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

#17

강사 : Innova Lee(이 상훈)

학생 : 김 시윤

1.리눅스 시스템 프로그래밍

운영체제 – 하드웨어를 제어하기 위한 소프트웨어.

*Software 는 크게 3개로 나뉜다.

1. OS
2. Compiler
3. DB

그중에 운영체제(OS)는 하드웨어 뿐만 아니라 소프트웨어도 제어한다.

GNU – GNU is Not Unix

돈을 받지 않는 소프트웨어의 탄생 그게 바로 리눅스.

Free 와 Open 의 차이.

Free SW = 자유 O 책인 X
Open SW = 자유 O 책임 O

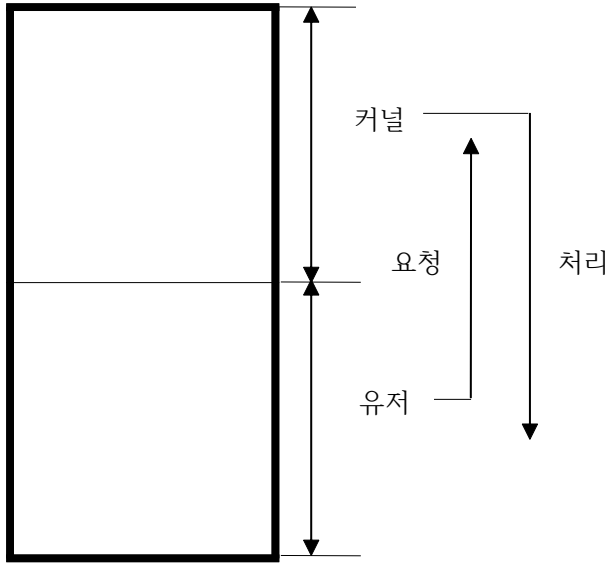
오픈소프트웨어는 정해진 라이센서를 위반해서는 안된다.

모놀리식 커널- 분리x
마이크로 커널- 분리

2.인터럽트

-내가 알고있는 인터럽트의 개념-
인터럽트는 어떤 사건 (전기적 신호)가 일어나면 하던일을 멈추고 그 사건을 해결한후 멈춘 지점으로 다시 돌아오는 것을 말한다.

시스템콜- 유일한 소프트웨어 인터럽트. HW에 의한 전기적 인터럽트가 아니다.



[그림 1] 시스템 콜 개념도.

시스템콜은 유저가 어떤 상황을 발생시켜 요청하면 그 사건을 커널에서 처리하고 다시 사용자한테 돌려준다.

3.시스템콜 예제

1. 용어 정리.

```
#include <fcntl.h>      //file control .h
#include <unistd.h>     //unis standard.h
```

O_WRONLY – 쓰기만 가능하다.
O_RDONLY – 읽기만 가능하다.
O_CREAT – 파일을 생성한다.
O_TRUNC – 실행할때마다 파일을 리셋한다.
O_EXCL – 중복파일 생성을 방지한다.
O_RDWR – 일기 쓰기 둘다 가능하다.

open("파일명.확장자",옵션,0644);

ex) open("temp.txt",O_CREAT | O_WRONLY | O_TRUNC);
temp.txt 라는 파일을 생성하고 쓰기만 가능하고 실행할때마다 초기화해라

read 와 write명령어

우선 **read**는 RDWR이나 RD명령어를 사용했을 경우 가능

```
int fd,nread;
char buf[1024]={0};

fd = open("temp.txt",O_RDONLY)

nread= read(fd,buf,1024);
```

1024 크기만큼 fd를 읽고 fd를 버퍼에 넣어라

write

```
int fd;
char buf[1024]={0};

fd=open("temp.txt",O_WRONLY | O_CREAT | O_TRUNC);

write(fd,buf,1024);
```

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

int main(void)
{
        int filedes;
        char pathname[]= "temp.txt";//포인터 변수
        if((filedes = open(pathname, O_CREAT | O_RDWR, 0644)) == ERROR)//filed
        {

                printf("File Open ERROR!\n");
                exit(1);
        }

        printf("fd = %d\n",filedes);
        close(filedes);

        return 0;
        }
~
~
#include <stdlib.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#define ERROR -1

int main(void)
{
        int filedes;
        char pathname[]= "temp.txt";//포인터 변수
        if((filedes = open(pathname, O_CREAT | O_RDWR | O_EXCL, 0644)) == ERROR)//
        {

                printf("File Open ERROR!\n");
                exit(1);
        }

        printf("fd = %d\n",filedes);
        close(filedes);

        return 0;
        }
~
```

파일생성.
O_EXCL 이 생기면 첫 실행 빼고는 이프문을 통과하지 못한다.
왜냐하면 temp.txt가 한번 통과한후 이미 생성되어 이프문이 거짓이 되기 때문이다.

MY_CP

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


int main(int argc, char **argv)
{
        int i;
        int in, out,nread;
        char buf[1024]={0};

        if(argc !=3)
        {
                printf("인자 입력 3개 하라고 \n");
                exit(-1);
        }

        for(i=0; i<argc; i++)
        {
                printf("당신이 입력한 인자는 = %s\n",argv[i]);

        }

        in = open(argv[1],O_RDONLY);
        out = open(argv[2],O_CREAT | O_WRONLY | O_TRUNC);
        while((nread = read(in,buf,1024))>0)
        {
                if(write(out,buf,nread)<nread)
                {

                close(in);
                close(out);
                printf("비정상 종료 \n");
                exit(-1);
                }
        }

                close(in);
                close(out);
                printf("정상종료\n");
                        return 0;
        }
```

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

int main(int argc, char **argv)
{
        int i;
        int in, out,nread;
        char buf[1024]={0};

        in = open(argv[1],O_RDONLY);
        out = open(argv[2],O_CREAT | O_WRONLY | O_TRUNC);
        while((nread = read(in,buf,1024))>0)
        {
                if(write(out,buf,nread)<nread)
                {

                close(in);
                close(out);
                printf("비정상 종료 \n");
                exit(-1);
                }
        }

                close(in);
                close(out);
                printf("정상종료\n");
                        return 0;
        }
```

argv를 응용하여 복사 기능을 만들었다.

Cat

```c
#include <stdio.h>

int main(void)
{
        FILE *fp= fopen("mycat.c","r");
        char buf[1024]="\0";
        int ret;
        while(ret = fread(buf,1,sizeof(buf),fp))
        {
                usleep(1000000);
                fwrite(buf,1,ret,stdout);
        }
        fclose(fp);
        return 0;
}
```

시스템콜을 하지않고 구현한결과 유슬립 1초 때매 느린 결과가 있었다.

시스템콜을 해 구현해보면

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

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

        fd=open(argv[1],O_RDONLY);
        while(ret=read(fd,buf,sizeof(buf)))
        {
                write(1,buf,ret);
        }
        close(fd);
        return 0;
}
```

//스샷을 못찍어 머리에서 쥐어짜서 짜봐서 돌려봤는데 성공했다!!gg

여기서 중요한건 write(**1**,buf,ret);
빨간색으로 표시된 1이다.

시스템에서 인식하는
0은 입력
1은 출력
2는 에러 이다
따라서 우리는 3부터 다룬다.

Scanf

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


int main(int argc, char **argv)//
{
        int ret,fd2;
        char buf[1024];

        fd2= open("scanf2.txt",O_WRONLY | O_CREAT | O_TRUNC,0644);
        if(ret = read(0,buf,sizeof(buf)))
        {
                write(1,buf,ret);
                write(fd2,buf,ret);
        }
        close(fd2);
        return 0;
}
~
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>


int main(int argc, char **argv)//
{
         int ret,fd;
         char buf[1024];
fd=open("scanf4.txt",O_WRONLY | O_CREAT | O_TRUNC,0644);

ret=read(0,buf,sizeof(buf));


write(1,buf,ret);
write(fd,buf,ret);

return 0;
}
```

위에는 내가짠 소스코드고 아래는 선생님의 소스코드를 비교해보았다.
잘 생각해보면 조건문 없이도 상관없었다.

WC

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

int main(int argc,char **argv)
{
        int fd = open(argv[1],O_RDONLY);
        int line=0;
        int word=0;
        int flag =0;
        int cnt = 0;
        char ch;
        if(argc !=2)
        {
                printf("You need 1 more parameter\n");
                printf("Usage: mywc filename\n");
                exit(-1);
        }

        if((fd = open(argv[1], O_RDONLY))<0)
        {
                perror("open() ");
                exit(-1);//프로세스 상태값
        }
        while(read(fd,&ch,1))//&ch 배열은 주소라 배열로 만든것 1byte씩 읽어라
        {
                cnt++;
                if(ch == '\n')
                        line++;
                if(ch != '\n' && ch != '\t' && ch !=' ')
                {
                        if(flag ==0)
                        {
                                word++;
                                flag =1;
                        }
                }
                else
                {
                        flag=0;
                }
        }
        close(fd);
        printf("%d %d %d %s\n",line,word,cnt,argv[1]);


return 0;
}
```

수업시간에 word 는 단어가 몇 개 나왔나 체크하는건데 알파벳수 공백을 제외한 글자수로 착각하여 이해를 못했었다.

그래서 그렇게 생각한 김에 글자 알파벳수를 나타내는 부분까지 추가해보기로 하였다.

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

int main(int argc,char **argv)
{
        int fd = open(argv[1],O_RDONLY);
        int line=0;
        int word=0;
        int flag =0;
        int cnt = 0;
        char ch;
        int alpabet=0;
        if(argc !=2)
        {
                printf("You need 1 more parameter\n");
                printf("Usage: mywc filename\n");
                exit(-1);
        }

        if((fd = open(argv[1], O_RDONLY))<0)
        {
                perror("open() ");
                exit(-1);//프로세스 상태값
        }
        while(read(fd,&ch,1))//&ch 배열은 주소라 배열로 만든것 1byte씩 읽어라
        {
                cnt++;
                if(ch == '\n')
                        line++;
                if(ch != '\n' && ch != '\t' && ch !=' ')
                {
                                alpabet++;


                        if(flag ==0)
                        {
                                word++;
                                flag =1;
                        }
                }
                else
                {
                        flag=0;
                }
        }
        close(fd);
        printf("line = %d, word =  %d, cnt = %d, alpabet =  %d, file_name= %s\n",line,word,cnt,alpabet,argv[1]);

return 0;
}
```

alpabet 이라는 변수를 추가하여 공백 엔터 탭을 제외한 글자에수 숫자가 증가하게 설정하였다.
결과가 정확한걸 확인하였다.