Chapter 6. Functions and Program Structure

April, 2016

Seungjae Baek

Dept. of software
Dankook University
http://embedded.dankook.ac.kr/~baeksj

이 장의 강의 목표

- 문자의 입력 방법을 이해한다.
- 중첩된 if문을 이해한다.
- while 반복문의 사용법을 익힌다.
- do 반복문의 사용법을 익힌다.
- 중첩된 반복문을 이해한다.
- break문의 사용법을 익힌다.
- continue문의 사용법을 익힌다.
- switch문의 사용법을 익힌다.
- goto문의 사용법을 익힌다.
- 이 장의 결론

문자의 입력 (1/3)

- 문자 입력

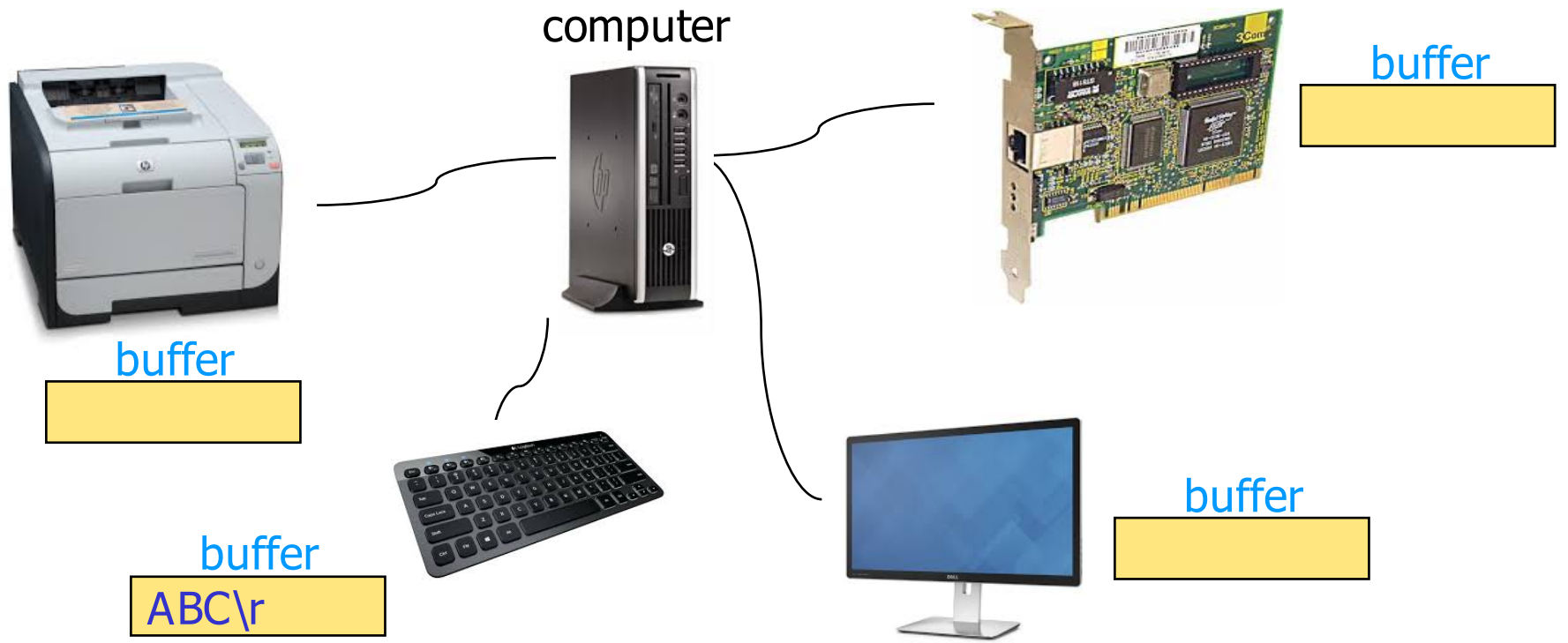
line_buffering.c

```
#include <stdio.h>

int main()
{
        char ch;

        for (;;) {
                printf("\rEnter characters : ");
                ch = getchar();
                printf("\n\rYou typed: %c\n", ch);
        }

        return 0;
}
```

scanf("%c", &ch);

☞ 키보드에서 **A**를 입력해 보세요.
실제 수행시켜 보면 예상과는 조금 다르게 동작.. ➔ **line buffer** 입력 방식 때문

문자의 입력 (2/3)



- 컴퓨터는 다양한 장치(**device**)를 사용한다.
- 대부분의 장치(**device**)는 버퍼라는 메모리 공간을 갖는다.
- **keyboard**에서 **A**를 입력하면?
  keyboard에서 **BC**를 입력하면?
  keyboard에서 **Enter**를 입력하면?
- 언제 **keyboard** 버퍼의 내용을 컴퓨터의 **CPU**(결국 응용)로 전달할 것인가?
  ➔ **no buffering, line buffering, full buffering**

문자의 입력 (3/3)

- 라인 버퍼링을 사용하지 않는 문자 입력 방식

no_buffering.c

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

int main()
{
            char ch;
            initscr();
            for (;;) {
                 printf("Enter characters : ");
                 ch = getchar();
                 printf("\nYou typed: %c\n", ch);
            }
            endwin
            return 0;
}
```

no_buffering2.c

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

int main()
{
        struct termios old_tio, new_tio;
        char ch;
        tcgetattr(STDIN_FILENO,&old_tio);
        new_tio=old_tio;
        new_tio.c_lflag &=(~ICANON & ~ECHO);
        tcsetattr(STDIN_FILENO,TCSANOW,&new_tio);
        for (;;) {
                printf("Enter characters : ");
                ch = getchar();
                printf("\nYou typed: %c\n", ch);
        }
        tcsetattr(STDIN_FILENO,TCSANOW,&old_tio);
        return 0;
}
```

☞ 역시 키보드에서 **A**를 입력해 보세요.

☞ 버퍼링 방식 제어 가능 **(canonical mode)**
☞ 그런데.. 장치에 버퍼 공간은 왜 필요할까?

중첩된 if 문

- if문 내부에 또 다른 if 문 사용 가능
  - ✓ 들여쓰기가 중요
  - ✓ 잘못된 들여쓰기의 예

```
if (a)
if (b)  printf("a and b are true\n");
else printf("To which statement does this else apply?");
```

```
if (a)
    if (b)
        printf("a and b are true\n");
    else
        printf("To which statement does this else apply? ➔ b");
```

구분해서 사용

```
if (a) {
    if (b)
        printf("a and b are true\n");
}
else
    printf("To which statement does this else apply? ➔ a");
```

반복문의 여러 형태

- for 반복문의 융통성

for_type.c

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

int main()
{
            int i;
            char ch;

            printf("Enter an integer: ");
            scanf("%d", &i);
            for (; i; i--)
                        printf("%d\n", i);

            for (i=1; i<11; )
                        printf("%d\n", i++);

            for (i=11; i<11; i++)
                        printf("%c\n", '\a');

            return 0;
}
```

while 반복문

- 형식

```
while (expression)
    statement;
```

- 1에서 10까지의 합은?

while_sum.c

```
#include <stdio.h>

int main()
{
        int i = 1;
        int total = 0;

        while(i < 10) {
                total = total + i;
                i++;
        }
        printf("total = %d\n", total);
        return 0;
}
```

for문은 while문으로 변화 가능

```
for (i=0; i<10; i++)
    statements;

i=0;
while (i<10) {
    statements;
    i++;
}
```

do 반복문 (1/2)

- 형식

```
do {
    statement;
} while (expression);
```

- 1에서 10까지의 합은?

do_sum.c

```
#include <stdio.h>

int main()
{
        int i = 1;
        int total = 0;

        do {
                total = total + i;
                i++;
        } while(i < 10);
        printf("total = %d\n", total);
        return 0;
}
```

for, while, do-while 변화 가능

```
for (i=0; i<10; i++)
    statements;

i=0;
while (i<10) {
    statements;
    i++;
}

i=0;
do {
    statements;
    i++;
} while (i<10);
```

do 반복문 (2/2)

- 형식

do_getchar.c

```
#include <stdio.h>
#include <unistd.h>
#include <termios.h>
int main()
{
        struct termios old_tio, new_tio;
        char ch;
        tcgetattr(STDIN_FILENO,&old_tio);
        new_tio=old_tio;
        new_tio.c_lflag &=(~ICANON & ~ECHO);
        tcsetattr(STDIN_FILENO,TCSANOW,&new_tio);
        do {
             printf("Enter characters : ");
             ch = getchar();
             printf("\nYou typed: %c\n", ch);
        } while (ch != 'q');
        tcsetattr(STDIN_FILENO,TCSANOW,&old_tio);
        return 0;
}
```

☞ **do 반복문이 for나 while 반복문과 다른점은?**
위 예제를 while 반복문으로 수정해 보세요.

중첩된 반복문 (1/2)

- 블록 내에는 어떠한 문장도 사용할 수 있다!

for_block_01.c

```
#include <stdio.h>

int main()
{
        int i, j;
        int sum;

        for (i=1; i<=10; i++) {
                sum = 0;
                for (j=1; j<=i; j++)
                        sum = sum + j;
                printf("The sum from 1 to %d = %d\n", i, sum);
        }

        for (i=0; i<26; i++)
                printf("%2c", 'A'+i);
        return 0;
}
```

중첩된 반복문 (2/2)

- 블록 내에는 어떠한 문장도 사용할 수 있다!

for_block_02.c

```
#include <stdio.h>

int main()
{
        int i, j;

        for (i=1; i<=5; i++) {
                for (j=1; j<=(5-i); j++)
                        printf(" ");
                for (j=1; j<=(2*i-1); j++)
                        printf("*");
                printf("\n");
        }
}
```

break 문 (1/2)

- 반복문의 밖으로 제어 이동

WhenOver5000.c

```
#include <stdio.h>

int main(void)
{
        int sum=0, num=0;

        while(1)
        {
                sum+=num;
                if(sum>5000)
                        break;
                num++;
        }

        printf("sum: %d \n", sum);
        printf("num: %d \n", num);
        return 0;
}
```

break 문 (2/2)

- 중복된 반복문에서는?

For_break.c

```
#include <stdio.h>

int main()
{
        int i,j;

        for (i=1; i<10; i++) {
                for (j=1; j<10; j++) {
                        printf("(%d, %d) ", i, j);
                        if (j == 5)
                                break;
                }
                printf("\n");
        }
}
```

continue 문

- 반복문의 조건 확인 위치로 제어 이동

For_break2.c

```
#include <stdio.h>

int main(void)
{
        int num;
        printf("start! ");

        for(num=0; num<20; num++)
        {
                if(num%2==0 || num%3==0)
                        continue;
                printf("%d ", num);
        }
        printf("end! \n");
        return 0;
}
```

☞ **continue** 문은 예외 처리에서 많이 사용

switch 문 (1/4)

- switch: 다중 선택문 (보통 3이상 선택 가능할 때 사용)

EnglishSchool.c

```
#include <stdio.h>

int main(void)
{
        int num;
        printf("1이상 5이하의 정수 입력: ");
        scanf("%d", &num);

        switch(num)
        {
        case 1:
                printf("1은 ONE \n");
                break;
        case 2:
                printf("2는 TWO \n");
                break;
        case 3:
                printf("3은 THREE \n");
                break;
        case 4:
                printf("4는 FOUR \n");
                break;
        case 5:
                printf("5는 FIVE \n");
                break;
        default:
                printf("I don't know! \n");
        }
        return 0;
}
```

switch 문 (2/4)

- 주의 사항
  - ✓ switch문은 char와 int만 사용 가능
  - ✓ case 문에 두 개 이상의 같은 상수를 사용할 수 없음
  - ✓ case 문과 관련된 문장들은 반드시 중괄호를 사용할 필요가 없음
  - ✓ 반드시 break를 사용할 필요는 없음

switch 문 (3/4)

AdvancedEnglishSchool.c

```
#include <stdio.h>

int main(void)
{
    char sel;
    printf("M 오전, A 오후, E 저녁 \n");
    printf("입력: ");
    scanf("%c", &sel);

    switch(sel)
    {
    case 'M':
    case 'm':
          printf("Morning \n");
          break;
    case 'A':
    case 'a':
          printf("Afternoon \n");
          break;
    case 'E':
    case 'e':
          printf("Evening \n");
          break;     // 사실 불필요한 break문!
    }
    return 0;
}
```

switch 문 (4/4)

- 연습문제:
  - ✓ 영문자를 입력 받아 자음/모음을 구별하는 프로그램 작성

vowel.c

```
#include <stdio.h>

int main()
{
    char ch;

    printf("\nEnter the letter: ");
    ch = getchar();

    switch (ch) {
          case 'a':
          case 'e':
          case 'i':
          case 'o':
          case 'u':
                    printf(" is a vowel\n");
                    break;
          default:
                    printf(" is a constant\n");
    }
    return 0;
}
```

goto 문

- 임의의 위치로 제어 이동 (무조건 분기)

GoToBasic.c

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("자연수 입력: ");
    scanf("%d", &num);

    if(num==1)
        goto ONE;
    else if(num==2)
        goto TWO;
    else
        goto OTHER;

ONE:
    printf("1을 입력하셨습니다! \n");
    goto END;
TWO:
    printf("2를 입력하셨습니다! \n");
    goto END;
OTHER:
    printf("3 혹은 다른 값을 입력하셨군요! \n");

END:
    return 0;
}
```

레이블

이 장의 결론

- 문자의 입력 방법인 getchar()를 배움
- Buffering에 대한 이해
- while, do 반복문의 사용법 이해
- break, continue 문의 사용법 이해
- switch, goto 문의 사용법 이해