
        {
            if(errno==EEXIST)
            {
                printf("대상파일(%s)가 이미 존재합니다. 지우고 복사할까요? : ",file2);
                scanf("%c",&answer);
                switch(answer)
                {
                    case'y' :
                        fd2=open(file2, O_CREAT | O_RDWR | O_TRUNC , 0644);
                        while((rd_cnt=read(fd1,buf,BUF_SIZE))>0)
                        {
                            if(write(fd2,buf,rd_cnt)!=rd_cnt)
                            {
                                perror("파일을 쓰는중 오류 발생.\n");
                                exit(3);
                            }
                        }
                        break;
                    default :
                        printf("파일을 복사하지 않습니다.\n");
                        break;
                }
            }
        }
    }
    break;
}

}
if(optind<argc)
{
    fprintf(stderr,"남은 arguments 처리해야함: ");
    int i;
    for(i=optind; i<argc; i++)
    {
        fprintf(stderr,"%s ",argv[i]);
    }
    printf("\n");
}
close(fd1);
close(fd2);
}

```

```

#include<sys/types.h>
#include<fcntl.h>
#include<unistd.h>
#include<stdlib.h>
#include<stdio.h>
#include<sys/errno.h>
#include<string.h>
#define BUF_SIZE 512

int main(int argc, char *argv[])
{
    int fd1,fd2, rd_cnt;
    int opt;
    char alast_opt;
    int check_argc;
    char answer;
    char buf[BUF_SIZE];
    extern char *optarg;
    extern int optind;
    char *file1,*file2;
    extern int opterr;
    opterr=1;

    //printf("argc %d optind %d",argc,optind);

    if(argc-optind<2)
    {
        fprintf(stderr,"파일 2개가 선택되지 않았습니다.\n사용법 : %s -옵션 파일명1 파일명2\n-x : 복사할 파일이
        존재하는 경우 오류메세지 출력후 종료\n-t : 복사할 파일이 존재하는 경우 무조건 지우고 복사\n-y : 복사할 파일이 존재하
        는 경우 사용자에게 메세지 출력, 사용자가 y를 답하는 경우 지우고 복사. 그렇지 않는 경우 종료.\n\n옵션이 지정되지 않으면
        옵션 -x로 동작합니다\n여러옵션을 지정한 경우 마지막 설정한 옵션으로 동작합니다.\n",argv[0]);
        exit(1);
    }

    while((opt=getopt(argc,argv,"x:t:y:"))!=-1)
    {
        //if(argc-optind<3)
        //    opt='x';
        //printf("현재 opt %c",opt);
        file1=argv[argc-optind+1]; //원본파일
        file2=argv[argc-optind+2]; //대상파일

        switch(opt)
        {
            case '?' :
                printf("x로 동작");
            case 'x' :
                if(argc-optind>1)
                //if(argv[optind]==file1)
                {
                    printf("마지막 옵션으로 진행합니다\n");
                    break;
                }
                else
                {
                    fd1=open(file1, O_RDONLY);
                    if(fd1<0)
                    {

```

```

        perror("여는데 오류 발생\n");
        if(errno==ENOENT)
            fprintf(stderr, "복사할 소스파일(%s)가 존재하지 않음.\n", file1);
        exit(2);
    }
    fd2=open(file2, O_CREAT | O_RDWR | O_EXCL , 0644);
    if(fd2<0)
    {
        perror("여는데 오류 발생\n");
        if(errno==EEXIST)
            fprintf(stderr, "복사할 대상파일(%s)이 이미 존재함.\n", file2);
        exit(2);
    }
    while((rd_cnt=read(fd1,buf,BUF_SIZE))>0)
    {
        if(write(fd2,buf,rd_cnt)!=rd_cnt)
        {
            perror("파일을 쓰는중 오류 발생.\n");
            exit(3);
        }
    }
    if(rd_cnt<0)
    {
        perror("파일을 읽는중 오류 발생.\n");
        exit(4);
    }
    break;
}
case 't' :
    if(argc-optind>1)
        //if(argv[optind]==file1)
    {
        printf("마지막옵션으로 진행합니다\n");
        break;
    }
    else
    {
        fd1=open(file1, O_RDONLY);
        if(fd1<0)
        {
            printf("%s\n", file1);
            perror("여는데 오류 발생\n");
            if(errno==ENOENT)
                fprintf(stderr, "복사할 소스파일(%s)가 존재하지 않음.\n", file1);
            exit(2);
        }
        fd2=open(file2, O_CREAT | O_RDWR | O_TRUNC , 0644);
        if(fd2<0)
        {
            perror("여는데 오류 발생\n");
            exit(2);
        }
        while((rd_cnt=read(fd1,buf,BUF_SIZE))>0)
        {
            if(write(fd2,buf,rd_cnt)!=rd_cnt)
            {
                perror("파일을 쓰는중 오류 발생.\n");
                exit(3);
            }
        }
    }
}

```

```

    }
    }
    if(rd_cnt<0)
    {
        perror("파일을 읽는중 오류 발생.\n");
        exit(4);
    }
    break;
}
case 'y' :
    if(argc-optind>1)
    //if(argv[optind]==file1)
    {
        printf("마지막 옵션으로 진행합니다.\n");
        break;
    }
    else
    {
        fd1=open(file1, O_RDONLY);
        if(fd1<0)
        {
            perror("여는데 오류 발생.\n");
            if(errno==ENOENT)
                fprintf(stderr, "복사할 소스파일(%s)가 존재하지 않음.\n", file1);
            exit(2);
        }
        fd2=open(file2, O_CREAT | O_RDWR | O_EXCL , 0644);
        if(fd2<0)
        {
            if(errno==EEXIST)
            {
                printf("대상파일(%s)가 이미 존재합니다. 지우고 복사할까요? : ", file2);

                scanf("%c", &answer);
                switch(answer)
                {
                    case 'y' :
                        fd2=open(file2, O_CREAT | O_RDWR | O_TRUNC , 0644);
                        while((rd_cnt=read(fd1, buf, BUF_SIZE))>0)
                        {
                            if(write(fd2, buf, rd_cnt)!=rd_cnt)
                            {
                                perror("파일을 쓰는중 오류 발생.\n");
                                exit(3);
                            }
                        }
                        break;
                    default :
                        printf("파일을 복사하지 않습니다.\n");
                        break;
                }
            }
        }
        break;
    }
}
}
if(optind<argc)
{
    fprintf(stderr, "남은 arguments 처리해야함: ");
    int i;
    for(i=optind; i<argc; i++)
    {
        fprintf(stderr, "%s ", argv[i]);
    }
    printf("\n");
}
close(fd1);
close(fd2);
}

```

165, 2-16

89%

```
[khm970514@lily ex2]$ cat tmp.bbb
DUP2 : Standard Output Redirection
[khm970514@lily ex2]$ cat unix.txt
Hello Unix
[khm970514@lily ex2]$ cat unix.bak
Hello Unix
[khm970514@lily ex2]$ gcc -o mycp3 mycp7.c
[khm970514@lily ex2]$ ./mycp3
파일 2개가 선택되지 않았습니다.
사용법 : ./mycp3 -옵션 파일명1 파일명2
-x : 복사할 파일이 존재하는 경우 오류메세지 출력후 종료
-t : 복사할 파일이 존재하는 경우 무조건 지우고 복사
-y : 복사할 파일이 존재하는 경우 사용자에게 메세지 출력, 사용자가 y를 답하는 경우 지우고 복사, 그렇지 않는 경우 종료.

옵션이 지정되지 않으면 옵션 -x로 동작합니다
여러옵션을 지정한 경우 마지막 설정한 옵션으로 동작합니다.
[khm970514@lily ex2]$ ./mycp3 -x unix.txt unix.bak
여는데 오류 발생
: File exists
복사할 대상파일 (unix.bak)이 이미 존재함.
[khm970514@lily ex2]$ ./mycp3 -t tmp.bbb unix.bak
남은 arguments 처리해야함: unix.bak
[khm970514@lily ex2]$ cat unix.bak
DUP2 : Standard Output Redirection
[khm970514@lily ex2]$ ./mycp3 -y unix.txt unix.bak
대상파일 (unix.bak)가 이미 존재합니다. 지우고 복사할까요? : y
남은 arguments 처리해야함: unix.bak
[khm970514@lily ex2]$ cat unix.bak
Hello Unix
[khm970514@lily ex2]$
```

교수님께서 다시 프로그래밍 해보라고 하셔서 다시 작성하게 되었습니다. 먼저 프로그램의 맨처음에 argc확인을 위해서 조건문을 넣었습니다. 조건은 다음과같이 생각하여

./m unix.txt unix.bak : argc 3 optind 1, ./m -x uni.txt : argc 3 optind 1, ./m -x -t unix.txt unix.bak : argc 5 optind 1, ./m -x unix.txt unix.bak : argc 4 optind 1, ./m unix.txt : argc2 optind1 (argc-optind)한값이 2보다 작은경우를 조건으로 주었습니다.

옵션으로 인해 argv[1],argv[2]등을 직접 사용할 수 없기 때문에 getopt 루프문 이후에 file1, file2에 argv[argc-optind+1], argv[argc-optind+2]를 이용하여 원본파일과 대상파일을 할당해주었습니다. open, read, write의 경우 stderr가아닌 perror로 처리해주었습니다.

변수이름을 fd1, fd2, rd_cnt, opt, answer등 의미를 임작할 수 있는 변수명을 활용해주었습니다.

<문제점>

1. 프로그램에 옵션이 없는 경우 (./mycp3 unix.txt unix.bak) x옵션으로 동작하지 못합니다. 처음 default와 case '?'의 경우를 활용하려고 하였는데 몇 번을 시도해도 되지않아서 생각해보니 아예 getopt함수를 호출하지 못하여 발생하는 문제인 것 같습니다. 그래서 이러한경우엔 getopt 루프문 이전에 argc검사단계에서 x옵션처럼 동작하려고 file1이 읽기가가능하면 실행하려고 작성하였는데 ./mycp3 -x unix.txt 이런 명령행 인자를 받으면 -x 또한 읽기가 능하여 실패하였습니다. 다른방법으로도 몇 번 시도해보았지만 시간이 다되어 제출하려고합니다.

2. 마지막에 설정한 옵션으로 동작하는 경우, 처음에는 optind값을 임의로 +하여 사용했는데 optind임의로 변경하는 것이 부적절하다 하여서 argc-optind>1인경우와 argv[optind]가 file1인경우를 생각하였습니다. optind값은 getopt 함수가 한번 호출될때마다 +1 되기 때문에 이 조건을 사용하였는데 ./mycp3 -x -t unix.txt unix.bak같은 명령을 주게되면 -x는 스킵되지만 -t로 동작하지 않는등의 문제가 발생했습니다. 이부분에 대해서는 저는 getopt루프문을 돌려 함수가 호출되면 optind가 1이 늘어나는점을 활용하여 조건을 내건 것인데 이상하게 optind가 +1되지않아 -x가 스킵되고 -t로 동작하려는 순간 optind가 그대로이기 때문에 -t또한 스킵하려는 문제가 발생하였습니다.

일주일을 연장시켜주셨지만 도저히 더 이상 방도가 떠오르지않아 응용 5번에 집중하고자 제출하려고합니다. 아직 스스로 부족한점이 많네요.. 감사합니다.