TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 최대성
c3d4s19@naver.com

2018.03.19. – 018일차 수업

<SW Interrupt>

유저 ----요청(system call)----> 커널

ex) open(), creat(), close(), read(), write(), ...

커널의 주요 구조체

task_struct{

files_struct ----(포인터)----> files

signal_struct

mm_struct

}

files{

f_POS --------> **fd

file_operations

inode

}

*fd[index] -> file descriptor(파일 기술어)

index값 -> 표준입력: 0 / 표준출력: 1 / 에러: 2 / 새 파일: 3 / 새 파일: 4 / ...

(open(), creat()의 return값)

close([파일 index 숫자])

-> 파일 닫기

read([파일 index 숫자], [버퍼], [몇 byte 읽을지])

-> return: [몇 byte 읽었는지]

write([파일 index 숫자], [버퍼], [몇 byte 쓸지])

-> return: [몇 byte 썼는지]

lseek([파일 index 숫자], [시작 위치], [길이])

-> return: [몇 byte 읽었는지]

creat([경로이름], 0644)

-> return: [파일 index 숫자]

open([경로이름], [설정], 0644)

-> return: [파일 index 숫자]

O_CREAT : 새로 만듬

O_TRUNC : 자르기(중복시 덮어쓰기) //truncation

O_RDONLY : 읽기 전용

O_WRONLY : 쓰기 전용

O_RDWR : 읽기 쓰기 둘다 가능

O_EXCL : 파일 있으면 오류 리턴, 없으면 정상 처리 //exclude

사용 예 : open(pathname, O_CREAT | O_RDWR | O_TRUNC, 0644)

---

* 최소 알아야할 리눅스 명령어

1. mv [이름 변경할 파일] [변경 이름]

-> 파일 이름 변경

2. cp [복사할 파일] [원하는 이름]

-> 파일 복사

3. rm -rf [파일명]

-> 파일삭제

4. ls

-> 현재 디렉토리 뭐가있는지

5. cd [이동 디렉토리명]

-> 디렉토리 이동

6. mkdir [파일명]

-> 디렉토리 만들기

7. gcc [파일명]

-> 프로그램 디버깅

8. mkfifo

-> FIFO(이름있는 pipe) 생성

9. touch [옵션] [파일명]

-> 파일의 날짜 시간 정보 변경

10. cat [파일명]

-> 파일 밖에서 내용 보기

11. tar -cvf [파일명.tar] [폴더명]

-> 압축하기

-xvf [파일명.tar]

-> 압축풀기

-zcvf [파일명.tar.gz] [폴더명]

-> tar.gz로 압축하기

-zxvf [파일명.tar.gz]

-> tar.gz 압축풀기

---

* 커널창 나누기 관련 명령어

:vs -> 수직으로 분리

:sp -> 수평으로 분리

:q -> 커서 있는 창 종료

:e [파일명] -> 파일명으로 들어감(없으면 생성)

Ctrl + ww -> 창 넘어가기

:![명령어] -> 창 안나가고 명령어 잠깐 실행

/[찾을단어] -> 단어 찾기

:set hlsearch -> 블록처리

---

* 헤더파일 추가하고 컴파일 할때

gcc -o [실행 파일 이름] [file1.c] [file2.c] ...

헤더파일은 .c파일과 붙어있으면 됨

---

* C언어 파일 입출력

fopen([파일명],[모드 'r','w','a'])

-> return: (FILE*)[open파일 포인터]

fwrite([쓸 문자열의 주소], [문자열 사이즈], [반복 횟수], [쓸 파일 포인터])

-> [문자열 사이즈] * [반복 횟수] 만큼의 길이가 파일에 써진다

fread([저장할 변수], [사이즈], [반복 횟수], [읽을 파일 포인터])

fclose([닫을 파일 포인터])

파일 입출력 사용 예제

```c
#include <stdio.h>
int main() {
        FILE* fp = fopen("data2.txt", "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;
}
```

---

* argc, argv로 프로그램에 인자 전달하기

main 함수의 원형

int main(int argc, char** argv)

int main(int [인자 개수], char** [문자열 배열(index마다 인자 내용이 들어가있음)])

arg 인자 사용 간단 arg_test.c 예제

```
#include <stdlib.h>
#include <stdio.h>
int main(int argc, char *argv[]) {
        int i;
        for (i = 0; i < argc; i++)
                printf("argv[%d]: %s₩n", i,
argv[i]);
        exit(0);
}
```

arg_test.c를 컴파일해서 실행파일을 만들고

$./arg_test a bc def ghij

라고 입력하면

argv[0]: ./arg_test

argv[1]: a

argv[2]: bc

argv[3]: def

argv[4]: ghij

라는 결과가 나온다

arg_test.c 관련자료 출처:

http://gday2code.tistory.com/72

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h> // unix standard
#include <fcntl.h> // file control
int main(int argc, char** argv) {
        int fd, ret;
        char buf[1024];
        if (argc != 2) {
                printf("인자 2개 필요!₩n");
                exit(-1);
        }
        write(1, argv[1], strlen(argv[1]));
        printf("₩n");
        ret = read(0, buf, sizeof(buf));
        write(1, buf, sizeof(ret));
        printf("sizeof(argv[1]) = %d₩n",
sizeof(argv[1]));
        close(fd);
        return 0;
}
```

```
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>        //perror() 함수 사용 목적
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() ");        //무슨
에러가 발생했는지 출력 print error
                exit(-1);
        }
        while (read(fd, &ch, 1)) {
                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;
}
```