

지나가면 또 까먹고 지나가면 또 까먹어서
오랜만에 첫번째시험 이후 나간 진도의 내용을 보니
처음본 것처럼 설레서 다시 여기서부터 복습합니다 ^^

Slap Allocator

- ㄱ. 30byte 크기의 구조체를 메모리부터 할당받아야함
- ㄴ. 페이지프레임은 4kb 인데 30byte 넣기위해 할당 받으면 낭비가 커짐.
- ㄷ. 할당,해제를 많이 반복해야 해서 느려짐.

->> 해결방안으로 Slap Allocator 가 있다.

Dup

- ㄱ. 파일 디스크립터 복사본을 만듦.
- ㄴ. 원본 디스크립터와 복사된 디스크립터의 읽기/쓰기 포인터는 공유.
- ㄷ. 즉 따로 존재하지않는다.
- ㄹ. 실패되면 -1 반환.

형태. int dup(int fildes);

lseek

SEEK_SET 파일의 시작
SEEK_CUR 현재 읽기/쓰기 포인터위치
SEEK_END 파일의 끝

예제.

```
#include <stdio.h>           // puts()
#include <string.h>           // strlen() memset()
#include <fcntl.h>             // O_WRONLY
#include <unistd.h>           // write(), close()

#define BUFF_SIZE    1024

int main()
{
    int    fd;
    char   buff[BUFF_SIZE];
    off_t  sz_file;

    fd = open( "./test.txt", O_RDONLY);

    memset( buff, '\0', BUFF_SIZE);
    lseek( fd, 10, SEEK_SET);           // 파일의 시작에서 10 번을 건너 뛴다.
    read( fd, buff, BUFF_SIZE);
    puts( buff);
```

```

memset( buff, '\0', BUFF_SIZE);
lseek( fd,  5, SEEK_SET);           // 파일의 시작에서 5 번을 건너 뛴다.
lseek( fd,  5, SEEK_CUR);          // 현재 위치에서 다시 5 번을 건너 뛴다.
read( fd, buff, BUFF_SIZE);
puts( buff);

memset( buff, '\0', BUFF_SIZE);
lseek( fd, -5, SEEK_END);           // 파일의 끝에서 앞으로 5 번을 건너 뛴다.
read( fd, buff, BUFF_SIZE);
puts( buff);

sz_file = lseek( fd,  0, SEEK_END); // 파일의 끝 위치를 반환
printf( "file size = %d\n", (int)sz_file);

close( fd);

return 0;
}

```

오늘 늦게와서 여기까지 복습하고
오늘 수업은 못들었지만 수업내용 친구들한테 받아서 쪽 봤습니다.

*****ip 주소를 알면 도메인을 알수있는 소스코드*****

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<arpa/inet.h>
#include<netdb.h>

typedef struct sockaddr_in  si;

void err_handler(char *msg)
{
    fputs(msg, stderr);
    fputc('\n', stderr);
    exit(1);
}

int main(int argc, char **argv)
{
    int i ;
    si addr ;
    struct hostent *host;

    if(argc != 2)
    {
        printf("use: %s <port>\n", argv[0]);
        exit(1);
    }
}

```

```

}
memset(&addr, 0, sizeof(addr));
addr.sin_addr.s_addr = inet_addr(argv[1]);
host = gethostbyaddr((char *)&addr.sin_addr, 4, AF_INET);

if(!host)
    err_handler("gethost error!");

printf("Official Name : %s\n", host->h_name);
for(i = 0; host->h_aliases[i]; i++)
    printf("aliases %d:%s \n", i+1, host->h_aliases[i]);

printf("Address Type: %s\n", (host->h_addrtype == AF_INET) ? "AF_INET" :
"AF_INET6");

for(i = 0 ; host->h_addr_list[i]; i++)
    printf("IP Addr %d :%s\n", i+1,
        inet_ntoa(*(struct in_addr *)host->h_addr_list[i]));
return 0;
}

```