



Debug Message

로봇SW 교육원

최상훈(shchoi82@gmail.com)

목차

2

- 실습예제
- 미리 정의된 매크로
- 매크로 함수
- 표준 에러 입출력
- 조건부 컴파일
- tail -f

실습 준비

3

- **실습파일 다운로드**

```
$ scp -P 50000 guest@220.70.1.98:/home/guest/dbgmsg.tar.gz .
```

- **압축풀기**

```
$ tar xvfz dbgmsg.tar.gz
```

실습1

4

파일명 : ex1.c

```
#include <stdio.h>

void change(int, int*, int**);

int gvala = 1;
int gvalb = 2;

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    change(val, p1, &p2);

    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval    = gvalb;
    tmpp1     = &gvalb;
    *tmpp2    = &gvalb;
}
```

실습1

5

```
$ gcc -Wall ex1.c -o ex1
$ ./ex1
val:1, *p1:1, *p2:2
$
```

실습2

6

파일명 : ex2.c

```
#include <stdio.h>

void change(int, int*, int**);

int gvala = 1;
int gvalb = 2;

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    printf("before p1:%p, p2:%p\n", p1, p2);      // <-- 디버깅 메세지
    change(val, p1, &p2);
    printf("after  p1:%p, p2:%p\n", p1, p2);      // <-- 디버깅 메세지
    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval    = gvalb;
    tmpp1     = &gvalb;
    *tmpp2    = &gvalb;
}
```

실습2

7

```
$ gcc -Wall ex2.c -o ex2
$ ./ex2
before p1:0x804a01c, p2:0x804a01c
after  p1:0x804a01c, p2:0x804a020
val:1, *p1:1, *p2:2
$
```

실습3

8

파일명 : ex3.c

```
#include <stdio.h>

void change(int, int*, int**);

int gvala = 1;
int gvalb = 2;

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    printf("before p1:%p, p2:%p, &p2:%p\n", p1, p2, &p2);
    change(val, p1, &p2);
    printf("after  p1:%p, p2:%p\n", p1, p2);
    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval    = gvalb;
    tmpp1     = &gvalb;
    *tmpp2    = &gvalb;
    printf("\tin change()\n");
    printf("\ttmpval:%d, tmpp1:%p, tmpp2:%p\n", tmpval, tmpp1, tmpp2);
}
```


실습3

9

```
$ ./ex3
before p1:0x804a020, p2:0x804a020, &p2:0xbff7feb4
      in change()
      tmpval:2, tmpp1:0x804a024, tmpp2:0xbff7feb4
after  p1:0x804a020, p2:0x804a024
val:1, *p1:1, *p2:2
$
```

미리 정의된 매크로

10

- 매크로(Standard Predefined Macros)

매크로	내용	타입
<code>__FILE__</code>	소스 파일명	a string constant
<code>__LINE__</code>	소스 줄 번호	a decimal integer constant
<code>__func__</code>	함수명	a string constant
<code>__DATE__</code>	전처리기 수행 날짜	a string constant
<code>__TIME__</code>	전처리기 수행 시간	a string constant

실습4

11

파일명 : ex4.c

```
#include <stdio.h>
void f1(void);
void f2(void);
void f3(void);

int main(void)
{
    printf("file name:%s\n", __FILE__);
    printf("function name:%s [%d]\n", __func__, __LINE__);
    printf("datetime:%s %s\n", __DATE__, __TIME__);
    f1();
    f2();
    f3();
    return 0;
}

void f1(void)
{
    printf("%s [%d]\n", __func__, __LINE__);
}

void f2(void)
{
    printf("%s [%d]\n", __func__, __LINE__);
}

void f3(void)
{
    printf("%s [%d]\n", __func__, __LINE__);
}
```

실습4

12

```
$ gcc -Wall ex4.c -o ex4
$ ./ex4
file name:ex4.c
function name:main [9]
datetime:Mar 30 2016 22:05:45
f1 [19]
f2 [24]
f3 [29]
$
```

실습5

13

파일명 : ex5.c

```

#include <stdio.h>
void change(int, int*, int**);

int gvala = 1;
int gvalb = 2;
#define ANSI_COLOR_RED      "\x1b[31m"
#define ANSI_COLOR_RESET    "\x1b[0m"
#define DBGMSG_PREFIX      ANSI_COLOR_RED "<< DBGMSG >> " ANSI_COLOR_RESET

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    printf(DBGMSG_PREFIX "[%d] p1:%p, p2:%p, &p2:%p\n", __LINE__, p1, p2, &p2);
    change(val, p1, &p2);
    printf(DBGMSG_PREFIX "[%d] p1:%p, p2:%p\n", __LINE__, p1, p2);
    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval = gvalb;
    tmpp1 = &gvalb;
    *tmpp2 = &gvalb;
    printf(DBGMSG_PREFIX "[%s]\n", __func__);
    printf(DBGMSG_PREFIX "tmpval:%d, tmpp1:%p, tmpp2:%p\n", tmpval, tmpp1, tmpp2);
}

```

실습5

14

```
$ ./ex5
<< DBGMSG >> [18] p1:0x804a01c, p2:0x804a01c, &p2:0xbfc521b4
<< DBGMSG >> [change]
<< DBGMSG >> tmpval:2, tmpp1:0x804a020, tmpp2:0xbfc521b4
<< DBGMSG >> [20] p1:0x804a01c, p2:0x804a020
val:1, *p1:1, *p2:2
$
```

```
#define ANSI_COLOR_GREEN    "\x1b[32m"
#define ANSI_COLOR_YELLOW   "\x1b[33m"
#define ANSI_COLOR_BLUE     "\x1b[34m"
#define ANSI_COLOR_MAGENTA  "\x1b[35m"
#define ANSI_COLOR_CYAN     "\x1b[36m"
..
```

표준 에러 출력

15

- **표준 파일 디스크립터**
 - 표준 입력, 표준 출력, 표준 에러
 - STDIN_FILENO, STDOUT_FILENO, STDERR_FILENO
 - 0, 1, 2
- **표준 파일 스트림**
 - 표준 입력, 표준 출력, 표준 에러
 - stdin, stdout, stderr
- **표준 출력 예**
 - `printf("stdout message\n");`
 - `fprintf(stdout, "stdout message\n");`
- **표준 에러 예**
 - `fprintf(stderr, "stderr message\n");`
- **표준 파일 디스크립터 I/O 리다이렉션**
 - **표준 출력 리다이렉션 예**
 - `$./a.out > filename`
 - **표준 에러 리다이렉션 예**
 - `$./a.out 2> filename`

실습6

16

```
$ ls abcd.c
ls: cannot access abcd.c: 그런 파일이나 디렉터리가 없습니다
$ ls abcd.c > filelist2
ls: cannot access abcd.c: 그런 파일이나 디렉터리가 없습니다
$ cat filelist2
$ ls -l filelist2
-rw-rw-r--. 1 shchoi82 shchoi82 0  3월 30 22:43 filelist2
$ ls abcd.c 2> filelist2
$ cat filelist2
ls: cannot access abcd.c: 그런 파일이나 디렉터리가 없습니다
$ ls -l filelist2
-rw-rw-r--. 1 shchoi82 shchoi82 75  3월 30 22:44 filelist2
$
```


실습7

17

파일명 : ex7.c

```
#include <stdio.h>
void change(int, int*, int**);

int gvala = 1;
int gvalb = 2;
#define ANSI_COLOR_RED      "\x1b[31m"
#define ANSI_COLOR_RESET    "\x1b[0m"
#define DBGMSG_PREFIX       ANSI_COLOR_RED "<< DBGMSG >> " ANSI_COLOR_RESET

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    fprintf(stderr, DBGMSG_PREFIX "[%d] p1:%p, p2:%p\n", __LINE__, p1, p2);
    change(val, p1, &p2);
    fprintf(stderr, DBGMSG_PREFIX "[%d] p1:%p, p2:%p\n", __LINE__, p1, p2);
    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval  = gvalb;
    tmpp1   = &gvalb;
    *tmpp2  = &gvalb;
    fprintf(stderr, DBGMSG_PREFIX "[%s]\n", __func__);
    fprintf(stderr, DBGMSG_PREFIX "tmpval:%d, tmpp1:%p, tmpp2:%p\n", tmpval, tmpp1, tmpp2);
}
```

실습7

18

```

$ gcc -Wall ex7.c -o ex7
$ ./ex7
<< DBGMSG >> [18] p1:0x804a020, p2:0x804a020
<< DBGMSG >> [change]
<< DBGMSG >> tmpval:2, tmpp1:0x804a024, tmpp2:0xbfad6d14
<< DBGMSG >> [20] p1:0x804a020, p2:0x804a024
val:1, *p1:1, *p2:2
$ ./ex7 > stdout_log
<< DBGMSG >> [18] p1:0x804a020, p2:0x804a020
<< DBGMSG >> [change]
<< DBGMSG >> tmpval:2, tmpp1:0x804a024, tmpp2:0xbfe83274
<< DBGMSG >> [20] p1:0x804a020, p2:0x804a024
$ ./ex7 2> stderr_log
val:1, *p1:1, *p2:2
$ cat stdout_log
val:1, *p1:1, *p2:2
$ cat stderr_log
<< DBGMSG >> [18] p1:0x804a020, p2:0x804a020
<< DBGMSG >> [change]
<< DBGMSG >> tmpval:2, tmpp1:0x804a024, tmpp2:0xbfbcb27c4
<< DBGMSG >> [20] p1:0x804a020, p2:0x804a024
$

```

매크로 함수

19

- 매크로 함수

```
#define SUM(x,y) x + y
#define SUM3(x,y,z) x + y + z
#define STRCAT_V1(x,y) x y
#define STRCAT_V2(x,y) #x #y
#define STR(x) #x
#define MERGE_V1(x,y) x ## y
#define MERGE_V2(x,y) STR(x ## y)
```

```
#include <stdio.h>
#define DBGMSG_PREFIX "<< DEBUG >> "
#define DBGMSG(msg, ...) fprintf(stderr, \
                                DBGMSG_PREFIX "[%s %s:%d] : " msg "\n", \
                                __FILE__, __func__, __LINE__, ##__VA_ARGS__)
```

실습8

20

파일명 : ex8.c

```
#include <stdio.h>

#define SUM(x,y) x + y
#define SUM3(x,y,z) x + y + z
#define STRCAT_V1(x,y) x y
#define STRCAT_V2(x,y) #x #y
#define STR(x) #x
#define MERGE_V1(x,y) x ## y
#define MERGE_V2(x,y) STR(x) ## y

Int main(void)
{
    int val1 = SUM(1,2);
    int val2 = SUM3(1,2,3);
    char *str1 = "string1" "string2";
    char *str2 = STRCAT_V1("I am ", "shchoi82.");
    char *str3 = STRCAT_V2(I am\x20, shchoi82..);
    char *str4 = STRCAT_V2(\x48, \x69);
    char *str5 = STR(I am\x20) STR(shchoi82...);
    char *str6 = MERGE_V1(str,1);
    char *str7 = MERGE_V2(str,2);

    printf("val1: %d\n", val1);
    printf("val2: %d\n", val2);
    printf("str1: %s\n", str1);
    printf("str2: %s\n", str2);
    printf("str3: %s\n", str3);
    printf("str4: %s\n", str4);
    printf("str5: %s\n", str5);
    printf("str6: %s\n", str6);
    printf("str7: %s\n", str7);
    return 0;
}
```

실습8

21

```
$ gcc -Wall --save-temps ex8.c -oex8
$ ./ex8
val1: 3
val2: 6
str1: string1string2
str2: I am shchoi82.
str3: I am shchoi82..
str4: Hi
str5: I am shchoi82...
str6: string1string2
str7: str2
$ vim ex8.i
```

```
...
int
main(void)
{
    int val1 = 1 + 2;
    int val2 = 1 + 2 + 3;
    char *str1 = "string1" "string2";
    char *str2 = "I am " "shchoi82.";
    char *str3 = "I am\x20" "shchoi82..";
    char *str4 = "\x48" "\x69";
    char *str5 = "I am\x20" "shchoi82...";
    char *str6 = str1;
    char *str7 = "str2";

    ...
}
```

실습9

22

파일명 : ex9.c

```
#include <stdio.h>

#define DBGMSG_V1 printf("%s %s %d\n", __FILE__, __func__, __LINE__)

#define DBGMSG_V2 fprintf(stderr, \
                        "%s %s %d\n", __FILE__, __func__, __LINE__)

#define DBGMSG_V3(msg) fprintf(stderr, \
                        "%s %s %d " msg "\n", __FILE__, __func__, __LINE__)

#define DBGMSG_V4(msg) fprintf(stderr, \
                        "[%s %s %d] : " msg "\n", __FILE__, __func__, __LINE__)

#define DBGMSG_V5(msg, arg1) fprintf(stderr, \
                        "[%s %s %d] : " msg "\n", __FILE__, __func__, __LINE__, arg1)

#define REDS      "\x1b[31m"
#define REDE      "\x1b[0m"
#define DBGMSG_PREFIX REDS "<< DBGMSG >> "

#define DBGMSG_V6(msg, arg1) fprintf(stderr, \
                        DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, \
                        __func__, __LINE__, arg1)
```

실습9

23

파일명 : ex9.c

```
void f(void) ;

int
main(void)
{
    DBGMSG_V1;
    DBGMSG_V2;
    DBGMSG_V3("debug message v3");
    DBGMSG_V4("debug message v4");
    f();
    return 0;
}

void
f(void)
{
    int x = 10;
    DBGMSG_V5("debug message v5, x = %d", x);
    DBGMSG_V6("debug message v6, x = %d", x);
}
```

실습9

24

```
$ gcc -Wall --save-temps ex9.c -oex9
$ ./ex9
ex9.c main 30
ex9.c main 31
ex9.c main 32 debug message v3
[ex9.c main 33] : debug message v4
[ex9.c f 42] : debug message v5, x = 10
<< DBGMSG >> [ex9.c f 43] : debug message v6, x = 10
$ gcc -Wall --save-temps ex9.c -oex9
$ vim ex9.i
```

```
int
main(void)
{
    printf("%s %s %d\n", "ex9.c", __func__, 30);
    fprintf(stderr, "%s %s %d\n", "ex9.c", __func__, 31);
    fprintf(stderr, "%s %s %d " "debug message v3" "\n", "ex9.c", __func__, 32);
    fprintf(stderr, "[%s %s %d] : " "debug message v4" "\n", "ex9.c", __func__, 33);
    f();
    return 0;
}

void
f(void)
{
    int x = 10;
    fprintf(stderr, "[%s %s %d] : " "debug message v5, x = %d" "\n", "ex9.c", __func__, 42, x);
    fprintf(stderr, "\x1b[31m" "<< DBGMSG >> " "[%s %s %d] : " "debug message v6, x = %d" "\n"
"\x1b[0m", "ex9.c", __func__, 43, x);
}
```


실습10

25

파일명 : ex10.c

```
#include <stdio.h>

#define REDS      "\x1b[31m"
#define REDE      "\x1b[0m"
#define DBGMSG_PREFIX REDS "<< DBGMSG >> "

#define DBGMSG_V6(msg, arg1) fprintf(stderr, \
                                DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, __func__, __LINE__, arg1)

#define DBGMSG_V7(msg, ...) fprintf(stderr, \
                                DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, __func__, \
                                __LINE__, ##__VA_ARGS__)

int
main(void)
{
    int x = 10;
    double y = 3.14;
    char *str = "Hi";
    DBGMSG_V7("debug message v7, x = %d, str = %s, y = %.2f", x, str, y);
    DBGMSG_V7();
    return 0;
}
```

```
$ ./ex10
```

```
<< DBGMSG >> [ex10.c main 20] : debug message v7, x = 10, str = Hi, y = 3.14
```

```
<< DBGMSG >> [ex10.c main 21] :
```

```
$
```

실습11

26

파일명 : ex11.c

```

#include <stdio.h>
#define REDS      "\x1b[31m"
#define REDE      "\x1b[0m"
#define DBGMSG_PREFIX REDS "<< DBGMSG >> "

#define DBGMSG(msg,...) fprintf(stderr, \
                                DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, __func__, \
                                __LINE__, ##__VA_ARGS__)

void change(int, int*, int**);
int gvala = 1;
int gvalb = 2;

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    DBGMSG("p1:%p, p2:%p", p1, p2);
    change(val, p1, &p2);
    DBGMSG("p1:%p, p2:%p", p1, p2);
    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval  = gvalb;
    tmpp1   = &gvalb;
    *tmpp2  = &gvalb;
    DBGMSG("Here!");
    DBGMSG("tmpval:%d, tmpp1:%p, tmpp2:%p", tmpval, tmpp1, tmpp2);
}

```

실습11

27

```
$ gcc -Wall ex11.c -oex11
$
$ ./ex11
<< DBGMSG >> [ex11.c main 21] : p1:0x804a020, p2:0x804a020
<< DBGMSG >> [ex11.c change 35] : Here!
<< DBGMSG >> [ex11.c change 36] : tmpval:2, tmpp1:0x804a024, tmpp2:0xbfe4f574
<< DBGMSG >> [ex11.c main 23] : p1:0x804a020, p2:0x804a024
val:1, *p1:1, *p2:2
$
```

조건부 컴파일

28

```
#ifdef DEBUG
#define REDS      "\x1b[31m"
#define REDE      "\x1b[0m"
#define DBGMSG_PREFIX REDS "<< DBGMSG >> "

#define DBGMSG(msg,...) fprintf(stderr, \
    DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, __func__, \
    __LINE__, ##__VA_ARGS__)
#else
#define DBGMSG(...)
#endif
```

실습12

29

파일명 : ex12.c

```

#include <stdio.h>
#ifdef DEBUG
#define REDS      "\x1b[31m"
#define REDE      "\x1b[0m"
#define DBGMSG_PREFIX REDS "<< DBGMSG >> "

#define DBGMSG(msg,...) fprintf(stderr, \
                                DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, __func__, \
                                                                __LINE__, ##__VA_ARGS__)

#else
#define DBGMSG(...)
#endif

void change(int, int*, int**);
int gvala = 1; int gvalb = 2;

int
main(void)
{
    int val = gvala;
    int *p1 = &gvala;
    int *p2 = &gvala;

    DBGMSG("p1:%p, p2:%p", p1, p2);
    change(val, p1, &p2);
    DBGMSG("p1:%p, p2:%p", p1, p2);
    printf("val:%d, *p1:%d, *p2:%d\n", val, *p1, *p2);

    return 0;
}

```

실습12

30

파일명 : ex12.c

```
void
change(int tmpval, int* tmpp1, int** tmpp2)
{
    tmpval    = gvalb;
    tmpp1     = &gvalb;
    *tmpp2    = &gvalb;
    DBGMSG("Here!");
    DBGMSG("tmpval:%d, tmpp1:%p, tmpp2:%p", tmpval, tmpp1, tmpp2);
}
```

```
$ gcc -Wall ex12.c -oex12
$ ./ex12
val:1, *p1:1, *p2:2
$ gcc -Wall -DDEBUG ex12.c -oex12
$ ./ex12
<< DBGMSG >> [ex12.c main 25] : p1:0x804a020, p2:0x804a020
<< DBGMSG >> [ex12.c change 39] : Here!
<< DBGMSG >> [ex12.c change 40] : tmpval:2, tmpp1:0x804a024, tmpp2:0xbf92fb64
<< DBGMSG >> [ex12.c main 27] : p1:0x804a020, p2:0x804a024
val:1, *p1:1, *p2:2
$
```

실습13

31

파일명 : ex13.c

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

#ifdef DEBUG
#define REDS      "\x1b[31m"
#define REDE      "\x1b[0m"
#define DBGMSG_PREFIX REDS "<< DBGMSG >> "

#define DBGMSG(msg,...) fprintf(stderr, \
                                DBGMSG_PREFIX "[%s %s %d] : " msg "\n" REDE, __FILE__, __func__, \
                                __LINE__, ##__VA_ARGS__)

#else
#define DBGMSG(...)
#endif

void f(void);

int
main(void)
{
    int i;
    for(i = 1 ; i <= 20 ; i++) {
        DBGMSG("%d job processing", i);
        f();
    }
    printf("completed\n");
    return 0;
}

void f(void)
{
    sleep(1);
}
```

실습13

32

```
$ gcc -Wall ex13.c -oex13
$ !.
./ex13
completed
$
```


실습13

33

```
$ gcc -Wall ex13.c -oex13 -DDEBUG
$ !.
./ex13
<< DBGMSG >> [ex13.c main 24] : 1 job processing
<< DBGMSG >> [ex13.c main 24] : 2 job processing
<< DBGMSG >> [ex13.c main 24] : 3 job processing
<< DBGMSG >> [ex13.c main 24] : 4 job processing
<< DBGMSG >> [ex13.c main 24] : 5 job processing
<< DBGMSG >> [ex13.c main 24] : 6 job processing
<< DBGMSG >> [ex13.c main 24] : 7 job processing
<< DBGMSG >> [ex13.c main 24] : 8 job processing
<< DBGMSG >> [ex13.c main 24] : 9 job processing
<< DBGMSG >> [ex13.c main 24] : 10 job processing
<< DBGMSG >> [ex13.c main 24] : 11 job processing
<< DBGMSG >> [ex13.c main 24] : 12 job processing
<< DBGMSG >> [ex13.c main 24] : 13 job processing
<< DBGMSG >> [ex13.c main 24] : 14 job processing
<< DBGMSG >> [ex13.c main 24] : 15 job processing
<< DBGMSG >> [ex13.c main 24] : 16 job processing
<< DBGMSG >> [ex13.c main 24] : 17 job processing
<< DBGMSG >> [ex13.c main 24] : 18 job processing
<< DBGMSG >> [ex13.c main 24] : 19 job processing
<< DBGMSG >> [ex13.c main 24] : 20 job processing
completed
$
```

실습13

34

shchoi82@localhost:~/dbgmsg

```
[shchoi82@localhost dbgmsg]$ ./ex13 2> stderr_log
```

shchoi82@localhost:~/dbgmsg

```
[shchoi82@localhost dbgmsg]$ tail -f stderr_log
```

```
<< DBGMSG >> [ex13.c main 22] : 1 job processing
<< DBGMSG >> [ex13.c main 22] : 2 job processing
<< DBGMSG >> [ex13.c main 22] : 3 job processing
<< DBGMSG >> [ex13.c main 22] : 4 job processing
<< DBGMSG >> [ex13.c main 22] : 5 job processing
<< DBGMSG >> [ex13.c main 22] : 6 job processing
<< DBGMSG >> [ex13.c main 22] : 7 job processing
<< DBGMSG >> [ex13.c main 22] : 8 job processing
<< DBGMSG >> [ex13.c main 22] : 9 job processing
<< DBGMSG >> [ex13.c main 22] : 10 job processing
<< DBGMSG >> [ex13.c main 22] : 11 job processing
```