

Spring 2019 Operating Systems

Lab 01: Basic C Training

0. 과제 수행 전 유의사항

프로그래밍은 자기 손으로 작성해야 한다. 아이디어를 서로 공유하고 도움을 받는 것은 좋으나, 다른 사람의 코드를 그대로 가져다 쓰는 것은 자신에게도 손해이고 다른 사람에게 피해를 끼치는 결과를 낳게되므로 허용하지 않는다. 과제 체크 시 두 사람의 코드가 유사하면, 두 사람 모두에게 과제 점수를 부여할 수 없으니 유의바란다.

1. 실습의 목표

C언어 프로그래밍에 있어 기본이 되는 콘솔출력, 파일 입출력, 배열에 관해 알아본다.

2. 작성해야할 파일

lab01_numbers.dat, lab01_histogram.c

[lab01_numbers.dat] 파일을 vi에디터로 생성하고 아래 내용을 입력한 후 파일을 저장한다. 본 과제 파일은 같은 폴더 안에 위치해 있는 것이 좋다. 따라서 mkdir로 폴더를 하나 만들고 그 안에 저장한다.

1	2	3	4	5	6	7	8	9	10
2	12	13	13	15	16	17	18	17	7
3	4	6	8	10	2	4	6	8	10
4	3	5	7	1	3	7	7	11	13
5	10	11	12	13	16	18	11	12	7
6	1	2	2	3	3	3	4	4	4
7	7	8	7	6	5	4	1	2	2
8	11	11	13	13	13	17	17	7	7
13	17	17	15	15					

[lab01_histogram.c]

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

#define MAX_ELMNTS 100
#define ANLYS_RNG 20

int getData(int numbers[], int size, int range);
void printData(int numbers[], int size, int lineSize);
void makeFrequency(int numbers[], int size, int frequency[], int range);
void makeHistogram(int frequency[], int range);

int main(void)
{
    int size;
    int nums[MAX_ELMNTS];
    int frequency[ANLYS_RNG];

    size = getData(nums, MAX_ELMNTS, ANLYS_RNG);
    printData(nums, size, 10);
    makeFrequency(nums, size, frequency, ANLYS_RNG);
    makeHistogram(frequency, ANLYS_RNG);

    return 0;
}

int getData(int data[], int size, int range)
```

```

{
    int dataIn;
    int loader = 0;
    FILE *fpData;

    range--;
    if (!(fpData = fopen("./lab01_numbers.dat", "r"))) {
        printf("Error opening files \a\a\n");
        exit(100);
    }
    while (loader < size && fscanf(fpData, "%d", &dataIn) != EOF) {
        if (dataIn >= 0 && dataIn <= range) {
            data[loader++] = dataIn;
        } else {
            printf("\nData point %d invalid. Ignored. \n", dataIn);
        }
    }
    if (loader == size) {
        printf("\nToo much data. Process what read.\n");
    }
    return loader;
}

void printData(int data[], int size, int lineSize)
{
    int numPrinted = 0;

    printf("\n\n");
    for (int i = 0; i < size; i++) {
        numPrinted++;
        printf("%2d ", data[i]);
        if (numPrinted >= lineSize) {
            printf("\n");
            numPrinted = 0;
        }
    }
    printf("\n\n");
    return ;
}

void makeFrequency(int nums[], int last, int frequency[], int range)
{
    for (int f = 0; f < range; f++)
        frequency[f] = 0;
    for (int i = 0; i < last; i++)
        frequency[nums[i]]++;

    return;
}

void makeHistogram(int freq[], int range)
{
    for (int i = 0; i < range; i++) {
        printf("%2d %2d ", i, freq[i]);
        for (int j = 1; j <= freq[i]; j++)
            printf("*");
        printf("\n");
    }
    return;
}

```

3. 컴파일

첫 번째 과제는 gcc -o lab01_histogram lab01_histogram.c 로 컴파일 할 수 있다. 직접 컴파일 해본다.

4. 컴파일된 파일 실행

실행은 ./lab01_histogram 으로 수행할 수 있다.

5. 출력내용 확인

다음과 같은 출력을 나타내면 성공이다.

```
kwp@A1409-OSLAB-01:~/os_lab/lab1$ ./lab01_histogram

1 2 3 4 5 6 7 8 9 10
2 12 13 13 15 16 17 18 17 7
3 4 6 8 10 2 4 6 8 10
4 3 5 7 1 3 7 7 11 13
5 10 11 12 13 16 18 11 12 7
6 1 2 2 3 3 3 4 4 4
7 7 8 7 6 5 4 1 2 2
8 11 11 13 13 13 17 17 7 7
13 17 17 15 15

0 0
1 4 ****
2 7 *****
3 7 *****
4 8 *****
5 4 ****
6 5 *****
7 11 *****
8 5 *****
9 1 *
10 4 ****
11 5 *****
12 3 ***
13 8 *****
14 0
15 3 ***
16 2 **
17 6 *****
18 2 **
19 0
```

6. 제출

위 파일을 코딩하고 테스트 한 다음, 코드에 주석을 붙여서 제출한다.