

TI DSP, MCU 및 Xilinx Zynq FPGA

프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

mhn97@naver.com

Quiz1

임의의 난수를 10개 발생시켜 이 값을 배열에 저장하고 배열에 저장된 값을 파일에 기록한다.

(중복은 안됨) 그리고 이 값을 읽어서 Queue를 만든다.

이후에 여기 저장된 값 중 짝수만 선별하여 모두 더한 후에 더한 값을 파일에 저장하고

저장한 파일을 읽어 저장된 값을 출력하도록 한다.

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

```
struct node{ //구조체 선언
```

```
    int data;
    struct node *link;
```

```
};
```

```
typedef struct node Queue;
```

```
Queue *get_node(){ //노드추가
```

```
    Queue *tmp;
    tmp = (Queue *)malloc(sizeof(Queue));
    tmp -> link = NULL;
    return tmp;
```

```
}
```

```
void enqueue(Queue **head, int data){ // 데이터 입력
```

```
    if(*head == NULL){
```

```

        *head = get_node(); //노드를 추가하는 함수호출
        (*head)->data=data;
        return;
    }
    enqueue(&(*head)->link,data); // head의 값이 NULL이 아니면 재귀호출
}

```

```

void print_queue(Queue *head){ //데이터 출력

```

```

    Queue *tmp;
    tmp = head;

    while(tmp){
        printf("queue : %d\n",tmp->data); //값 출력
        tmp = tmp->link; //head의 값을 링크를 따라 옮긴다
    }
}

```

```

void dequeue(Queue **head,int data){ //데이터 삭제

```

```

    Queue *tmp;
    tmp = *head;
    if((*head) == NULL){ //head의 값이 비었으면
        printf("Queue is Empty!!!");
    }
    if(tmp->data == data){ //찾는 값과 같으면 free()
        free(tmp);
        (*head) = tmp->link; //링크를 이어줌
    }else dequeue(&(tmp->link),data); //head의 주소와 찾는 값과 다르면 링크의 주소의 값과 데이터
//터를 가지고 재귀호출
}

```

```

int main(int argc, char **argv){

```

```

int in,sumf;
char buf[1024];
int i,j;
int add = 0;
int arr[10]={0};
Queue *head=NULL;

if(argc != 2){ //입력을 2개 하지 않으면 경고
    printf("2개 입력!!!\n");
    exit(-1);
}

srand(time(NULL));

for(i=0;i<10;i++){ //난수를 중복되지 않게 배열에 저장
again;;
    arr[i] = (rand()%10)+1;
    for(j=0;j<=i-1;j++){
        if(arr[i] == arr[j]) //값이 중복되는지 체크
            goto again; //만약 중복이면 again으로 돌아가 다시 난수 생성
    }
    printf("난수 : arr[%d] = %d\n",i,arr[i]); //저장된 난수 출력
}

in = open(argv[1], O_WRONLY | O_CREAT,0644); //입력한 두번째 인자를 이름으로 파일 생성

for(i=0;i<10;i++){
    sprintf(buf,"%d",arr[i]); //int형으로 선언된 배열의 숫자를 문자열로 읽어 파일에 저장하기
//위해 sprintf를 사용
    write(in,buf,strlen(buf)); //생성한 파일에 버퍼의 길이만큼 값을 저장
}

close(in);
printf("정상종료!!!\n");

for(i=0;i<10;i++){ //배열 값을 큐에 넣음

```

```

        enqueue(&head,arr[i]);
    }

    printf("\n큐 출력!!!\n");

    print_queue(head);

    /*    printf("\n지울꺼야!!!\n");

        for(i=0;i<5;i++){
            dequeue(&head,arr[i]);

            print_queue(head);

        }*/
    for(i=0;i<10;i++){ //짝수의 합
        if((arr[i]%2) == 0){
            add += arr[i];
        }
    }

    printf("\n짝수합: %d\n",add);

    sumf = open("sum.txt",O_WRONLY | O_CREAT | O_TRUNC, 0644); //파일 생성

    sprintf(buf,"%d",add); //숫자를 문자열로 읽어 파일에 적기 위해 sprintf를 사용
    write(sumf,buf,strlen(buf)); //생성한 sumf파일에 버퍼의 길이만큼 적음
    close(sumf);

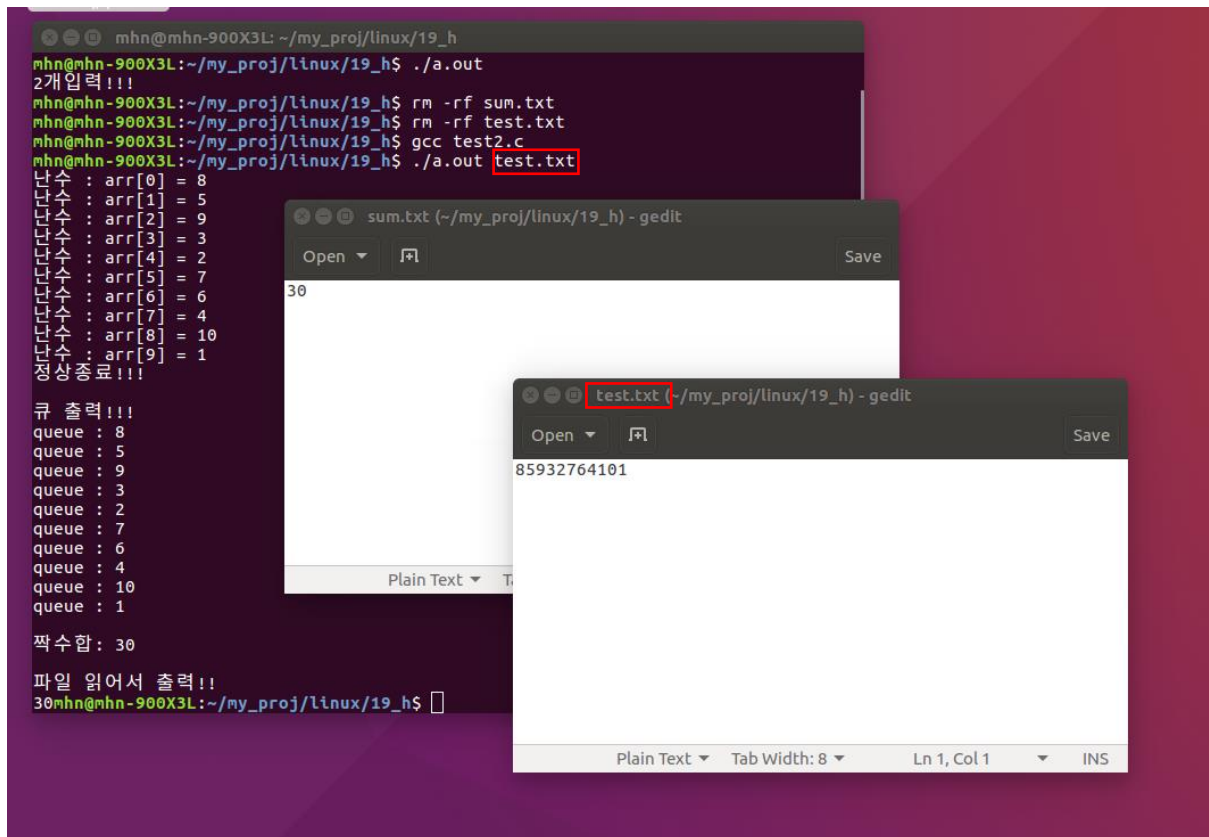
    printf("\n파일 읽어서 출력!!\n");

    read(sumf,buf,1024); //1024바이트씩 sumf를 버퍼로 읽음
    write(1,buf,sumf); //표준출력

    return 0;
}

```

<결과>



The screenshot displays a Linux environment with a terminal window and two gedit text editors. The terminal window shows the execution of a program, the removal of old files, the compilation of a new program, and the execution of the new program. The two gedit windows show the contents of the files created or modified during the process.

```
mhn@mhn-900X3L: ~/my_proj/linux/19_h
mhn@mhn-900X3L:~/my_proj/linux/19_h$ ./a.out
2개 입력!!!
mhn@mhn-900X3L:~/my_proj/linux/19_h$ rm -rf sum.txt
mhn@mhn-900X3L:~/my_proj/linux/19_h$ rm -rf test.txt
mhn@mhn-900X3L:~/my_proj/linux/19_h$ gcc test2.c
mhn@mhn-900X3L:~/my_proj/linux/19_h$ ./a.out test.txt
난수 : arr[0] = 8
난수 : arr[1] = 5
난수 : arr[2] = 9
난수 : arr[3] = 3
난수 : arr[4] = 2
난수 : arr[5] = 7
난수 : arr[6] = 6
난수 : arr[7] = 4
난수 : arr[8] = 10
난수 : arr[9] = 1
정상종료!!!

큐 출력!!!
queue : 8
queue : 5
queue : 9
queue : 3
queue : 2
queue : 7
queue : 6
queue : 4
queue : 10
queue : 1

짜수합: 30

파일 읽어서 출력!!
30mhn@mhn-900X3L:~/my_proj/linux/19_h$
```

The first gedit window, titled 'sum.txt (~/.my_proj/linux/19_h) - gedit', contains the number 30.

The second gedit window, titled 'test.txt (~/.my_proj/linux/19_h) - gedit', contains the number 85932764101.