

Xilinx Zynq FPGA,TI DSP, MCU 기반의 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 정한별

hanbulkr@gmail.com

< 19 일차 >

<vim 환경 설정 및 커널 구조 파고들기>

중요 : **task_struct** 중요하다.

(task_struct 안에 중요한 녀석들이 다 들어있다.)

-Driver: 하드웨어를 동작시키는 녀석이 들어있다.

-fs : 파일 시스템

-include : 헤더 파일들을 포함 시키는 녀석, → 함수들의 대해 알 수 있다. ‘{ ‘ 이것을 포함한 녀석을 찾아보면된다.

일단, 사용하기에 앞서서 함수를 참조 하는 기능을 설정해야 한다.

1. 리눅스 커널 관련 압축 파일을 받는다.

```
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ vi queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ gcc -o queue5 queue5.c
queue5.c: In function 'dequeue':
queue5.c:54:8: error: 'tmp' undeclared (first use in this function)
    free(tmp);
       ^
queue5.c:54:8: note: each undeclared identifier is reported only once for each function it appears in
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ vi queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ gcc -o queue5 queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ ./queue5
Segmentation fault (core dumped)
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ gcc -o queue5 queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ vi queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ gcc -o queue5 queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ ./queue5
10
20
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ vi queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ gcc -o queue5 queue5.c
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ ./queue5
10
20
10 Now delete!!!
20
jhb@onestar: ~/My/Homework/hanbyuljung/testtest$ cd
jhb@onestar: ~$ mkdir kernel
jhb@onestar: ~$ cd kernel/
jhb@onestar: ~/kernel$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.4.tar.gz
--2018-03-19 16:09:54-- https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.4.tar.gz
Resolving mirrors.edge.kernel.org (mirrors.edge.kernel.org)... 147.75.69.165, 2604:1380:1000:8100::1
Connecting to mirrors.edge.kernel.org (mirrors.edge.kernel.org)[147.75.69.165]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 132860730 (127M) [application/x-gzip]
Saving to: 'linux-4.4.tar.gz'

linux-4.4.tar.gz      100%[=====] 126.71M  885KB/s
2018-03-19 16:12:59 (702 KB/s) - 'linux-4.4.tar.gz' saved [132860730/132860730]
jhb@onestar: ~/kernel$ tar zxvf linux-4.4.tar.gz
jhb@onestar: ~/kernel$
```

다운로드.

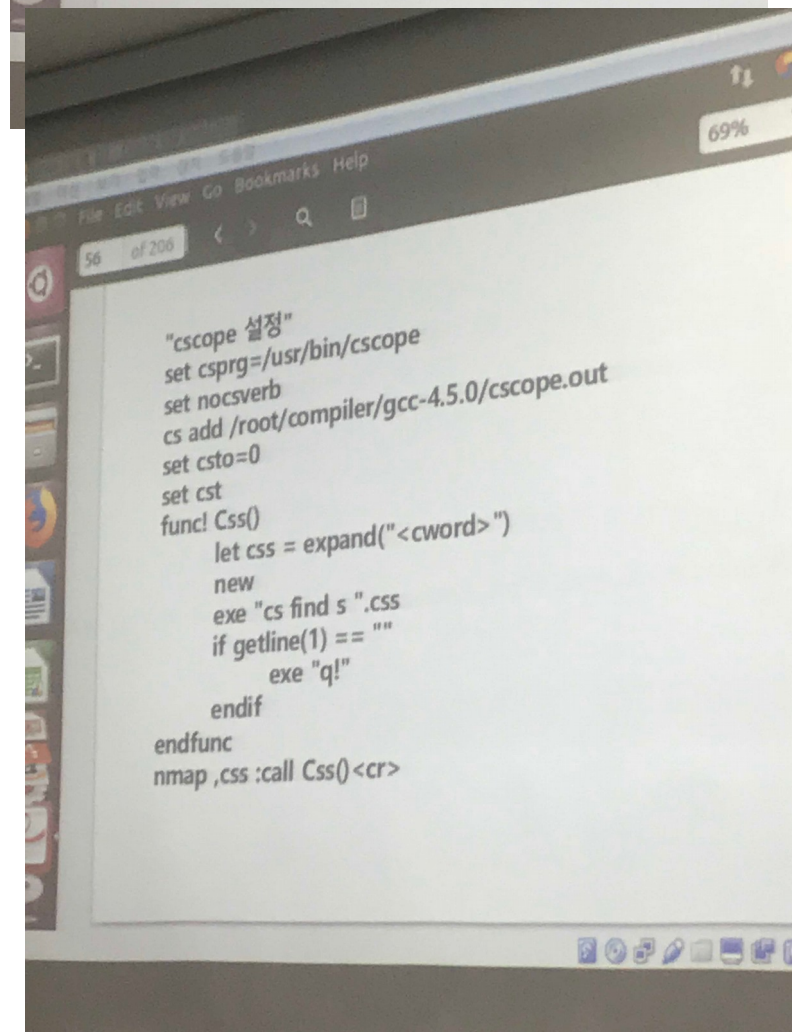
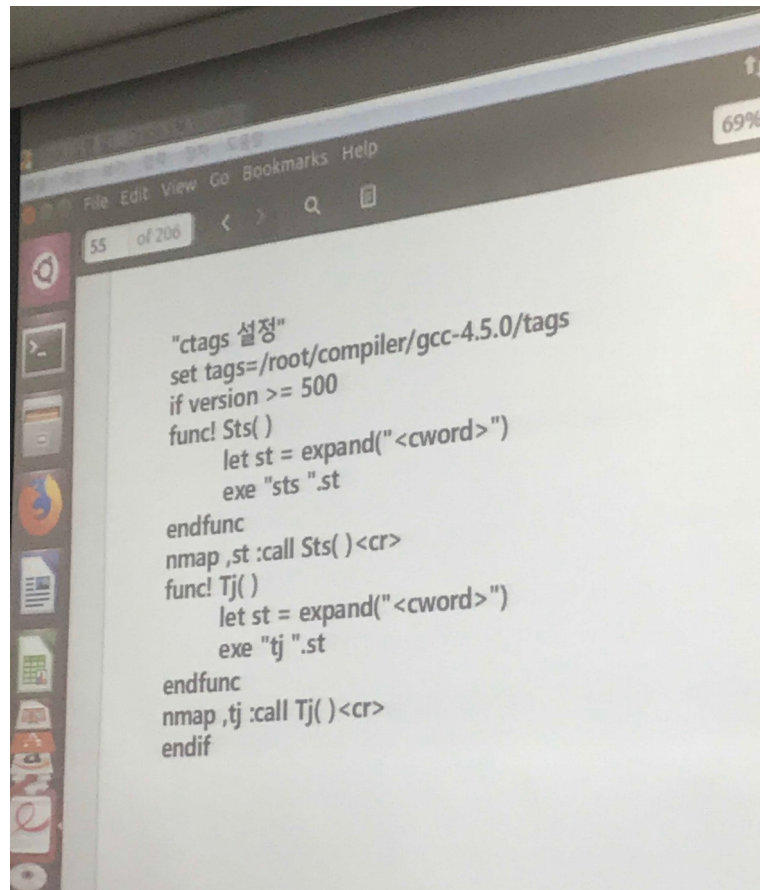
wget https://mirrors.edge.kernel.org/pub/libux/kernel/v4.x/linux-4.4.tar.gz

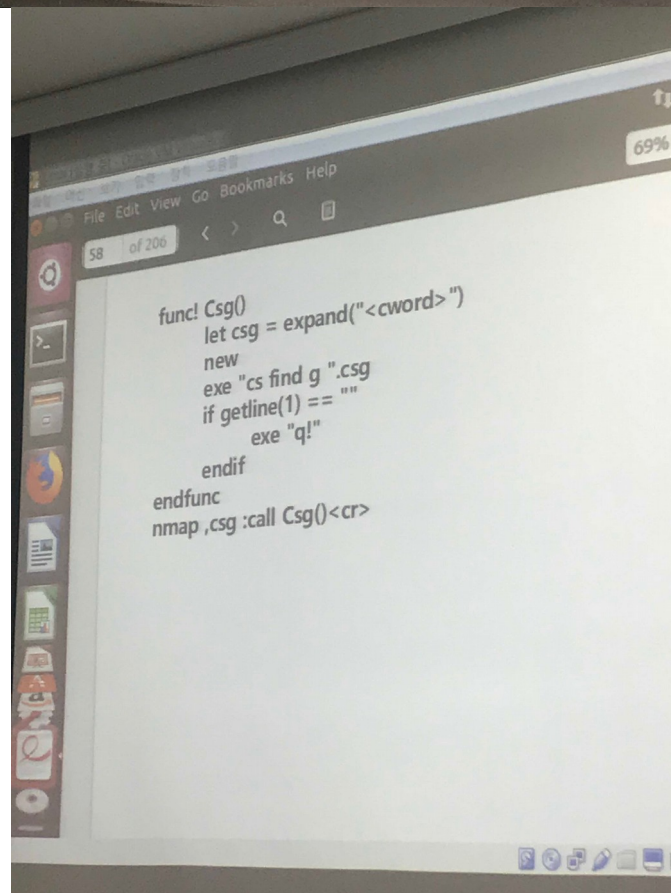
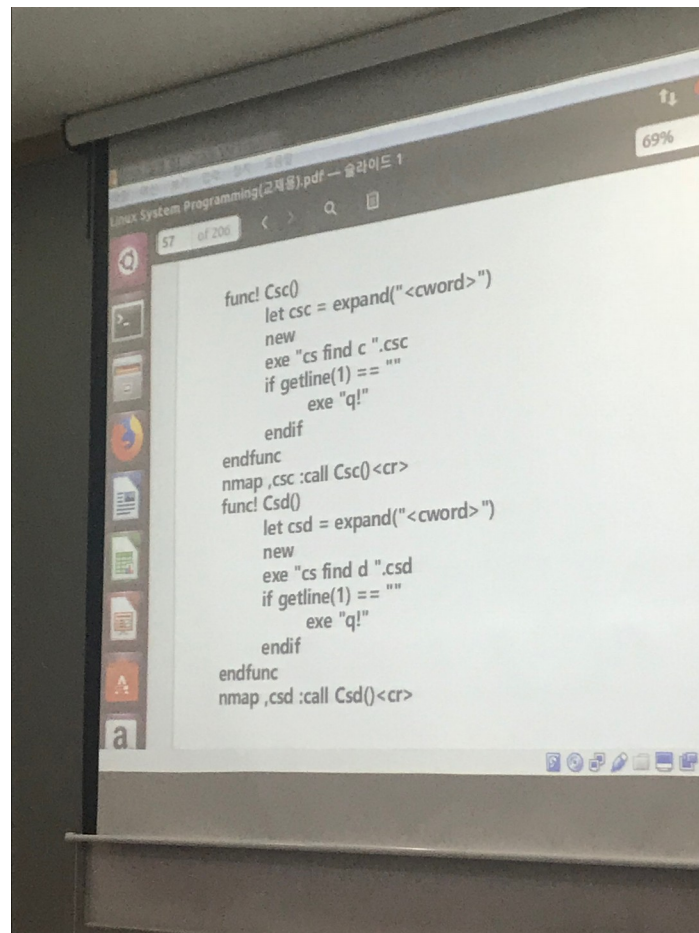
압축풀기.

tar zxvf linux-4.4.tar.gz

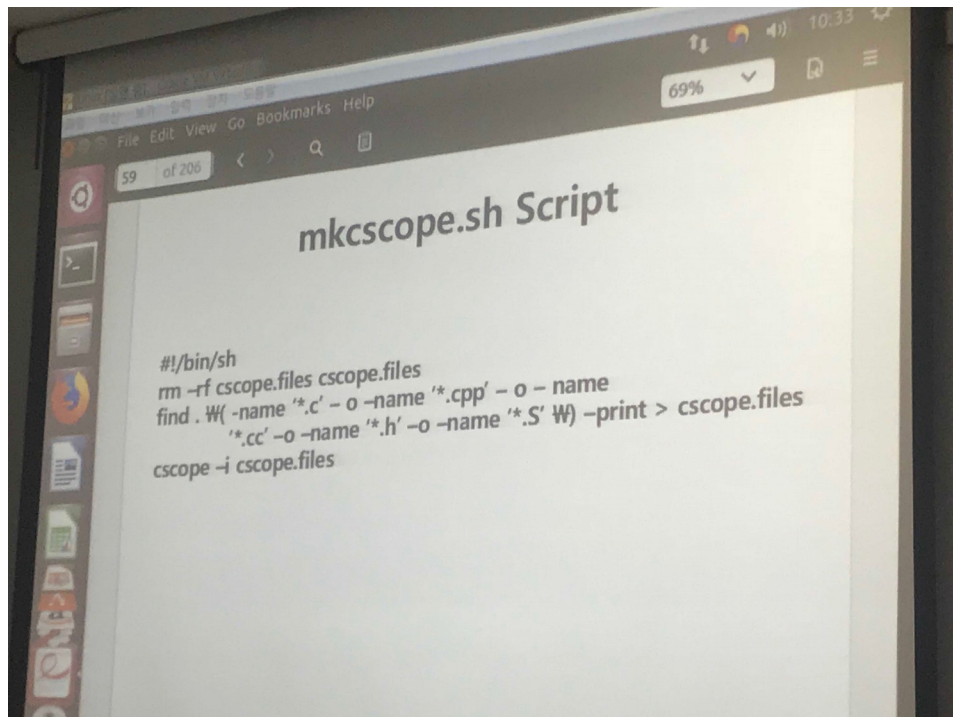
Kernel 폴더에 설치 후 파일이 30 개 미만이면 다시받아야 한다.

2. ~/.vimrc 파일에 설정을 시작한다.

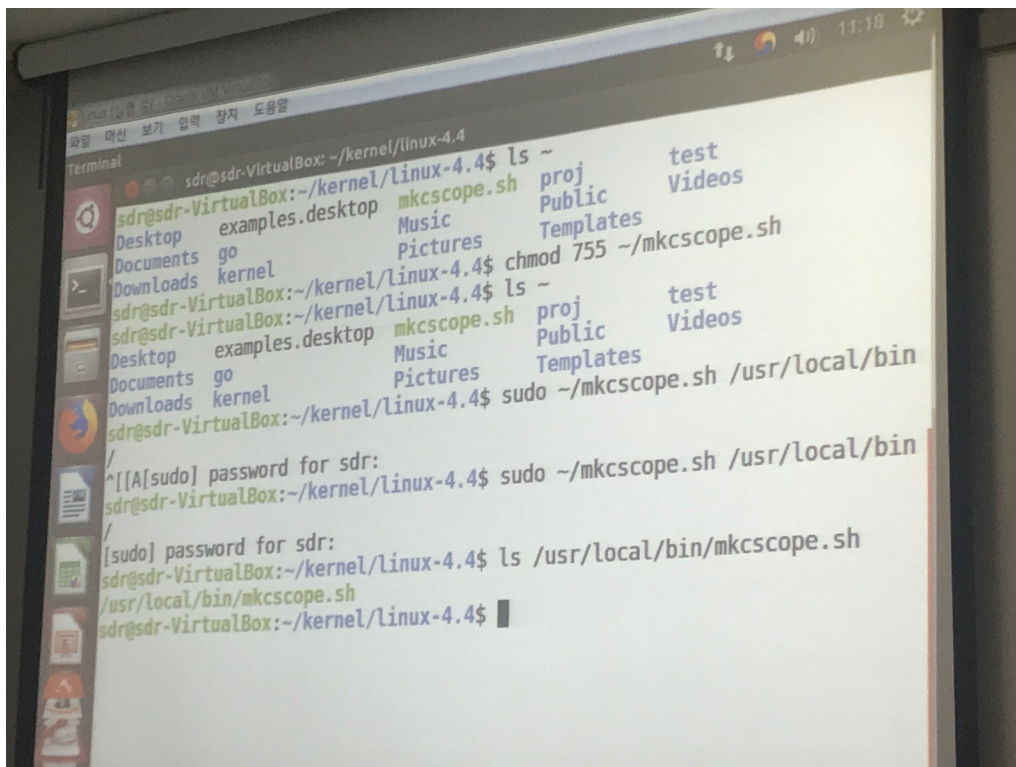




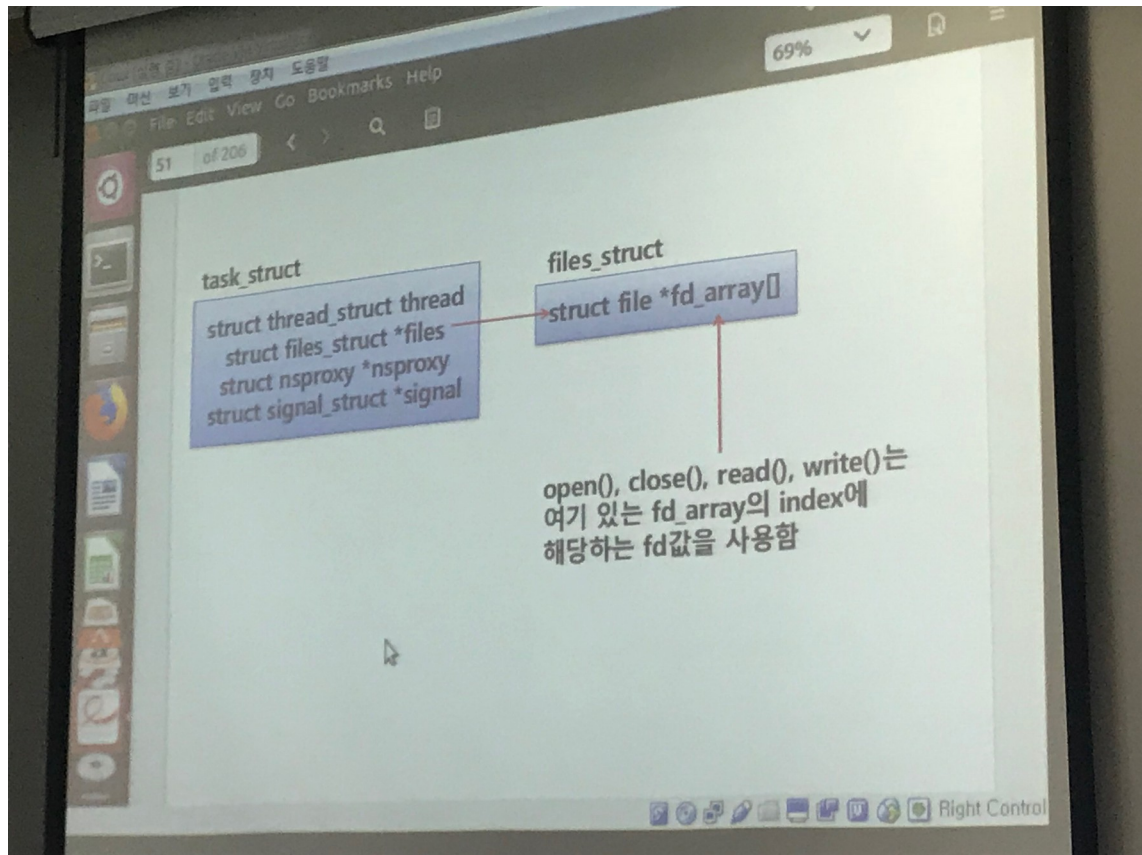
Home 디렉토리에 mkcscope.h Script 를 만든다.



sudo apt-get install ctags cscope
-> /kernel/linux-4.4 위치에서 install 한다.



sudo ~/mkcscope.sh /usr/local/bin/
이러면 뭔가 다운로드를 하는데 끝나면 ctrl+ d 로 나가면 된다.



커널을 도는 순서

1. task_struct → files_struct 을 ctrl +] 을 눌러 함수로 들어감 →
1 번 엔터 → file 을 ctrl +] 을 눌러 함수를 들어감 ->
찾을거({) 찾아서 q 를 눌러서 그 줄의 번호를 입력한다.

*ctrl +] 를 누르면 함수를 들어간다.

*ctrl + t 를 누르면 들어간 함수 이전 단계로 빠져 나온다.

<Quiz 1 번>

임의의 난수를 발생시켜서 이 값을 배열에 저장하고 배열에 저장된 값을 파일에 기록한다.(중복은 안됨)
그리고 이 값을 읽어서 queue 를 만든다. 이후에 여기 저장된 값중 짝수만 선별하여 모두 더한 후에
더한 값을 파일에 저장하고 저장한 파일을 읽어 저장된 값을 출력하도록 한다.

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

```
#define EMPTY 0
```

```

typedef struct __queue
{
    int data;
    struct __queue *link;
}queue;

queue *get_node()
{
    queue *tmp;
    tmp = (queue*)malloc(sizeof(queue));
    tmp->link = EMPTY;
    return tmp;
}

void enqueue(queue **head, int data)
{
    queue *tmp = *head;
    if(*head == NULL)
    {
        *head = get_node();
        (*head)->data = data;
        return ;
    }
    enqueue(&((*head)->link), data);
}

/*
queue *dequeue(queue **head, int data)
{
    queue *tmp = head;
    if(*head == NULL)
    {
        printf("값이 없습니다.");
        return *head;
    }
    else if(head->data != data)
    {
        (*head)->link = dequeue(&head);
    }
    else {
        printf("delete %d\n", data);
        tmp = tmp ->link;
        free(tmp);
        return (*head)->link;
    }
    return head;
}

```

```

void print(queue *haed)
{

}

*/
void print(int *arr, int size)
{
    int i;
    for(i=0; i<size ; i++)
        printf("arr[%d] = %d \n",i, arr[i]);

}
int rand_non_same(int *arr)
{
    int i,buf, res;
    int tmp[10];
    int j=0;
    buf = 0;

    srand(time(NULL));
    while(buf < 10)
    {
        //첨에 값을 받고
        res = rand()%10+1;

        // 값을 받을것을 현재 버퍼가 차있는 곳 까지 검색을 한다.
        // 검색중에
        // 찾는데 마지막 버퍼 위치까지 없을 때 그 값을 다음 버퍼 위치에 저장한다.
        for(i=0; i < 10 ; i++)
        {
            if(res == arr[i]){
                //printf("check1\n ");
                break ;
            }

            else if(i == buf){
                //printf("check2 buf %d\n",buf);
                arr[buf++] = res;
                break;
            }
        }
    }
}

int even_num(int *arr,int len)
{
    int num=0,i=0;

```



```

        for(i=0; i<len ;i++)
            if(arr[i]%2 == 0) num += arr[i];

        return num;
    }

int main(int argc , char **argv)
{
    queue *head = EMPTY;
    int arr[10]={0};
    int i=0;
    int fd,fd2,fread,buf[1024],even;
    int len = sizeof(arr) / sizeof(int);
    char read_buf[1024];

    fd = open(argv[1] , O_RDWR | O_CREAT | O_APPEND ,0644);
    fd2 = open(argv[1] ,O_RDONLY | O_TRUNC,0644);

    rand_non_same(arr);
    print(arr, len);

    printf("\n");

    for(i=0 ;i <len ;i++)
    {
        enqueue(&head, arr[i]);
    }
    printf("다 출력 하였습니다.\n");

    even = even_num(arr, len);

    printf("%d\n",even);

    for(i=0; i<len; i++){
        sprintf(read_buf,"%d \n", arr[i]);
        write(fd, read_buf, strlen(read_buf));
    }

    sprintf(read_buf,"\n 짝수의 합 = %d \n", even);
    write(fd, read_buf, strlen(read_buf));

    fread = read(fd2, buf, sizeof(buf));
    write(1 , buf ,fread);

    close(fd);
    return 0;
}

```

```
arr[0] = 5
arr[1] = 9
arr[2] = 1
arr[3] = 4
arr[4] = 6
arr[5] = 2
arr[6] = 7
arr[7] = 10
arr[8] = 3
arr[9] = 8
```

30
5
9
1
4
6
2
7
10
3
8

짜수의 한 - 30

```
1 5
2 9
3 1
4 4
5 6
6 2
7 7
8 10
9 3
10 8
11
12 짝수의 합 = 30
```

