

시스템 SW 실습 2 Programming Assignment #0

2016310936 우승민

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

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

char buffer[1];

typedef struct word_list{
    char *a;
    int count;
    struct word_list *next;
} word;
```

과제를 structure 구조로 만들었습니다. Structure의 구성은 입력 받을 단어, 들어온 단어의 개수를 세는 count 다음 structure와 연결하는 next주소를 입력하였습니다.

처음 Input text 파일을 받을 때는 main 함수 while문에서 buffer array를 통해 한 글자씩 read하며 x에 입력 받게 하였습니다. 첫 글자는 영어로 시작하여야 하므로 영어 대문자와 소문자 범위의 단어를 받게 하였고, 그 뒤 글자는 -과 `를 포함하여 받게 하였습니다. 마지막 글자 또한 영어 단어로 끝나야 하므로 -과 `를 전부 NULL값 처리하였고 그 이후 structure에 단어를 받게 하였습니다.

```
int main()
{
    int c;
    word* first=NULL;
    char *x = (char *)malloc(sizeof(char)*30);
    int i=0;
    int j=0;

    while((c = read(0,buffer,1))!= 0){
        if(i==0){
            if(((buffer[0]<='z')&&(buffer[0]>='a'))||((buffer[0]<='Z')&&(buffer[0]>='A'))){
                x[i] = buffer[0];
                i++;
            }
        }
        else{
            if(((buffer[0]<='z')&&(buffer[0]>='a'))||((buffer[0]<='Z')&&(buffer[0]>='A'))||(buffer[0]==39)||(buffer[0]=='-')){
                x[i] = buffer[0];
                i++;
            }
            else{
                while((x[i-1]==39)||(x[i-1]=='-')){
                    x[i-1]='\0';
                    i--;
                }
                first = insert(first,x,i);
                for(j=0; j<=i; j++)
                    x[j]='\0';

                i=0;
            }
        }
    }
    if(x[0]>50)
        first = insert(first,x,i);

    print_list(first);

    return 0;
}
```

While문 아래 if문은 마지막 단어를 while문에서 받지 않아 추가적으로 입력하였습니다.

```
int compare(char *x1, char *y1, int k){
    if((x1[k]<=90)&&(y1[k]>90))
    {
        if(x1[k] +32 > y1[k])
            return 1;
        else
            return -1;
    }

    if((x1[k]>90)&&(y1[k]<=90))
    {
        if(x1[k] >= y1[k]+32)
            return 1;
        else
            return -1;
    }

    if((x1[k]==0)&&(y1[k]!=0))
        return -1;
    if((x1[k]!=0)&&(y1[k]==0))
        return 1;
    if((x1[k]==0)&&(y1[k]==0))
        return 0;
    if(x1[k] > y1[k])
        return 1;
    else if(x1[k] < y1[k])
        return -1;
    else{
        k++;
        compare(x1,y1,k);
    }
}
```

compare함수를 통해 받는 단어들을 정렬하는 기준을 잡았습니다.

첫 글자가 대문자면 32만큼 더해 소문자로 만들고 숫자를 비교하고 재귀함수 형태로 알파벳위치를 하나씩 늘려 계속 비교하게 하였습니다.

왼쪽이 크면 1 오른쪽이 크면 -1 같으면 0을 return하게 만들었습니다.

여기서 받은 return값에 따라 insert함수에서 단어의 위치에 맞게 정렬하며 linked list를 만들게 하였습니다.

마지막으로 print함수는 tmp라는 char array를 만들어서 sprintf를 통해 단어의 숫자를 표시한 count를 저장하였고 strncat함수를 통해 단어와 tmp를 합쳐 한번에 출력하게 만들었습니다.

```
void print_list(word *first){
    while(first != NULL)
    {
        char tmp[4];
        sprintf(tmp,"%d",first->count);
        strncat(first->a, " ",1);
        strncat(first->a, tmp, sizeof(tmp));
        write(1,first->a, strlen(first->a));
        write(1,"\n",1);
        first = first->next;
    }
}
```

```
C:\Users\wsm42\Downloads\시실2>bash
saumsung@DESKTOP-IC1E19F:/mnt/c/Users/wsm42/Downloads/시실2$ cd pa0
saumsung@DESKTOP-IC1E19F:/mnt/c/Users/wsm42/Downloads/시실2/pa0$ make clean
rm -f main.o parser
saumsung@DESKTOP-IC1E19F:/mnt/c/Users/wsm42/Downloads/시실2/pa0$ make
gcc -g -W -Wall -c main.c -o main.o
main.c: In function 'insert':
main.c:43:6: warning: implicit declaration of function 'compare' [-Wimplicit-function-declaration]
    if(compare(cur->a, temp->a,0)==-1){
       ^~~~~~
main.c: In function 'compare':
main.c:123:1: warning: control reaches end of non-void function [-Wreturn-type]
}
gcc main.o -o parser
saumsung@DESKTOP-IC1E19F:/mnt/c/Users/wsm42/Downloads/시실2/pa0$ ./parser <ROW.txt
a 1
boat 1
but 1
down 1
dream 1
Gently 1
is 1
Life 1
Merrily 1
mer'ily 1
mer-ily 1
merrily 1
Row 1
row 2
stream 1
the 1
your 1
```

```
saumsung@DESKTOP-IC1E19F:/mnt/c/Users/wsm42/Downloads/시실2/pa0$ ./parser <EMMA.txt
A 125
Abbey 24
Abbey-Mill 7
Abbots 1
Abdy's 1
About 3
Absence 1
Absolute 1
Absolutely 1
Absurd 1
According 1
Accordingly 1
Acquit 1
Actually 1
Adair 1
Adelaide 2
Adopt 1
After 21
Agreed 1
Agricultural 1
Ah 46
Aladdin's 1
Alas 3
Alderneys 1
All 10
Almost 1
Although 1
Altogether 1
Always 2
Am 3
Ambition 1
```