TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

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

2018.03.20. – 019일차 수업

---

* 리눅스 커널 초기 설정

mkdir kernel

-> 커널 디렉토리 생성

cd kernel

-> 커널 디렉토리 이동

wget https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.4.tar.gz

-> [웹]에서 get 하는 명령어

<설치 완료 후>

tar zxvf linux-4.4.tar.gz

-> 압축 풀기

cd linux-4.4

-> linux-4.4 파일 생성

ls

->설치파일 확인 후 파일이 30개 미만이면 다시 받기

sudo apt-get install ctags cscope

-> /kernel/linux-4.4 위치에서 install함

아래 ".vimrc"과 "mkcscope.sh" 파일 만들고 코드작성

vi ~/.vimrc

<.vimrc 코드 내용>

```
set nu
set ts=4
set autoindent

"ctags 설정"
set  tags=/home/ds/kernel/linux-4.4/tags  "->주소값에 맞게 변경"
if version >= 500
func! Sts()
        let st = expand("<cword>");
        exe "sts ".st
endfunc
nmap ,st :call Sts()<cr>
func! Tj()
        let st = expand("<cword>");
        exe "tj ".st
endfunc
nmap ,tj :call Tj()<cr>
endif

"cscope 설정"
set csprg=/usr/bin/cscope
set nocsverb
cs add /home/ds/kernel/linux-4.4/cscope.out "->주소값에 맞게 변경"
set csto=0
set cst
func! Css()
        let css = expand("<cword>");
        new
        exe "cs find s ".css
        if getline(1) == ""
                exe "q!"
        endif
endfunc
nmap ,css :call Css()<cr>

func! Csc()
        let csc = expand("<cword>");
        new
        exe "cs find c ".csc
```

```
        if getline(1) == ""
                exe "q!"
        endif
endfunc
nmap ,csc :call Csc()<cr>
func! Csd()
        let csd = expand("<cword>");
        new
        exe "cs find d ".csd
        if getline(1) == ""
                exe "q!"
        endif
endfunc
nmap ,csd :call Csd()<cr>

func! Csg()
        let csg = expand("<cword>");
        new
        exe "cs find g ".csg
        if getline(1) == ""
                exe "q!"
        endif
endfunc
nmap ,csg :call Csg()<cr>
```

vi ~/mkcscope.sh

<mkcscope.sh 코드 내용>

```
#!/bin/sh
rm -rf cscope.files cscope.files
find . ₩( -name '*.c' -o -name '*.cpp' -o -name '*.cc'
-o -name '*.h' -o -name '*.S' ₩) -print > cscope.files
cscope -i cscope.files
```

ctags -R

chmod 755 ~/mkcscope.sh

-> 색상(권한) 변경 - 녹색으로 변함

sudo ~/mkcscope.sh /usr/local/bin/

mkcscope.sh

vi -t task_struct

-> /kernel/linux-4.4 위치에서 task_struct(리눅스 핵심 구조체) 찾기

---

* ctag, cscope 관련 명령어

Ctrl +] -> 커서 위 정보들 찾기

Ctrl + t -> 뒤로 돌아가기

space -> 한 페이지씩 넘김

q [번호선택] -> 선택 번호의 정보로 이동

(번호 선택시 "include/linux/fs.h<<file>>" 이쪽 부분에서

"xxx_struct { " 이렇게 중괄호 열려있는 번호를 선택해야 구조체 정의가 나옴)

---

open()으로 같은 파일을 두 번 여는 경우 진행포인터가 서로 별개로 움직인다.

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

int main() {
        char buf[1024];
        int fd[2];

        fd[0] = open("tar.c", O_RDONLY);
        read(fd[0], buf, 10);
        write(1, buf, 10);

        fd[1] = open("tar.c", O_RDONLY);
        read(fd[1], buf, 10);
        write(1, buf, 10);
        return 0;
}
```

-> 둘다 같은 내용이 print 된다.

---

* tar 압축 프로그램 코딩

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

typedef struct {
        char fname[20];
        int fsize;
}F_info;

int file_size(int fd) {
        int fsize, old;
        old = lseek(fd, 0, SEEK_CUR);
        fsize = lseek(fd, 0, SEEK_END);
        lseek(fd, old, SEEK_SET);
        return fsize;
}

int main(int argc, char* argv[]) {
        int src, dst, ret;
        char buf[1024];
        F_info info;
        int i;
        dst = open(argv[argc - 1], O_WRONLY |
O_CREAT | O_TRUNC, 0644);
        for (i = 0; i < argc - 2; i++) {
                src = open(argv[i + 1],
O_RDONLY);
                strcpy(info.fname, argv[i + 1]);
                info.fsize = file_size(src);
                write(dst, &info, sizeof(info));
                while (ret = read(src, buf,
sizeof(buf)))
                        write(dst, buf, ret);
                close(src);
        }
        close(dst);
        return 0;
}
```

---

* tar 압축 해제 프로그램 코딩

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

typedef struct {
        char fname[20];
        int fsize;
}F_info;

#define min(x, y)       (((x) < (y)) ? (x) : (y))

int main(int argc, char* argv[]) {
        int src, dst, len, ret;
        F_info info;
        char buf[1024];
        src = open(argv[1], O_RDONLY);
        while (read(src, &info, sizeof(info))) {
                dst = open(info.fname, O_WRONLY |
O_TRUNC | O_CREAT, 0644);
                while (info.fsize > 0) {
                        len = min(sizeof(buf),
info.fsize);
                        ret = read(src, buf,
len);
                        write(dst, buf, ret);
                        info.fsize -= ret;
                }
                close(dst);
        }
        close(src);
        return 0;
}
```

---

* lseek 함수 해석

lseek([index], [시작], [설정])

->return: 읽은 byte(글자수)의 개수

SEEK_END -> 시작 길이부터 끝까지 읽기

SEEK_CUR -> 시작 길이부터 현재 포인터 값까지 읽기

SEEK_SET -> 현재 포인터 값을 [시작] 위치로 이동

(이 경우 return값 의미없음)

```
int file_size(int fd) {
        int fsize, old;
        old = lseek(fd, 0, SEEK_CUR);     // 1번
        fsize = lseek(fd, 0, SEEK_END);
        lseek(fd, old, SEEK_SET);// 2번
        return fsize;
}
// 1번 2번 이용해서 SEEK_CUR(현재 포인터값)을
복구시킴
// read() write()에 의해 현재 포인터값이 변경될 수
도 있음
```
