Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - 이상훈
gcccompil3r@gmail.com

수강생 - 서재언
20640@naver.com

1. 02/2_2.c(과제풀이)

a. 4byte integer → 1byte char 문자로 받음

b. 4byte입력값 int num = 2004016
→ (char *)&num → 0x001e9430이 little endian 방식으로 0x30, 0x94, 0x1E, 0x00 순으로 배열
→ 배열 & 1000 0000>>1

```
   #include <stdio.h>
2

3 void compute_int_bit(char *val, int size)
4 {
5                   int i, j;
6                   unsigned char comp = 0;
7

8                   printf("val = ");
9

10                  for(i = size - 1; i >= 0; i--)
11                  {
12                                  comp = 128;
13                                  for(j = 1; j <= 8; j++)
14                                  {
15                                                  if(i == size - 1 && j == 1)
16                                                  {
17                                                                  printf("%d ", val[i] & comp ? 1 : 0);
18                                                  }
19                                                  else
20                                                  {
21                                                                  printf("%d", val[i] & comp ? 1 : 0);
22                                                  }
23
```

1. 02/2_2.c(과제풀이)

```
                                        comp >>= 1;
24                              }
25                              printf(" ");
26              }
27 }
28

29 int main(void)
30 {
31              int size = sizeof(int);
32              int val = 2004016;
33

34              compute_int_bit((char *)&val, size);
35

36              return 0;
37 }
```

2. 02/2_4.c(과제풀이)

- 저글링 구현

```
1 #include <stdio.h>
2

3 void pointer_juggling(void)
4 {
5               int n1 = 2, n2 = 4, n3 = 7;
6               int *tmp = NULL;
7               int *n_p[3] = {&n1, &n2, &n3};
8               int **npp = NULL;
9

10              int i = 0, j = 0;;
11

12              printf("n1 = %d\n", *n_p[0]);
13              printf("n2 = %d\n", *n_p[1]);
14              printf("n3 = %d\n\n", *n_p[2]);
15

16              for(;;)
17              {
18                              npp = &n_p[i];
19                              tmp = *npp;
20                              *npp = n_p[i + 1];
21                              n_p[i + 1] = n_p[i + 2];
22                              n_p[i + 2] = tmp;
23

24                              printf("n1 = %d\n", *n_p[0]);
25                              printf("n2 = %d\n", *n_p[1]);
26                              printf("n3 = %d\n\n", *n_p[2]);
27
```

2. 02/2_4.c(과제풀이)

- 저글링 구현

```
28                                      if(j++ == 3)
29                                              break;
30              }
31 }
32

33 int main(void)
34 {
35              pointer_juggling();
36              return 0;
37 }
```

3. 03/1_1.c(과제풀이)

- 문자열 짝수번째 출력

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5               char strArray[100] ={0};
6               int i;
7
8               printf("문자열을 입력하세요: ");
9               scanf("%s", strArray);
10
11              for(i=0; i < 100; i++)
12              {
13                              if(!(i%2))
14                              {
15
16                                              printf("%c ", strArray[i]);
17
18                                              if(strArray[i] == '\0')
19                                              {
20                                                              break;
21                                              }
22                              }
23              }
24
25              printf("\n");
26
27              return 0;
28 }
```

4. 03/1_3.c(과제풀이)

- 배열과 역배열

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5                int array[10] = {3, 77, 10, 7, 4, 9, 1, 8, 21, 33};
6                int reverse[10] = {0};
7                int i;
8
9                printf("현재 숫자배열: ");
10
11               for(i=0; i < 10; i++)
12               {
13                               printf("%d ", array[i]);
14               }
15
16               for(i=0; i < 10; i++)
17               {
18                               reverse[9-i] = array[i];
19               }
20
21               printf("reverse 숫자배열: ");
22
23               for(i=0; i < 10; i++)
24               {
25                               printf("%d ", reverse[i]);
26               }
27
28
29               return 0;
30 }
```

5. 03/1_4.c(과제풀이)

- 홀수합*짝수합 출력

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5               int array[10] = {3, 77, 10, 7, 4, 9, 1, 8, 21, 33};
6               int evenSum=0;
7               int oddSum=0;
8               int i;
9
10              for(i=0; i < 10; i++)
11              {
12                              if(!(i%2))
13                              {
14                                              evenSum += array[i];
15                              }
16                              else
17                              {
18                                              oddSum += array[i];
19                              }
20              }
21
22              printf("현재 숫자: ");
23
24              for(i=0; i < 10; i++)
25              {
26                              printf("%d ", array[i]);
27              }
28
29              printf("\n");
30
31              printf("짝수의 합: %d\n", evenSum);
32              printf("홀수의 합: %d\n", oddSum);
33
34              return 0;
35 }
```

6. 03/1_6.c(과제풀이)

- 3*3 행렬 덧셈, 뺄셈, 곱셈

```
1 #include <stdio.h>
2
3 void multipleMatrix(int a[][3], int b[][3])
4 {
5               int multiple_matrix[3][3] = {0};
6               int result;
7               int i, j, k;
8
9               //행렬 곱셈
10
11              for(i=0; i <3; i++)
12              {
13                              for(j=0; j <3; j++)
14                              {
15                                              result = a[i][j]*b[j][i];
16                              }
17
18                              multiple_matrix[i][k] = result;
19                              k++;
20
21                              if(k >2)
22                              {
23                                 k=0;
24                              }
25              }
26
27              printf("행렬 a[3][3]*b[3][3] = \n");
28
```

6. 03/1_6.c(과제풀이)

- 3*3 행렬 덧셈, 뺄셈, 곱셈

```
29              for(i=0; i<3; i++)
30              {
31                              for(j=0; j <3; j++)
32                              {
33                                              printf("%d " , multiple_matrix[i][j]);
34                              }
35
36                              printf("\n");
37              }
38
39              printf("\n");
40 }
41
42 void addMatrix(int a[][3], int b[][3])
43 {
44
45              int add_matrix[3][3] = {0};
46              int result;
47              int i, j;
48
49              //행렬 덧셈
50
51              for(i=0; i <3; i++)
52              {
53                              for(j=0; j <3; j++)
54                              {
55                                              add_matrix[i][j] = a[i][j]+b[i][j];
56                              }
57              }
58
59              printf("행렬 a[3][3]+b[3][3] = \n");
```

6. 03/1_6.c(과제풀이)

- 3*3 행렬 덧셈, 뺄셈, 곱셈

```
60
61              for(i=0; i<3; i++)
62              {
63                              for(j=0; j <3; j++)
64                              {
65                                              printf("%d " , add_matrix[i][j]);
66                              }
67
68                              printf("\n");
69              }
70
71              printf("\n");
72 }
73
74 void minusMatrix(int a[][3], int b[][3])
75 {
76
77              int minus_matrix[3][3] = {0};
78              int result;
79              int i, j;
80
81              //행렬뺄셈
82
83              for(i=0; i <3; i++)
84              {
85                              for(j=0; j <3; j++)
86                              {
87                                              minus_matrix[i][j] = a[i][j]-b[i][j];
88                              }
89              }
```

6. 03/1_6.c(과제풀이)

- 3*3 행렬 덧셈, 뺄셈, 곱셈

```
91              printf("행렬 a[3][3]-b[3][3] = \n");
92
93              for(i=0; i<3; i++)
94              {
95                              for(j=0; j <3; j++)
96                              {
97                                              printf("%d " , minus_matrix[i][j]);
98                              }
99
100                             printf("\n");
101             }
102
103             printf("\n");
104
105 }
106
107 void divisionMatrix(int a[][3], int b[][3])
108 {
109             int division_matrix[3][3] = {0};
110             int result;
111             int i, j;
112
113             //행렬 나눗셈
114
115             for(i=0; i <3; i++)
116             {
117                             for(j=0; j <3; j++)
118                             {
119                                             division_matrix[i][j] = a[i][j]/b[i][j];
120                             }
121             }
```

6. 03/1_6.c(과제풀이)

- 3*3 행렬 덧셈, 뺄셈, 곱셈

```
122
149             for(i=0; i <3; i++)
150             {
151                             for(j=0; j<3; j++)
152                             {
153                                             printf("%d ", a_matrix[i][j]);
154                             }
155
156                             printf("\n");
157             }
158
159
160             printf("행렬 b:\n");
161
162             for(i=0; i <3; i++)
163             {
164                             for(j=0; j<3; j++)
165                             {
166                                             printf("%d ", b_matrix[i][j]);
167                             }
168
169                             printf("\n");
170             }
171
172             printf("\n");
173
174             multipleMatrix(a_matrix, b_matrix);
175             addMatrix(a_matrix, b_matrix);
176             minusMatrix(a_matrix, b_matrix);
177             divisionMatrix(a_matrix, b_matrix);
178
179
180             return 0;
181 }
```

7. day_6/01.c (6일차 학습내용)

- typedef 선언

```
1 #include <stdio.h>
2

3 typedef int INT;
4 typedef int* PINT;
5

6

7 int main(void)
8 {
9                       INT num = 3;
10                      PINT ptr = &num;
11

12                      printf("num = %d\n", *ptr);
13

14                      return 0;
15

16 }
```

8. day_6/02.c (6일차 학습내용)

- typedef 선언

```
1 #include <stdio.h>
2

3 typedef int INT[5];
4

5 int main(void)
6 {
7               int i;
8               INT arr = {1,2,3,4,5};
9

10              for(i=0; i <5; i++)
11              {
12                              printf("arr[%d] = %d\n", i, arr[i]);
13              }
14

15              return 0;
16 }
```

9. day_6/03.c (6일차 학습내용)

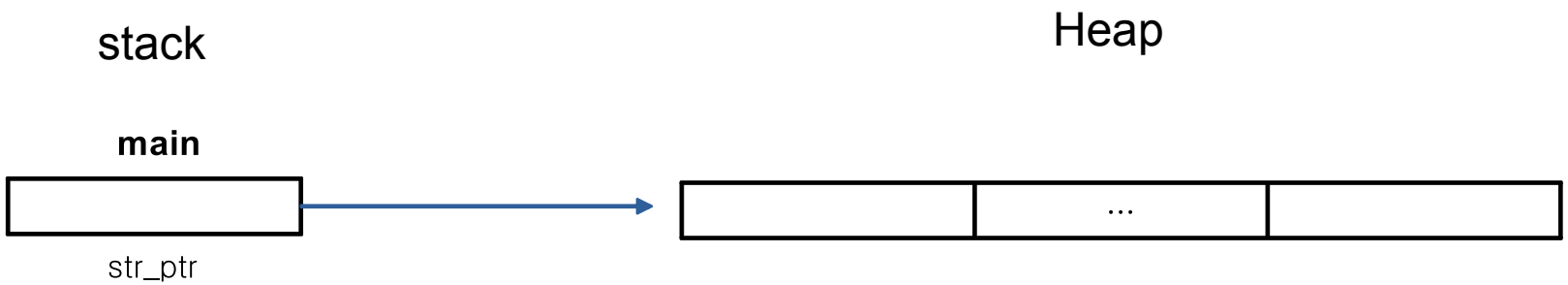
- malloc(), free()

  네트워크장비에서 동적으로 증가되는 메모리확장을 위해 사용(라우터, 스위치, 도청기)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3

4 int main(void)
5 {
6                   char *str_ptr = (char *)malloc(sizeof(char)*20);
7

8                   printf("Input String: ");
9                   scanf("%s", str_ptr);
10

11

12                  if(str_ptr != NULL)
13                                  printf("string = %s\n", str_ptr);
14

15                  free(str_ptr);
16

17                  return 0;
18

19 }
```

9. day_6/03.c (6일차 학습내용)

- 메모리구성



10. day_6/04.c (6일차 학습내용)

- calloc

malloc과 기능유사, 할당 바이트 직접입력

```
1 #include <stdio.h>
2 #include <stdlib.h>
3

4 int main(void)
5 {
6                   int *num_ptr = (int *)calloc(2, sizeof(int));
7

8                   printf("Input Integer: ");
9                   scanf("%d%d", &num_ptr[0], &num_ptr[1]);
10

11                  if(num_ptr != NULL)
12                                  printf("Integer = %d %d\n", num_ptr[0], num_ptr[1]);
13

14                  free(num_ptr);
15

16                  return 0;
17 }
18
```

11. day_6/05.c (6일차 학습내용)

- struct

· sizeof() 취급 printf() 옵션은 %lu(long unsigned)
· struct는 커스텀 데이터 타입

```
1 #include <stdio.h>
2 #include <stdlib.h>
3

4 int main(void)
5 {
6                   int *num_ptr = (int *)calloc(2, sizeof(int));
7

8                   printf("Input Integer: ");
9                   scanf("%d%d", &num_ptr[0], &num_ptr[1]);
10

11                  if(num_ptr != NULL)
12                                  printf("Integer = %d %d\n", num_ptr[0], num_ptr[1]);
13

14                  free(num_ptr);
15

16                  return 0;
17 }
18
```

12. day_6/06.c (6일차 학습내용)

- typedef struct

· 구조체는 선언될 때 메모리에 할당
· 커널 소스코드에서 __(밑줄 2개) 선언된 문장은 되도록이면 건들지 말라는 의미

```
1 #include <stdio.h>
2
3 #define NAME_LEN 30
4 #define ID_LEN                  15
5
6
7 typedef struct __id_card{
8                 char name[NAME_LEN];
9                 char id[ID_LEN];
10                unsigned int age;
11 }id_card;
12
13
14 int main(void)
15 {
16                int i;
17                id_card arr[2] = {
18                {"Marth Kim", "800900-112123", 23},
19                {"july eun", "123456-1234567", 31}
20                };
21
22                for(i=0; i < 2; i++)
23                {
24                                printf("name = %s, id = %s, age = %d\n", arr[i].name, arr[i].id, arr[i].age);
25                }
26
27                return 0;
28 }
```

13. day_6/07.c (6일차 학습내용)

- 구조체안에 구조체

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 typedef struct __id_card{
6                 char *name;
7                 char *id;
8                 unsigned int age;
9 }id_card;
10
11 typedef struct __city{
12                id_card *card;
13                char city[30];
14 }city;
15
16 int main(void)
17 {
18                int i;
19                city info = {NULL, "Seoul"};
20                info.card = (id_card *)malloc(sizeof(id_card));
21
22                info.card->name = "tony";
23                info.card->id = "123456-1234567";
24                info.card->age = 33;
25
26                printf("city = %s, name = %s, id = %s, age = %d\n",
27                                info.city, info.card->name, info.card->id, info.card->age);
28
29                free(info.card);
30
31                return 0;
32 }
```

14. day_6/08.c (6일차 학습내용)

- 구조체안에 구조체

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 typedef struct __id_card{
6                 char *name;
7                 char *id;
8                 unsigned int age;
9 }id_card;
10
11 typedef struct __city{
12                id_card *card;
13                char city[30];
14 }city;
15
16 int main(void)
17 {
18                int i;
19                city info = {NULL, "Seoul"};
20                info.card = (id_card *)malloc(sizeof(id_card));
21
22                info.card->name = "tony";
23                info.card->id = "123456-1234567";
24                info.card->age = 33;
25
26                printf("city = %s, name = %s, id = %s, age = %d\n",
27                                info.city, info.card->name, info.card->id, info.card->age);
28
29                free(info.card);
30
31                return 0;
32 }
```

14. day_6/08.c (6일차 학습내용)

- 구조체안에 구조체

15. day_6/09.c (6일차 학습내용)

- 구조체 타입 포인터

```
1 #include <stdio.h>
2
3 typedef struct __data{
4                       int val;
5                       struct __data *data_ref;
6 }data;
7
8
9 int main(void)
10 {
11                      int i;
12                      data *data_p;
13
14                      data d1 = {3, NULL};
15                      data d2 = {7, NULL};
16
17
18                      d1.data_ref = &d2;
19                      d2.data_ref = &d1;
20
21                      data_p = &d1;
22
23                      for(i=0; i <= 10; i++)
24                      {
25
26                                              printf("%3d", data_p->val);
27
28                                              (data_p->val)++;
29
30                                              data_p = data_p->data_ref;
31
32
```

15. day_6/09.c (6일차 학습내용)

```
                        if(!(i%2))
33                      {
34                              printf("\t");
35                      }
36
37              }
38
39              printf("\n");
40
41              return 0;
42 }
```

16. day_6/10.c (6일차 학습내용)

- enum을 통한 루프문작성

```
1 #include <stdio.h>
2

3 typedef enum __packet{
4

5                       ATTACK,
6                       DEFENCE,
7                       HOLD,
8                       STOP,
9                       SKILL,
10                      REBIRTH,
11                      DEATH = 44,
12                      KILL,
13                      ASSIST
14 }packet;
15

16 int main(void)
17 {
18                      packet packet;
19
20                      for(packet = ATTACK; packet <= REBIRTH; packet++)
21                      {
22                                      printf("enum num = %d\n", packet);
23                      }
24

25                      for(packet = DEATH; packet <= ASSIST; packet++)
26                      {
27                                      printf("enum num = %d\n", packet);
28                      }
29

30                      return 0;
31 }
```

17. day_6/11.c (6일차 학습내용)

- 함수포인터

```
1 #include <stdio.h>
2

3 void aaa(void)
4 {
5                       printf("aaa called\n");
6 }
7

8 void bbb(void(*p)(void))
9 {
10

11                      p();
12                      printf("bbb called\n");
13 }
14

15 int main(void)
16 {
17                      bbb(aaa);
18
19                      return 0;
20 }
return 0;
31 }
```

18. day_6/11.c (6일차 학습내용)

- 함수포인터

· 함수포인터를 활용해 객체 구현가능(함수이름은 주소이기 때문 응용가능)

void (*signal(int signum, void (* handler)(int)))(int) ;

*프로토타입

리턴: void(*) int
함수명: signal
인자: int signum, void (*handler)(int)

· 함수 프로토타입

리턴, 함수명, 인자에 대한 기술서

· void (*p)(void):

void 를 리턴하고 void를 인자로 취하는 함수포인터

(예제)
int aaa(int k);
int *p(int);

19. day_6/12.c (6일차 학습내용)

- 구조체를 활용한 함수포인터 활용

```
1 #include <stdio.h>
2
3
4 typedef struct add_class
5 {
6                   int in1;
7                   int in2;
8                   double dn1;
9                   double dn2;
10
11                  int (*int_op)(int, int);
12                  double (*double_op)(double, double);
13 }ac;
14
15
16 int Iadd(int n1, int n2)
17 {
18                  return n1+n2;
19 }
20
21 int Imul(int n1, int n2)
22 {
23                  return n1*n2;
24 }
25
```

19. day_6/12.c (6일차 학습내용)

- 구조체를 활용한 함수포인터 활용

```
26 double Dadd(double n1, double n2)
27 {
28                  return n1*n2;
29 }
30
31
32
33 int main(void)
34 {
35                  int res;
36                  double dres;
37                  ac tc_inst = {3, 7, 3.3, 7.7, NULL, NULL};
38
39                  tc_inst.int_op = Iadd;
40                  res = tc_inst.int_op(tc_inst.in1, tc_inst.in2);
41                  printf("res = %d\n", res);
42
43                  tc_inst.int_op = Imul;
44                  res = tc_inst.int_op(tc_inst.in1, tc_inst.in2);
45                  printf("res = %d\n", res);
46
47                  tc_inst.double_op = Dadd;
48                  dres = tc_inst.double_op(tc_inst.dn1, tc_inst.dn2);
49                  printf("dres = %lf\n", dres);
50
51                  return 0;
52
53
54 }
```