

영상 처리 5가지 수행하기 (헤더읽기, 데이터읽기, 영상헤더 쓰기, 영상 데이터 쓰기)

(1) 영상 헤더 읽기

```
#include <stdio.h>
int main()
{
    char M, N;                //문자 변수 M과 N 선언
    int XX, YY, MAX;          //정수 변수 XX, YY, MAX 선언
    FILE *file1;              //파일 변수 file1 선언
    file1 = fopen("five.pgm", "r"); //five.pgm을 읽기모드로 옴
    fscanf(file1, "%c", &M);    //five.pgm에서 PGM 아스키코드 포맷의 P와
                                //2를 읽어 들임
    fscanf(file1, "%c", &N);
    fscanf(file1, "%d", &XX);    //입력영상의 가로크기를 읽어 들임
    fscanf(file1, "%d", &YY);    //입력영상의 세로크기를 읽어 들임
    fscanf(file1, "%d", &MAX);    //입력영상의 최대 명암도를 읽어 들임
    //제대로 읽었는지 화면에 출력
    printf("%c", M);           //PGM 아스키코드포맷의 'P' 출력
    printf("%c\n", N);         //PGM 아스키코드포맷의 '2' 출력
    printf("%d ", XX);         //가로축의 크기 5 출력
    printf("%d\n", YY);        //세로축의 크기 7 출력
    printf("%d\n", MAX);       //최대 명암도 255 출력
    return 0;
}
```

 Computer Engineering - Prof. Jung, Minchul

```
[kimhyungho@raspberr y4 ~]nano hw1.c
[kimhyungho@raspberr y4 ~]gcc hw1.c
[kimhyungho@raspberr y4 ~]nano five.pgm
[kimhyungho@raspberr y4 ~]a.out
P2
5 7
255
[kimhyungho@raspberr y4 ~]
```

gcc를 통해서 hw1.c를 컴파일하고 nano를 사용하여 five.pgm파일을 작성합니다. a.out을 사용하여 fscanf로 five.pgm의 헤더 부분을 읽어 들이고 printf를 사용하여 헤더부분을 잘 읽었는지 화면에 출력해봅니다. five.pgm은 pgm 아스키코드 포맷을 사용했기 때문에 P2가 출력되었고 5×7 크기의 영상이기 때문에 5와 7이 출력되었습니다. 그리고 최대 명암도가 255이기 때문에 255가 잘 출력된 것을 확인할 수 있었습니다.

(2) 영상 데이터 읽기(@로 화면에 표시)

```
#include <stdio.h>
int main()
{
    int image[600][800];           //800x600 영상까지 처리가능
    int x, y;                      //반복문에서 사용할 정수 변수 x, y 선언
    char M, N;                    //문자 변수 M과 N선언
    int XX, YY, MAX;              //정수 변수 XX, YY, MAX 선언
    FILE *file1;                  //파일 변수 file1 선언
    file1 = fopen("five.pgm", "r"); //five.pgm을 읽기모드로 옴
    fscanf(file1, "%c", &M);       //five.pgm에서 PGM 아스키코드 포맷의 P와
    fscanf(file1, "%c", &N);       //2를 읽어 들임
    fscanf(file1, "%d", &XX);      //입력영상의 가로크기를 읽어 들임
    fscanf(file1, "%d", &YY);      //입력영상의 세로크기를 읽어 들임
    fscanf(file1, "%d", &MAX);     //입력영상의 최대 명암도를 읽어 들임

    for(y = 0; y < YY; y++)        //반복문을 이용하여, 0과 255로 나타낸
        for(x = 0; x < XX; x++)    //나타낸 학번의 끝자리 5를 읽어 들임
            fscanf(file1, "%d", &image[y][x]);
    //읽은 영상의 데이터를 '@'와 ' '을 통해서 출력함
    for(y=0;y<YY;y++) {
        for(x=0;x<XX;x++) {
            if(image[y][x]==255)    //255(흰색)으로 나타낸 학번의 끝자리를 부분을
                printf("@");       //특수문자 '@'를 통해서 출력
            else                    //0(검은색)으로 나타낸 배경부분을
                printf(" ");        //공백으로 출력
        }
        printf("\n");
    }

    return 0;
}
```



Computer Engineering - Prof. Jung, Minchul

```
[kimhyungho@raspberr y4 ~]nano hw2.c
[kimhyungho@raspberr y4 ~]gcc hw2.c
[kimhyungho@raspberr y4 ~]a.out
@@@@@
@
@@@@@
      @
      @
@      @
      @@@
[kimhyungho@raspberr y4 ~]
```

gcc를 사용하여 hw2.c를 컴파일 하고 a.out을 통해 실행했습니다. hw2.c의 내용에 fscanf로 five.pgm의 헤더 부분과 반복문과 fscanf를 통하여 영상의 데이터 포맷 부분을 읽어 들였습니다. hw2.c에서는 읽어 들인 헤더 부분은 printf를 통하여 확인하지 않고 데이터 포맷 부분만 printf로 숫자부분은 '@'로 배경부분은 공백을 통해서 출력하여 확인했습니다.

(4) 영상 헤더 쓰기

```
#include <stdio.h>
int main()
{
    int image[600][800];           //800x600 영상까지 처리가능
    int x, y;                      //반복문에서 사용할 정수 변수 x, y 선언
    char M, N;                    //문자 변수 M과 N선언
    int XX, YY, MAX;              //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2;          //파일 변수 file1과 file2 선언
    file1 = fopen("five.pgm", "r"); //five.pgm을 읽기모드로 옴
    fscanf(file1, "%c", &M);       //five.pgm에서 PGM 아스키코드 포맷의 P와
    fscanf(file1, "%c", &N);       //2를 읽어 들임
    fscanf(file1, "%d", &XX);      //입력영상의 가로크기를 읽어 들임
    fscanf(file1, "%d", &YY);      //입력영상의 세로크기를 읽어 들임
    fscanf(file1, "%d", &MAX);     //입력영상의 최대 명암도를 읽어 들임

    for(y = 0; y < YY; y++)       //반복문을 이용하여, 0과 255로 나타낸
        for(x = 0; x < XX; x++)   //나타낸 화번의 끝자리 5를 읽어 들임
            fscanf(file1, "%d", &image[y][x]);

    file2 = fopen("out.pgm", "w"); //out.pgm을 쓰기 모드로 옴
    fprintf(file2, "%c", M);       //읽어 들인 PGM 아스키코드 포맷의 'P'와
    fprintf(file2, "%cWn", N);     //'2'를 out.pgm에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX);   //읽어 들인 최대 명암도를 입력

    return 0;
}
```

 Computer Engineering - Prof. Jung, Minchul

```
[kimhyungho@raspberrry4 ~]gcc hw4.c
[kimhyungho@raspberrry4 ~]a.out
[kimhyungho@raspberrry4 ~]ls -l out.pgm
-rw----- 1 kimhyungho class2020 11  4월 16 23:43 out.pgm
[kimhyungho@raspberrry4 ~]cat out.pgm
P2
5 7
255
[kimhyungho@raspberrry4 ~]
```

gcc를 사용하여 hw4.c를 컴파일하고 a.out을 사용하여 실행합니다. ls -l out.pgm 명령어를 사용하여 쓰기모드를 이용하여 생성한 out.pgm의 생성여부를 확인할 수 있습니다. hw4.c에서는 fscanf로 영상의 헤더 부분과 데이터 포맷 부분을 읽어 들였습니다. 그리고 out.pgm을 쓰기모드로 열고 그 안에 fprintf를 사용하여 영상의 헤더 부분을 입력했습니다. cat out.pgm을 사용하여 out.pgm에 저장한 영상의 헤더 부분을 확인해 볼 수 있었습니다.

(5) 영상 데이터 쓰기

```
#include <stdio.h>
int main()
{
    int image[600][800];           //800x600 영상까지 처리가능
    int x, y;                      //반복문에서 사용할 정수 변수 x, y 선언
    char M, N;                    //문자 변수 M과 N선언
    int XX, YY, MAX;              //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2;          //파일 변수 file1과 file2 선언
    file1 = fopen("five.pgm", "r"); //five.pgm을 읽기모드로 옴
    fscanf(file1, "%c", &M);       //five.pgm에서 PGM 아스키코드 포맷의 P와
    fscanf(file1, "%c", &N);       //2를 읽어 들임
    fscanf(file1, "%d", &XX);      //입력영상의 가로크기를 읽어 들임
    fscanf(file1, "%d", &YY);      //입력영상의 세로크기를 읽어 들임
    fscanf(file1, "%d", &MAX);     //입력영상의 최대 명암도를 읽어 들임

    for(y = 0; y < YY; y++)        //반복문을 이용하여, 0과 255로 나타낸
        for(x = 0; x < XX; x++)    //학번의 끝자리 5를 읽어 들임
            fscanf(file1, "%d", &image[y][x]);

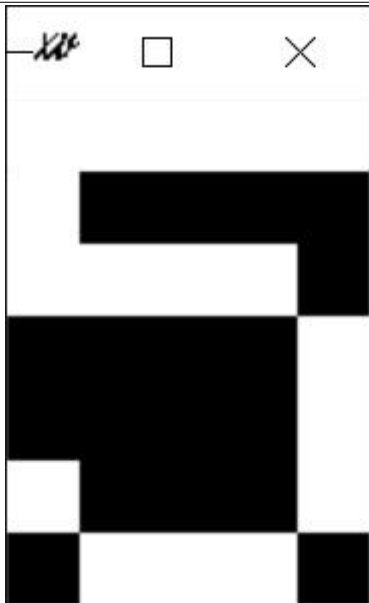
    //디지털 영상 처리 알고리즘

    file2 = fopen("out.pgm", "w"); //out.pgm을 쓰기 모드로 옴
    fprintf(file2, "%c", M);       //읽어 들인 PGM 아스키코드 포맷의 'P'와
    fprintf(file2, "%cWn", N);     //'2'를 out.pgm에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX);   //읽어 들인 최대 명암도를 입력

    for(y = 0; y < YY; y++){
        for(x = 0; x < XX; x++){    //반복문을 이용하여 0과 255로 나타낸
            fprintf(file2, "%4d", image[y][x]); //학번의 끝자리 5를 out.pgm에 입력
        }
        fprintf(file2, "Wn");
    }

    return 0;
}
```

```
[kimhyungho@raspberrypi4 ~]gcc hw5.c
[kimhyungho@raspberrypi4 ~]a.out
[kimhyungho@raspberrypi4 ~]ls -l out.pgm
-rw----- 1 kimhyungho class2020 158  4월 16 23:45 out.pgm
[kimhyungho@raspberrypi4 ~]cat out.pgm
P2
5 7
255
255 255 255 255 255
255 0 0 0 0
255 255 255 255 0
0 0 0 0 255
0 0 0 0 255
255 0 0 0 255
0 255 255 255 0
[kimhyungho@raspberrypi4 ~]xv out.pgm
```



gcc를 사용하여 hw5.c를 컴파일하고 a.out을 사용하여 실행합니다. ls -l out.pgm 명령어를 사용하여 쓰기모드를 이용하여 생성한 out.pgm의 생성여부를 확인할 수 있습니다. hw5.c에서도 fscnaf로 영상의 헤더 부분과 데이터 포맷 부분을 읽어 들였습니다. 그리고 out.pgm을 쓰기모드로 열고 그 안에 fprintf를 사용하여 영상의 헤더 부분과 데이터 포맷 부분을 모두 입력했습니다. cat out.pgm을 사용하여 out.pgm에 입력된 영상의 헤더부분과 데이터포맷부분을 확인하고 헤더와 데이터포맷부분이 있으니 영상을 xv로 출력해서 확인 할 수도 있습니다. xv를 사용하여 영상을 출력한 결과 검은색 배경과 흰색으로 숫자 5를 출력한 것을 확인할 수 있습니다.

(3) 영상 (보수) 처리

```
#include <stdio.h>
int main()
{
    int image[600][800];           //800x600 영상까지 처리가능
    int x, y;                      //반복문에서 사용할 정수 변수 x, y 선언
    char M, N;                    //문자 변수 M과 N선언
    int XX, YY, MAX;              //정수 변수 XX, YY, MAX 선언
    FILE *file1, *file2;          //파일 변수 file1과 file2 선언
    file1 = fopen("five.pgm", "r"); //five.pgm을 읽기모드로 옴
    fscanf(file1, "%c", &M);       //five.pgm에서 PGM 아스키코드 포맷의 P와
    fscanf(file1, "%c", &N);       //2를 읽어 들임
    fscanf(file1, "%d", &XX);      //입력영상의 가로크기를 읽어 들임
    fscanf(file1, "%d", &YY);      //입력영상의 세로크기를 읽어 들임
    fscanf(file1, "%d", &MAX);     //입력영상의 최대 명암도를 읽어 들임


    for(y = 0; y < YY; y++)        //반복문을 이용하여, 0과 255로 나타낸
        for(x = 0; x < XX; x++)    //나타낸 화번의 끝자리 5를 읽어 들임
            fscanf(file1, "%d", &image[y][x]);

    //디지털 영상 처리 알고리즘
    for(y = 0; y < YY; y++)        //반복문을 이용하여, 255 - 명암도로
        for(x = 0; x < XX; x++)    //영상의 명암을 반전시킵니다.
            image[y][x] = 255 - image[y][x];

    file2 = fopen("out.pgm", "w"); //out.pgm을 쓰기 모드로 옴
    fprintf(file2, "%c", M);       //읽어 들인 PGM 아스키코드 포맷의 'P'와
    fprintf(file2, "%cWn", N);     //'2'를 out.pgm에 입력
    fprintf(file2, "%d %dWn", XX, YY); //읽어 들인 가로축과 세로축 크기를 입력
    fprintf(file2, "%dWn", MAX);   //읽어 들인 최대 명암도를 입력

    for(y = 0; y < YY; y++){
        for(x = 0; x < XX; x++){    //반복문을 이용하여 명암이 반전 된
            fprintf(file2, "%4d", image[y][x]); //화번의 끝자리를 영상을 out.pgm에 입력
        }
        fprintf(file2, "Wn");
    }

    return 0;
}
```

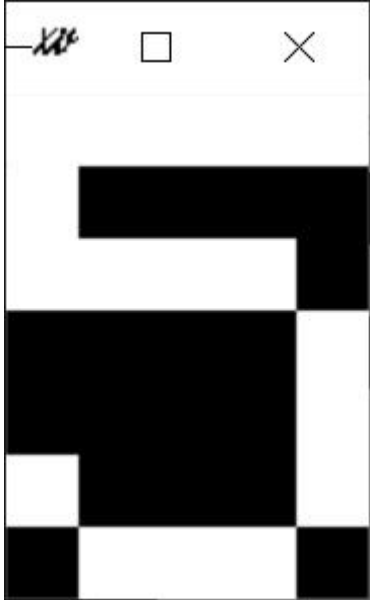

 Computer Engineering - Prof. Jung, Minchul

```

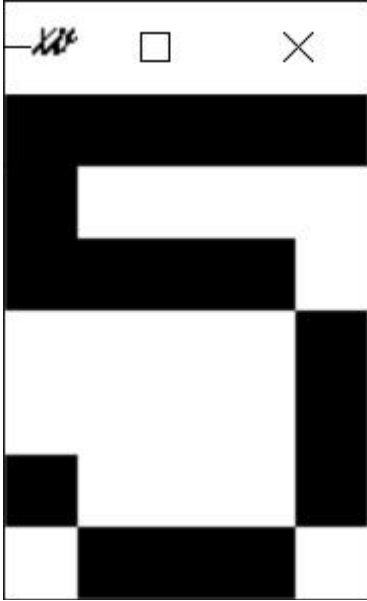
[kimhyungho@raspberrypi4 ~]gcc hw3.c
[kimhyungho@raspberrypi4 ~]a.out
[kimhyungho@raspberrypi4 ~]ls -l out.pgm
-rw----- 1 kimhyungho class2020 158  4월 17 00:37 out.pgm
[kimhyungho@raspberrypi4 ~]xv five.pgm &
[1] 13305
[kimhyungho@raspberrypi4 ~]xv out.pgm &
[2] 13306
[kimhyungho@raspberrypi4 ~]

```

입력영상



출력 영상



gcc를 사용하여 hw3.c를 컴파일하고 a.out을 사용하여 실행합니다. ls -l out.pgm 명령어를 사용하여 쓰기모드를 이용하여 생성한 out.pgm의 생성여부를 확인할 수 있습니다. hw3.c에서도 fscanf로 영상의 헤더 부분과 데이터 포맷 부분을 읽어 들였습니다. 반복문과 $image[y][x] = 255 - image[y][x]$ 를 사용하여 명암을 반전시켰습니다. (255인 부분은 0이되고 0이었던 부분은 255가 됩니다.) 그리고 out.pgm을 쓰기모드로 열고 그 안에 fprintf를 사용하여 영상의 헤더 부분과 명암이 반전된 데이터 포맷 부분을 입력했습니다. 명암이 반전된 영상을 xv로 출력해서 확인 할 수 있습니다. 입력 영상인 five.pgm을 xv로 출력한 결과 하얀색 숫자에 검은색 배경임을 확인할 수 있고 명암을 반전시킨 출력 영상 out.pgm은 검은색 숫자에 흰색 배경임을 확인할 수 있습니다.

워드파일(hwp/doc)로 작성하는 보고서 파일에는 각 5개 과정의

①주석 처리된 소스(화면 캡처금지, 텍스트로 작성), ②입력(실행), ③출력(실행)화면이 반드시 있어야 하며 과제는 상대 평가 됩니다. (각 문제당 10점, 50점 만점).