

소프트웨어 공학 개론

T3

Fail Result of system testing by T2

Test Case Number	Description
T2_01	대기 열에 <u>userID</u> 미 표시
T2_02	사용자 등록 시 admin 아이디가 붙어서 저장
T2_03	사용자 등록 시 길이가 긴 문자열 입력 시 프로그램 종료
T2_04	프린트 중단 시 현재까지 출력된 파일이 출력되지 않음
T2_05	1분 안에 여러 번 출력 요청 시 파일이 덮어쓰워 생성됨
T2_06	파일 한 줄에 40자 입력 후 출력 요청 시 프로그램 종료 파일이 11줄 일 때 2장으로 인식하는가 - 14줄까지 한 장으로 인식 290글자 출력 시 잉크가 두 번 소모

T2_01

SRS에 나와 있는 LCD를 보면 예시라고 되어있어서
대기중인 ID를 LCD에 띄우는 것이 아닌
대기 중인 사람의 수를 띄우는 방향으로 개발

=> UserManual에 명시

T2_02

```
void user_join(char * ID)
{
    FILE *fp = fopen("../userlist.txt", "a");

    fprintf(fp, "%s", ID);

    fclose(fp);
}
```

```
void user_join(char * ID)
{
    FILE *fp = fopen("../userlist.txt", "a");

    fprintf(fp, "%s\n", ID);

    fclose(fp);
}
```

T2_03

ID의 길이에 제한을 두었다. ID는 최대 10글자가 가능하다.

=> **UserManual에 명시**

T2_04

```
while(fgets(one_line, 499, original_file) != NULL)
{
    PS.print_current_state = 1; // 인쇄 중으로 상태 변경
    if(read_update()) // 새로운 값 입력 들어옴
    {
        int now_command;
        now_command = read_commmand();

        switch(now_command)
        {
            case 1 :
                printcontroller(now_command);
                break;
            case 2 :
                return;
            case 3 :
            case 4 :
                refillcontroller(now_command);
                break;
            case 5 :
            case 6 :
            case 7 :
                manageusercontroller(now_command);
                break;
        }
    }
}
```

```
while(fgets(one_line, 499, original_file) != NULL)
{
    PS.print_current_state = 1; // 인쇄 중으로 상태 변경
    if(read_update()) // 새로운 값 입력 들어옴
    {
        int now_command;
        now_command = read_commmand();

        switch(now_command)
        {
            case 1 :
                printcontroller(now_command);
                break;
            case 2 :
                fclose(new_file);
                return;
            case 3 :
            case 4 :
                refillcontroller(now_command);
                break;
            case 5 :
            case 6 :
            case 7 :
                manageusercontroller(now_command);
                break;
        }
    }
}
```

T2_05

```
char buf_year[5];
char buf_mon[3];
char buf_mday[3];
char buf_hour[3];
char buf_min[3];
char new_file_name[13];

original_file = fopen(WT.file, "r");

timer = time(NULL);

t = localtime(&timer);

n=sprintf (buf_year, "%d",t->tm_year+1900);
n=sprintf (buf_mon, "%d",t->tm_mon+1);
n=sprintf (buf_mday, "%d",t->tm_mday);
n=sprintf (buf_hour, "%d",t->tm_hour);
n=sprintf (buf_min, "%d",t->tm_min);

sprintf(new_file_name,"../%d%d%d%d.txt", t->tm_year+1900, t->tm_mon+1, t->tm_mday, t->tm_hour, t->tm_min);
```

```
char buf_year[5];
char buf_mon[3];
char buf_mday[3];
char buf_hour[3];
char buf_min[3];
char buf_sec[3];
char new_file_name[13];

original_file = fopen(WT.file, "r");

timer = time(NULL);

t = localtime(&timer);

n=sprintf (buf_year, "%d",t->tm_year+1900);
n=sprintf (buf_mon, "%d",t->tm_mon+1);
n=sprintf (buf_mday, "%d",t->tm_mday);
n=sprintf (buf_hour, "%d",t->tm_hour);
n=sprintf (buf_min, "%d",t->tm_min);
n=sprintf (buf_sec, "%d", t->tm_sec);

sprintf(new_file_name,"../%d%d%d%d%d.txt", t->tm_year+1900, t->tm_mon+1, t->tm_mday, t->tm_hour, t->tm_min, t->tm_sec);
```

T2_06

```
while (fgets(line, 255, file) != NULL)
{
    // 한 줄을 읽어 저장(마지막에 자동으로 종료코드 삽입)
    Num_of_Line++; // 줄 수 증가
    for (i = 0; i < 30; i++) // 한 줄의 길이 만큼 반복문
    {
        ch = line[i];
        if ((ch>32) && (ch<127)) // 알파벳이나 특수문자 인가?
            Num_of_Alpha++;
        else if (ch == 32) // 공백인가?
            Num_of_Space++;
    }
}
```

```
while (fgets(line, 255, file) != NULL)
{
    // 한 줄을 읽어 저장(마지막에 자동으로 종료코드 삽입)
    Num_of_Line++; // 줄 수 증가
    for (i = 0; i < 30; i++) // 한 줄의 길이 만큼 반복문
    {
        ch = line[i];
        if ((ch>32) && (ch<127)) // 알파벳이나 특수문자 인가?
            Num_of_Alpha++;
        else if (ch == 32) // 공백인가?
            Num_of_Space++;

        if(line[i] == '\n')
        {
            break;
        }
    }
}
```


Fail Result of system testing by T3

Test Case Number	Description
T3_01	충전이 끝나고 대기하던 인쇄가 진행 될 때 대기자 수가 줄어들지 않는다.
T3_02	dslab이라고 userlist.txt에 쓰이긴 하지만 다음 줄에 쓰이지 않아서 기존의 마지막 ID 뒤에 이어서 써진다.
T3_03	이미 충전 중이라는 알람이 뜨지 않는다.
T3_04	최대치로 충전한다는 알람은 뜨나, 최대치로 충전되지는 않는다.

T3_01

```
while(fgets(read_wait_line, 49, fp) != NULL)
{
    temp_file_pointer = ftell(fp);
    fclose(fp);
    char * temp_paper = (char *)malloc(sizeof(char)*10);
    WT.ID = strtok(read_wait_line,",");
    WT.file = strtok(NULL,",");
    temp_paper = strtok(NULL,",");
    WT.paper = atoi(temp_paper);
    PS.print_current_state = 1;
```

```
printinterface(WT);
```

```
if(PS.wait_num > 0)
{
    PS.wait_num --;
}
```

```
while(fgets(read_wait_line, 49, fp) != NULL)
{
    temp_file_pointer = ftell(fp);
    fclose(fp);
    char * temp_paper = (char *)malloc(sizeof(char)*10);
    WT.ID = strtok(read_wait_line,",");
    WT.file = strtok(NULL,",");
    temp_paper = strtok(NULL,",");
    WT.paper = atoi(temp_paper);
    PS.print_current_state = 1;
```

```
if(PS.wait_num > 0)
{
    PS.wait_num --;
}
```

```
printinterface(WT);
```

T3_02

T2_02 부분과 동일

```
void user_join(char * ID)
{
    FILE *fp = fopen("../userlist.txt", "a");

    fprintf(fp, "%s", ID);

    fclose(fp);
}
```

```
void user_join(char * ID)
{
    FILE *fp = fopen("../userlist.txt", "a");

    fprintf(fp, "%s\n", ID);

    fclose(fp);
}
```

T3_03

```
break;
case 6:
// 충전하면 최대 잉크량 초과할 때
printf(" Warning !! \n");
printf(" 잉크를 충전하면 최대 잉크 량을 초과합니다. \n");
printf(" 잉크를 최대치로 충전합니다. \n");
break;
case 7:
// 유저 추가, 삭제, 보여주기 명령 했는데 휴식 중이 아닐 때
printf(" Warning !! \n");
printf(" 휴식 중이 아니므로 명령을 수행할 수 없습니다. \n");
break;
case 8:
// 등록할 사람이 이미 존재 할 때
printf(" Warning !! \n");
printf(" 등록할 사용자가 이미 존재합니다. \n");
break;
case 9:
// 삭제할 사람이 존재 하지 않을 때
printf(" Warning !! \n");
printf(" 삭제할 사용자가 존재하지 않습니다. \n");
break;
```

```
case 7:
// 유저 추가, 삭제, 보여주기 명령 했는데 휴식 중이 아닐 때
printf(" Warning !! \n");
printf(" 휴식 중이 아니므로 명령을 수행할 수 없습니다. \n");
break;
case 8:
// 등록할 사람이 이미 존재 할 때
printf(" Warning !! \n");
printf(" 등록할 사용자가 이미 존재합니다. \n");
break;
case 9:
// 삭제할 사람이 존재 하지 않을 때
printf(" Warning !! \n");
printf(" 삭제할 사용자가 존재하지 않습니다. \n");
break;
case 10:
// 충전 요청인데 이미 충전 중
printf(" Warning !! \n");
printf(" 현재 충전 중이므로 또 다른 충전을 수행할 수 없습니다. \n");
```

T3_04

```
void refill_ink(int amount)
{
    int i;
    PS.print_current_state = 2;

    for(i = 1; i<(amount+1); i++) // amount 만큼 반복
    {
        if(read_update()) // 새로운 값 입력 들어옴
        {
            int now_command;
            now_command = read_commmand();

            switch(now_command)
            {
                case 1 :
                case 2 :
                    printcontroller(now_command);
                    break;
                case 3 :
                case 4 :
                    refillcontroller(now_command);
                    break;
                case 5 :
                case 6 :
                case 7 :
                    manageusercontroller(now_command);
                    break;
            }
        }
        PS.print_current_state = 2;
        PS.temp_ink;
        PS.current_ink ++;
        if(((i%100) == 0) || (amount == i)) // 100 충전했거나, 마지막 충
        {
            lcdinterface("admin", 0, 0); // lcd 내부에서 sleep
        }
    }
    if(PS.wait_num >0)
    {
        read_wait();
    }
}
```

```
void refill_paper(int amount)
{
    int i;
    PS.print_current_state = 2;

    for(i = 1; i<(amount+1); i++) // amount 만큼 반복
    {
        if(read_update()) // 새로운 값 입력 들어옴
        {
            int now_command;
            now_command = read_commmand();

            switch(now_command)
            {
                case 1 :
                case 2 :
                    printcontroller(now_command);
                    break;
                case 3 :
                case 4 :
                    refillcontroller(now_command);
                    break;
                case 5 :
                case 6 :
                case 7 :
                    manageusercontroller(now_command);
                    break;
            }
        }
        PS.print_current_state = 2;
        PS.temp_paper;
        PS.current_paper ++;
        if(((i%10) == 0) || (amount == i)) // 10장 충전했거나, 마지막
        {
            lcdinterface("admin", 0, 0); // lcd 내부에서 sleep
        }
    }
    if(PS.wait_num >0)
    {
        read_wait();
    }
}
```

T3_04

```
void refill_ink(int amount)
{
    int i;
    PS.print_current_state = 2;
    if((amount + PS.current_ink) > MAX_INK) // 최대치로 충전
    {
        amount = MAX_INK - (PS.current_ink);
    }

    for(i = 1; i<(amount+1); i++) // amount 만큼 반복
    {
        if(read_update()) // 새로운 값 입력 들어옴
        {
            int now_command;
            now_command = read_command();

            switch(now_command)
            {
                case 1 :
                case 2 :
                    printcontroller(now_command);
                    break;
                case 3 :
                case 4 :
                    refillcontroller(now_command);
                    break;
                case 5 :
                case 6 :
                case 7 :
                    manageusercontroller(now_command);
                    break;
            }
        }
        PS.print_current_state = 2;
        PS.temp_ink ++;
        PS.current_ink ++;
        if(((i%100) == 0) || (amount == i)) // 100 충전했거나, 마지막
        {
            lcdinterface("admin", 0, 0); // lcd 내부에서 sleep
        }
    }
}
```

```
void refill_paper(int amount)
{
    int i;
    PS.print_current_state = 2;
    if((amount + PS.current_paper) > MAX_PAPER)
    {
        amount = MAX_PAPER - (PS.current_paper);
    }

    for(i = 1; i<(amount+1); i++) // amount 만큼 반복
    {
        if(read_update()) // 새로운 값 입력 들어옴
        {
            int now_command;
            now_command = read_command();

            switch(now_command)
            {
                case 1 :
                case 2 :
                    printcontroller(now_command);
                    break;
                case 3 :
                case 4 :
                    refillcontroller(now_command);
                    break;
                case 5 :
                case 6 :
                case 7 :
                    manageusercontroller(now_command);
                    break;
            }
        }
        PS.print_current_state = 2;
        PS.temp_paper++;
        PS.current_paper ++;
        if(((i%10) == 0) || (amount == i)) // 10장 충전했거나, 마지막 충전 일때
        {
            lcdinterface("admin", 0, 0); // lcd 내부에서 sleep
        }
    }
    if(PS.wait_num > 0)
    {
        read_wait();
    }
}
```

Q & A

Thank You

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the right side of the frame, creating a modern, layered effect against the white background.