

Xilinx Zynq FPGA, TI DSP, MCU **기반의 프로그래밍 및 회로 설계** **전문가 과정**

<리눅스 시스템 프로그래밍>
2018.03.20-18일차

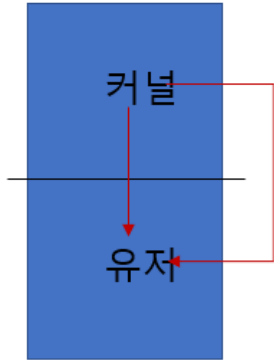
강사 - 이상훈
gcccompil3r@gmail.com

학생 - 안상재
sangjae2015@naver.com

1. SW 3 대장

-OS, compiler, DB

2. 커널 - 유저(영역)



3. system call : open, read, close

- fd : 파일 descriptor

- read(fd, buf, 읽을크기)

- write(fd, buf, 쓸 크기)

- open, close 는 짝(파일을 open 하면 close 함으로써 메모리 누수를 막음.)

4. 헤더 파일

<> : 시스템 헤더 (1.컴파일, 2.os)

" " : 사용자 정의 헤더

```
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
```

```
int main(int argc, char **argv)    // argc는 명령어라인에 입력된 갯수, argv는 종류의 배열
{
    int fd = open(argv[1],O_RDONLY); // 명령어라인에 입력된 인자의 1번째 파일을 읽기전용으로 open함.
    int line = 0;
    int word = 0;
    int flag = 0;
    int cnt = 0;
    char ch;

    if(argc != 2) 인자가 2개가 아니면
    {
        printf("You need 1 more parameter\n");
        printf("Usage: mywc filename\n");
        exit(-1); // 종료
    }

    if((fd = open(argv[1], O_RDONLY)) < 0)    // 파일이 잘못 열리면
    {
        perror("open() "); // 에러 표시
        exit(-1); // 종료
    }

    while(read(fd, &ch, 1))    // fd에서 1문자 읽어서 ch 변수에 저장.
    {
        cnt++; // 카운트 증가

        if(ch == '\n')
            line++; // ch가 개행이면 line 증가

        if(ch != '\n' && ch != '\t' && ch != ' ')    // ch가 개행,탭,공백 모두 아니면(문자 또는 숫자라면)
        {
            if(flag == 0)
            {
                word++;
            }
        }
    }
}
```

```

        flag = 1;
    }
}
else
{
    flag = 0;
}
}
//close(fd);

printf("%d %d %d %s\n", line, word, cnt, argv[1]);

return 0;
}

```

```

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

```

```

// polling

```

```

/*
int main(void)
{
    FILE *fp = fopen("mycat.c","r");        // mycat.c 파일을 읽기전용으로 open함.
    char buf[1024] = "Wn";
    int ret;

    while(ret = fread(buf,1,sizeof(buf),fp))
    {
        usleep(1000000);
        fwrite(buf,1,ret,stdout);
    }
    fclose(fp);

    return 0;
}
*/

```

```

// system call

```

```

/*
int main(int argc, char **argv)
{
    int fd, ret;

```

```

char buf;

if(argc != 2)    // 명령행 인자가 2개가 아니면
{
    printf("Usage: mycat filenameWn");
    exit(-1);    // 종료
}

fd = open(argv[1], O_RDONLY);    // 1번째 파일을 읽기전용으로 open함.

while(ret = read(fd,buf,sizeof(buf))) // fd파일에서 buf의 크기만큼 읽어서 buf에 저장함.
{
    write(1, buf, ret);    // 표준 출력(모니터:1)에 buf의 데이터를 ret크기만큼 출력함.
}
close(fd);

return 0;
}
*/

```

```

int main(int argc, char **argv)
{
    char buf[1024];
    int fd,ret;

    while(1)
    {
        ret = read(0, buf, sizeof(buf)); // 표준 입력(키보드:0)에서 buf의 크기만큼 read해서
buf에 저장하고 read한 길이를 ret에 저장함.
        if(ret)
            write(1,buf,ret);// 표준출력(모니터:1) 에 buf의 데이터를 ret크기만큼 출력함.
    }

    return 0;
}

```

```

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

```

```

int main(void)
{
    int filesdes;
    off_t newpos;

    filesdes = open("data1.txt", O_RDONLY);    // data1.txt파일을 읽기전용으로 읽음.

    newpos = lseek(filesdes, (off_t)0, SEEK_END); // off_t에서 0만큼의 오프셋에서 SEEK_END만큼

```

filedes에서 데이터를 읽음.

```
printf("file size :%ld\n", newpos);
```

```
//return 0;
```

```
}
```

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

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

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

```
#include <fcntl.h>
```

```
int main(void)
```

```
{
```

```
    int filedес;
```

```
    off_t newpos;
```

```
    filedес = open("data1.txt", O_RDONLY);    // data1.txt 파일을 읽기전용으로 open함.
```

```
    newpos = lseek(filedes, (off_t)0, SEEK_END); // 0에서 SEEK_END 까지 filedес에서 문자열을  
읽어서 크기를 반환함.
```

```
    printf("file size :%ld\n", newpos);
```

```
    //return 0;
```

```
}
```

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

```
#include <fcntl.h>
```

```
int main(void)
```

```
{
```

```
    int fdin, fdout;
```

```
    ssize_t nread;
```

```
    char buf[1024];
```

```
    fdin = open("temp1.txt", O_RDONLY); // temp1.txt파일을 읽기전용으로 읽음.
```

```
    fdout = open("temp2.txt", O_WRONLY|O_CREAT|O_TRUNC,0644);
```

```
    while((nread = read(fdin,buf,1024)) > 0)    //fdin파일에서 1024크기만큼 read해서 buf에  
저장함.
```

```
    {
```

```
        if(write(fdout, buf, nread) < nread)    // buf의 데이터를 fdout 파일에 nread만큼
```

```
write함.
```

```
        {
```

```
            close(fdin);
```

```
            close(fdout);
```

```
        }
```

```
    }
```

```
        close(fdin);
        close(fdout);

    return 0;
}
```

```
#include <fcntl.h>
#include <unistd.h>
```

```
int main(void)
{
    int filed1, filed2;

    filed1 = open("data1.txt",O_WRONLY|O_CREAT|O_TRUNC,0644); //data1.txt파일을 읽기,쓰기,
파일생성, 기존 데이터 삭제의 옵션으로 open함.
    filed2 = creat("data2.txt",0644);    // data2.txt파일을 생성함.

    close(filed1);
    close(filed2);

    return 0;
}
```

```
#include <unistd.h>
#include <fcntl.h>
```

```
int main(void)
{
    int fdin, fdout;
    ssize_t nread;
    char buf[1024];

    fdin = open("temp1.txt", O_RDONLY); // temp1.txt파일을 읽기전용으로 open함.
    fdout = open("temp2.txt", O_WRONLY|O_CREAT|O_TRUNC,0644); // temp2.txt파일을
읽기전용,파일생성,기존 데이터 삭제 옵션으로 open함.

    while((nread = read(fdin,buf,1024)) > 0)    // fdin파일에서 1024크기만큼 read해서 buf에
저장함.
    {
        if(write(fdout, buf, nread) < nread)    // fdout파일에 nread크기만큼 buf의 데이터를
write함.
        {
            close(fdin);
            close(fdout);
        }
    }

    close(fdin);
    close(fdout);
}
```

```

        return 0;
    }
}

#include <stdlib.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#define ERROR -1

int main(void)
{
    int filedес;
    char pathname[] = "temp.txt";

    if(filedes = open(pathname, O_CREAT | O_RDWR|0644) == ERROR) // pathname의 문자열의
파일을 파일생성, 읽기쓰기 옵션으로 open함.
    {
        printf("File Open Error!\n");
        //exit(1);
    }

    printf("fd = %d\n", filedес);
    close(filedes);
    return 0;
}

```

***scanf 프로그램**

```

#include "my_scanf.h"

int my_scanf(char *buf, int size)
{
    int nr = read(0, buf, size);    // 표준입력(키보드:0)에서 size만큼 read해서 buf에 저장함.
    return nr;
}

#ifndef __MY_SCANF_H__    // __MY_SCANF_H__이 정의가 안되어 있으면 밑으로 내려감. (헤더파일이
여러번 중복이 되는 것을 막음.)
#define __MY_SCANF_H__    //
#include <fcntl.h>
#include <unistd.h>

int my_scanf(char *, int);    // prototype 선언.

#endif

```



```
#include <string.h>
#include <stdio.h>
#include "my_scanf.h" // 사용자 정의함수 my_scanf.h 를 include함.
```

```
int main(void)
{
    char buf[1024] = {0};
    int nr;

    nr = my_scanf(buf, sizeof(buf));
    printf("nr = %d\n",nr);
    write(1, buf, nr);

    return 0;
}
```