Clickseo.com

C Programming

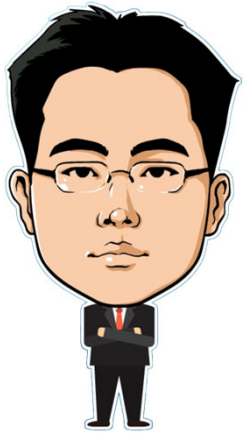
동적 메모리 할당

(Dynamic Memory Allocation)

Seo, Doo-Ok

Clickseo.com

clickseo@gmail.com



cc BY NC ND

목 차

백문이불여일타(百聞而不如一打)

- 동적 메모리 할당

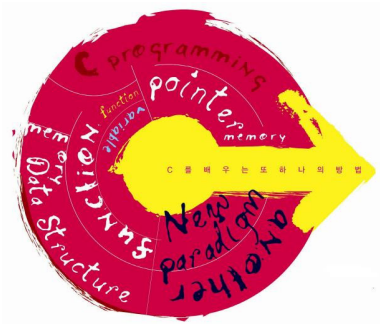
- 2차원 배열과 동적 메모리 할당

동적 메모리 할당

BY NC ND

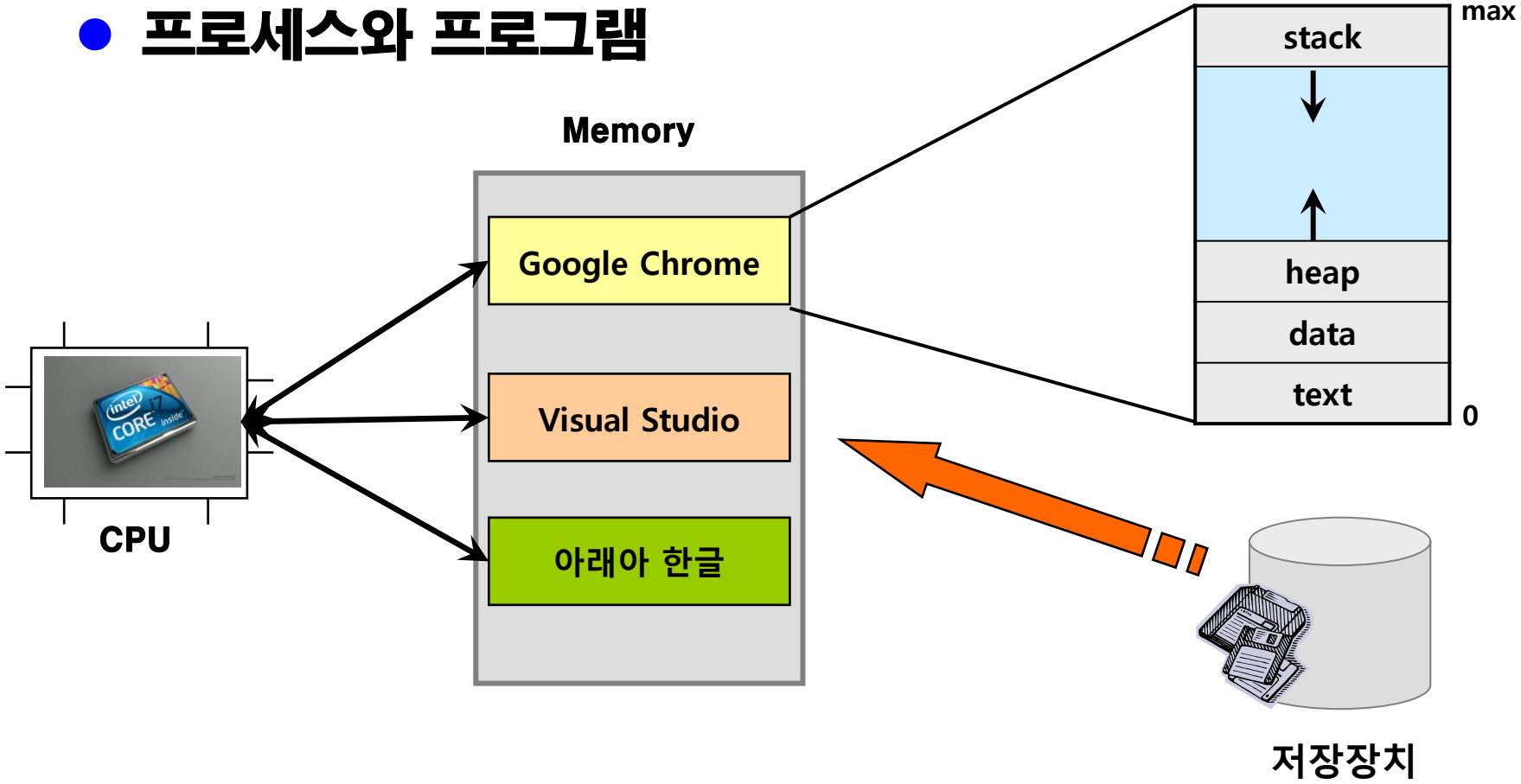
백문이불여일타(百聞而不如一打)

- **동적 메모리 할당**
  - 동적 메모리 할당 관련 함수
  - 피보나치 수열: 동적 프로그래밍
- **2차원 배열과 동적 메모리 할당**



동적 메모리 할당 (1/2)

- 프로세스와 프로그램



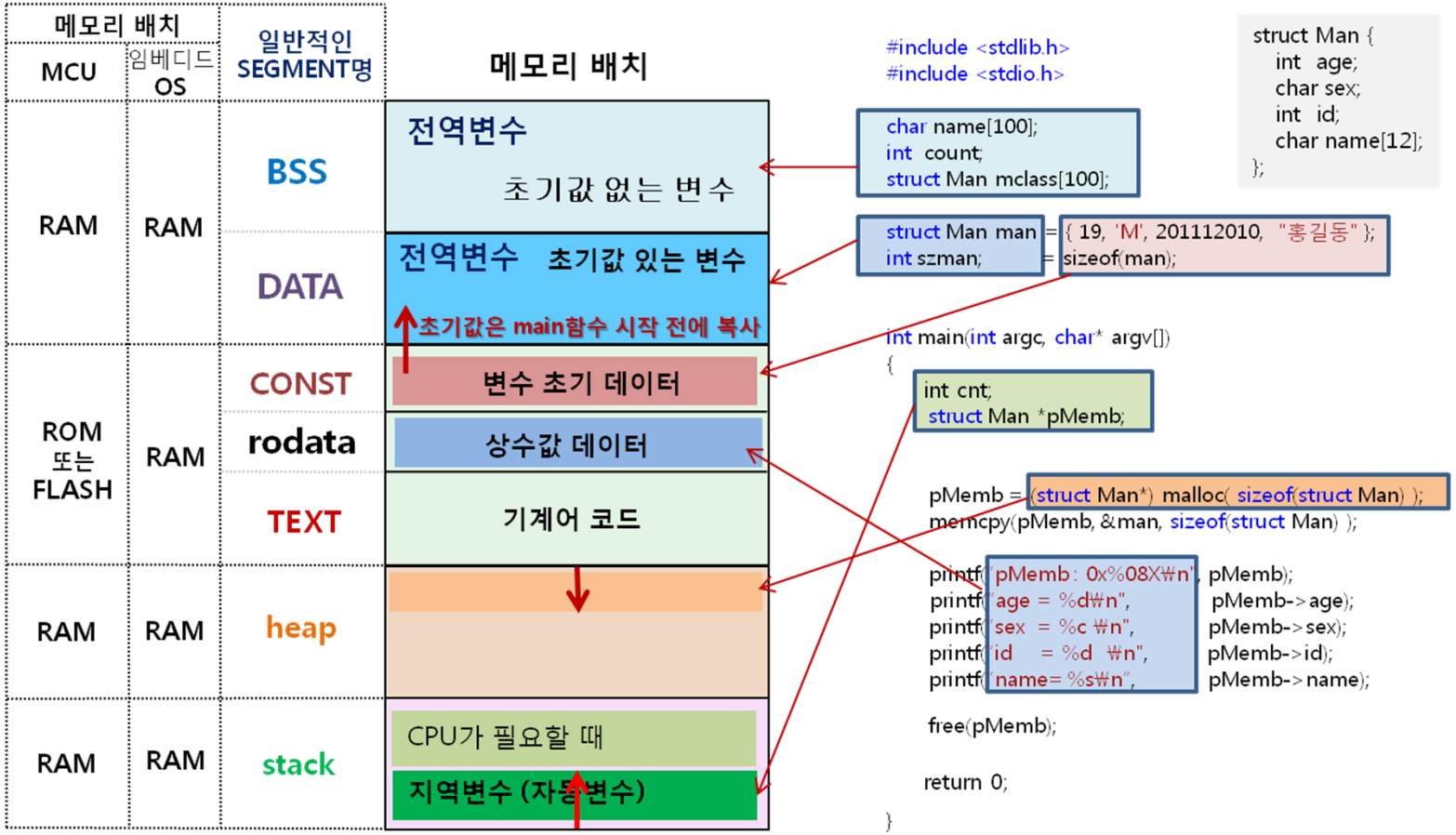
프로세스: 운영체제에서 프로세스는 "실행중인 프로그램"

프로그램: 컴퓨터를 실행시키기 위해 차례대로 작성된 "명령어 집합"

동적 메모리 할당 (2/2)

● 프로세스와 프로그램

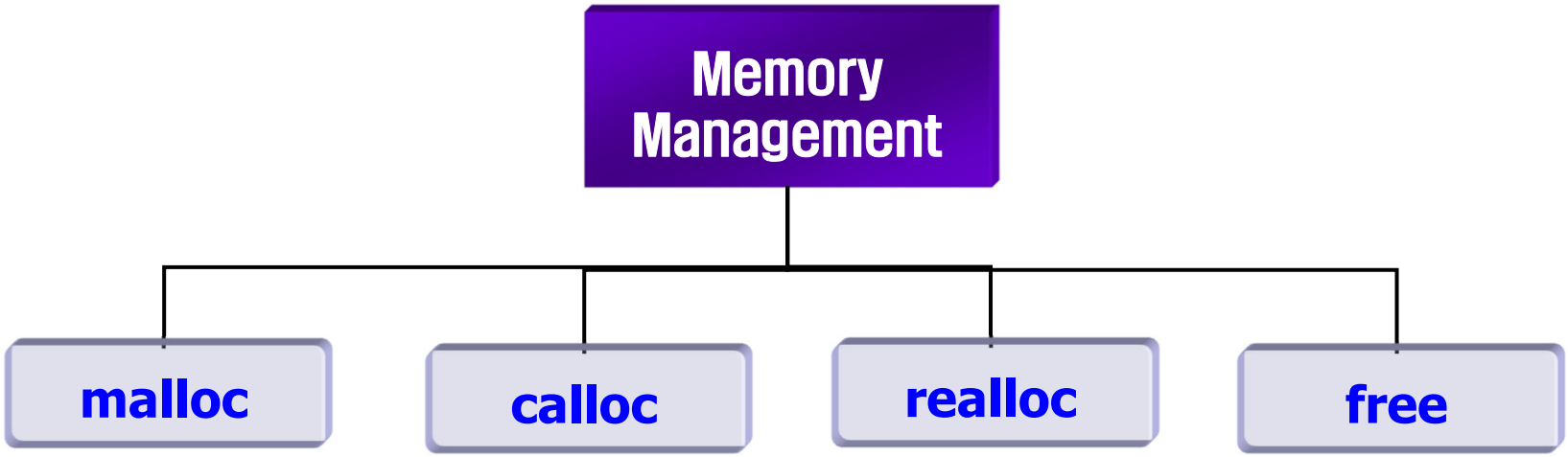
❍ C 언어에 의한 메모리 맵: 일반화 및 단순화 된 내용



[출처: "동적메모리할당", https://ko.wikipedia.org/, Wikepedia.]

동적 메모리 할당: 관련 함수 (1/7)

- **동적 메모리 할당:** 관련 함수
  - 동적 메모리 관리에 쓰이는 4가지 함수
    - 표준 라이브러리 헤더 파일 <stdlib.h> 에서 찾을 수 있다.
    - 메모리 할당: malloc, calloc, realloc
    - 메모리 해제: free

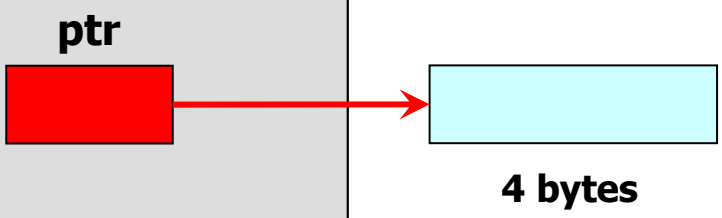


동적 메모리 할당: 관련 함수 (2/7)

- **메모리 할당: malloc**
  - 매개변수로 필요한 메모리의 바이트의 수를 가지며, 그 바이트 수를 수용할 수 있는 크기의 메모리 블록을 할당한다.
    - 할당된 메모리의 첫 번째 바이트를 void 포인터로 되돌린다.
    - 할당된 메모리는 초기화되어 있지 않다(쓰레기 값).

```
void    *malloc(size_t  size);
```

```
int         *ptr = NULL;

ptr = (int *)malloc(sizeof(int))
if (ptr == NULL) {
            printf("동적 메모리 할당 실패!!! ₩n");
            exit(1);
}
```

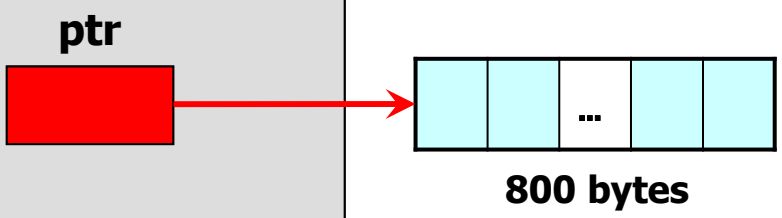


동적 메모리 할당: 관련 함수 (3/7)

- **인접 메모리 할당: calloc**
  - 일반적으로 배열을 위한 메모리 할당에 주로 쓰인다.
    - 특정 크기의 요소들을 배열로 담을 수 있을 만큼 연속적인 메모리 블록을 할당
      - 할당된 배열의 첫 번째 요소를 가리키는 포인터를 반환
      - 할당된 메모리를 초기화(즉, 할당된 메모리의를 0 으로 초기화)

```
void    *calloc(size_t element-count, size_t element-size);
```

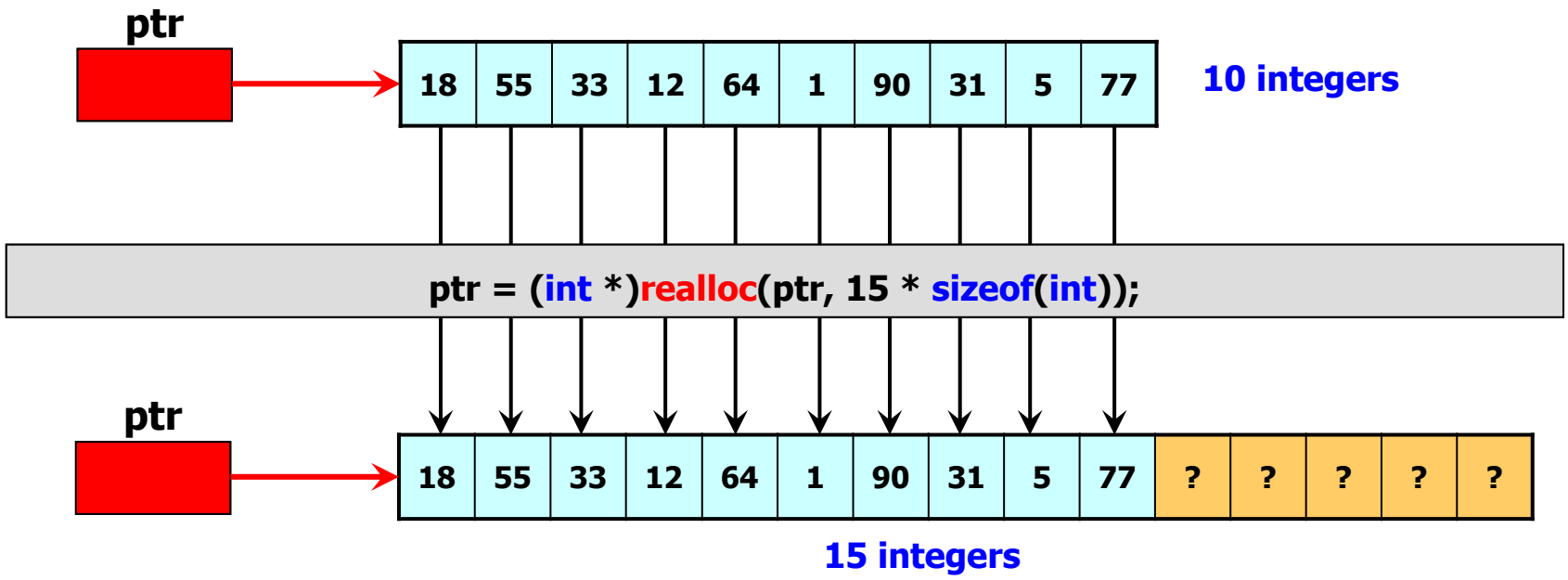
```
int         *ptr = NULL;

ptr = (int *)calloc(200, sizeof(int))
if (ptr == NULL) {
            printf("동적 메모리 할당 실패!!! ₩n");
            exit(1);
}
```



동적 메모리 할당: 관련 함수 (4/7)

- **메모리 재할당: realloc**
  - 메모리 재할당에 쓰인다.

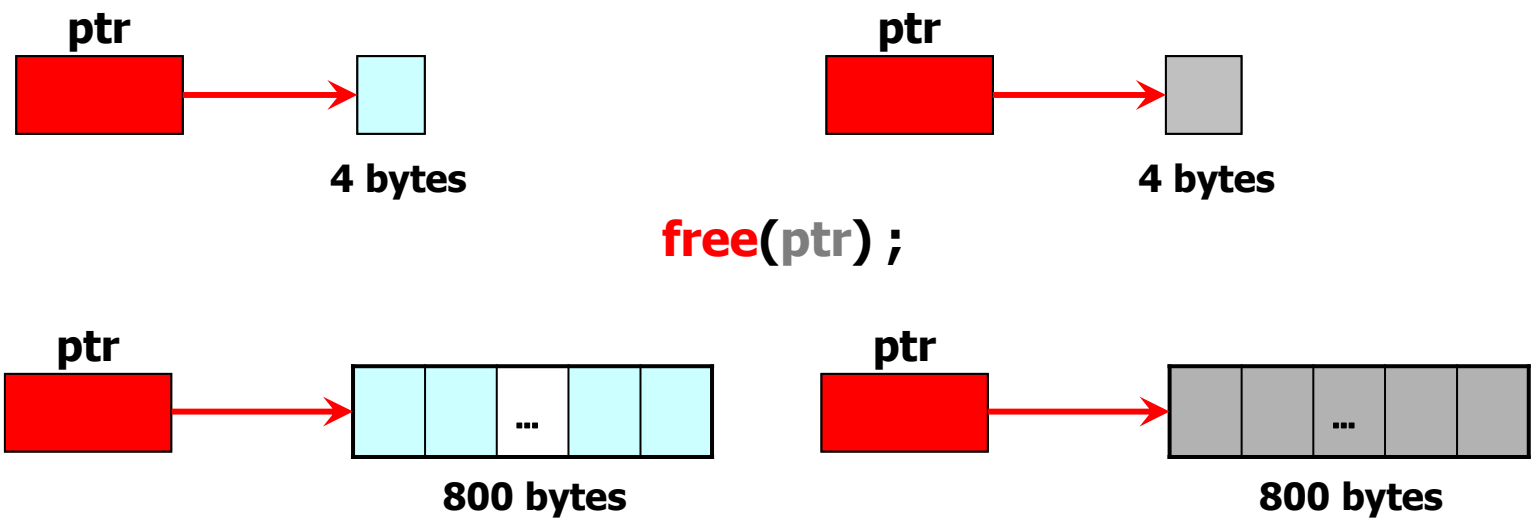
```
void      *realloc(void *ptr, size_t  newSIZE);
```



동적 메모리 할당: 관련 함수 (5/7)

- **메모리 해제: free**
  - malloc, calloc, realloc에 의해 할당된 메모리가 더 이상 필요 없을 때는 free 함수를 사용하여 해제한다.

```
void      free(void *ptr);
```



동적 메모리 할당: 관련 함수 (6/7)

예제 12-1: 동적 메모리 할당 -- malloc, free

```
#include <stdio.h>
#include <stdlib.h>   // malloc, free
#define strMAXSIZE 12
int main(void)
{
        // char  name[12];
        char     *pName = NULL;

        // 동적 메모리 할당: 1차원 배열
        pName = (char *)malloc(sizeof(char) * strMAXSIZE);
        if(pName == NULL)      {
                printf("동적 메모리 할당 실패!!! \n");
                exit(1);
        }
        printf("이름: ");
        gets_s(pName, sizeof(char)*strMAXSIZE);     // gets(pName);
        printf("\n당신의 이름은 \"%s\" 입니다.\" \n", pName);

        // 동적 메모리 해제
        free(pName);
        return 0;
}
```

Microsoft Visual Studio 디버그

```
이름: 홍길동

당신의 이름은 "홍길동" 입니다."

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

pName

12 bytes

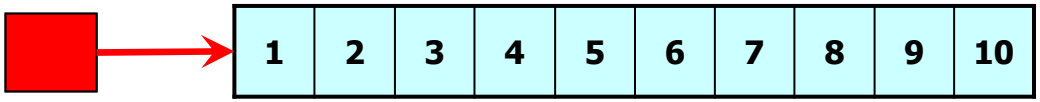
동적 메모리 할당: 관련 함수 (7/7)

예제 12-2: 동적 메모리 할당 -- calloc, free

```
#include <stdio.h>
#include <stdlib.h>   // calloc, free
#include <memory.h>   // memset // #include <string.h>
#define arrMAXSIZE 10
int main(void)
{
        // int    arr[arrMAXSIZE];
        int       *pArr = NULL;

        // pArr = (int *)malloc(arrMAXSIZE * sizeof(int));
        // pArr = memset(pArr, 0, arrMAXSIZE * sizeof(int));
        pArr = (int *)calloc(arrMAXSIZE, sizeof(int));
        if(pArr == NULL)        {
                puts("동적 메모리 할당 실패!!! ");
                exit(1);
        }
        printf("%d개의 정수를 입력하세요. \n", arrMAXSIZE);
        for(int i = 0; i<arrMAXSIZE; ++i)
                scanf_s("%d", pArr + i);      // scanf("%d", pArr + i);

        printf("\n ### 입력된 데이터 ### \n");
        for (int i=0; i<arrMAXSIZE; ++i)
                printf("%3d", *(pArr + i));
        printf("\n");

        free(pArr);
        return 0;
}
```

Microsoft Visual Studio 디버그

```
10개의 정수를 입력하세요.
1
2
3
4
5
6
7
8
9
10

 ### 입력된 데이터 ###
  1  2  3  4  5  6  7  8  9 10

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```



pArr

40 bytes

CC BY NC ND

동적 메모리 할당

피보나치 수열: 동적 프로그래밍

피보나치 수열: 동적 프로그래밍 (1/3)

연습문제 12-3: 피보나치 수열 -- 재귀적 용법

```c
#include <stdio.h>
int Fibo(int num);
int main(void)
{
        int     num;

        printf("### 피보나치 수열 구하기 ### \n\n");
        printf("몇 번째 수열까지 출력할까요:");
        scanf_s("%d", &num);                    // scanf ("%d", &num);

        for (int i=1; i<=num; ++i) {
                if (i % 5)      printf("%8d", Fibo(i));
                else            printf("%8d\n", Fibo(i));
        }
        printf("\n");
        return 0;
}

// 피보나치 수열: 재귀적 용법
int Fibo(int num) {
        // 재귀 함수: 탈출 조건
        if (num == 1 || num == 2)
                return 1;
        return Fibo(num - 1) + Fibo(num - 2);     // O(2^n)
}
```

Microsoft Visual Studio 디버그

```
### 피보나치 수열 구하기 ###

몇 번째 수열까지 출력할까요: 35
       1       1       2       3       5
       8      13      21      34      55
      89     144     233     377     610
     987    1597    2584    4181    6765
   10946   17711   28657   46368   75025
  121393  196418  317811  514229  832040
 1346269 2178309 3524578 5702887 9227465


C:\Users\click\OneDrive\문서\cppClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

Microsoft Visual Studio 디버그

```
### 피보나치 수열 구하기 ###

몇 번째 수열까지 출력할까요: 35
       1       1       2       3       5
       8      13      21      34      55
      89     144     233     377     610
     987    1597    2584    4181    6765
   10946   17711   28657   46368   75025
  121393  196418  317811  514229  832040
 1346269 2178309 3524578 5702887 9227465
35번째 피보나치 수열 계산 시간: 0.198

C:\Users\click\OneDrive\문서\cppClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

피보나치 수열: 동적 프로그래밍 (2/3)

연습문제 12-4: 피보나치 수열 -- 동적 프로그래밍 (1/2)

```
// 피보나치 수열: 동적 프로그래밍 -- (1) 동적 배열: 동적 메모리 할당
int Fibo(int num) {
        // 재귀 함수: 탈출 조건
        if(num == 1 || num == 2)
                return 1;

// 동적 배열: 동적 메모리 할당
// int  *pArr = (int *)realloc(NULL, num * sizeof(int))
// int  *pArr = (int *)malloc(num * sizeof(int));
        int     *pArr = (int*)calloc(num, sizeof(int));
        if (pArr == NULL) {
                printf("메모리 할당 실패!!!\n");
                exit(100);
        }

        // pArr[0] = pArr[1] = 1;
        *pArr = *(pArr + 1) = 1;

        int     i, temp;
        for(i=2; i < num; ++i)
                *(pArr + i) = *(pArr + i - 1) + *(pArr +i - 2);
        temp = *(pArr + i - 1);
        free(pArr);

        return temp;
}
```

```
Microsoft Visual Studio 디버그

### 피보나치 수열 ###

몇 번째 수열까지 출력할까요: 35
      1       1       2       3       5
      8      13      21      34      55
     89     144     233     377     610
    987    1597    2584    4181    6765
  10946   17711   28657   46368   75025
 121393  196418  317811  514229  832040
1346269 2178309 3524578 5702887 9227465

35 번째 피보나치 수열 계산 시간: 0.004000 초

C:\Users\click\OneDrive\문서\cClickseo\x64\Debug\
이 창을 닫으려면 아무 키나 누르세요...
```

```
Fibonacci(n)
{
   f[1] ← f[2] ← 1;
   for i ← 3 to n
      f[i] ← f[i-1] +f[i-2];
   return f[n];
}
```

피보나치 수열: 동적 프로그래밍 (3/3)

연습문제 12-4: 피보나치 수열 -- 동적 프로그래밍 (2/2)

```
// 피보나치 수열: 동적 프로그래밍 -- (2) 일반 변수
int  Fibo(int  num) {
        if(num == 1 || num == 2)
                return 1;

        // 일반 변수: first, second, res
        int     first = 1, second = 1, res = 0;
        for(int i=2; i<num; ++i) {
                res  =  first  +  second;
                first = second;
                second  =  res;
        }
        return res;
}
```

Microsoft Visual Studio 디버그

```
### 피보나치 수열 ###

몇 번째 수열까지 출력할까요: 35
       1       1       2       3       5
       8      13      21      34      55
      89     144     233     377     610
     987    1597    2584    4181    6765
   10946   17711   28657   46368   75025
  121393  196418  317811  514229  832040
 1346269 2178309 3524578 5702887 9227465

35 번째 피보나치 수열 계산 시간: 0.002000 초

C:\Users\click\OneDrive\문서\cClickseo\x64\Debug\
이 창을 닫으려면 아무 키나 누르세요...
```

```
Fibonacci(n)
{
   first ← second ← 1;
   for i ← 3 to n
      res ← first + second;
   return res;
}
```

2차원 배열과 동적 메모리 할당

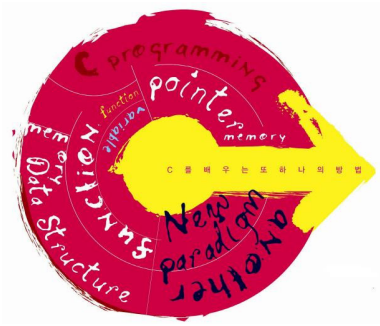
BY NC ND

백문이불여일타(百聞而不如一打)

- 동적 메모리 할당 함수

- **2차원 배열과 동적 메모리 할당**

  - main 함수: 명령행 인자



2차원 배열과 동적 메모리 할당

예제 12-3: 2차원 배열과 동적 메모리 할당

```c
#include <stdio.h>
#include <stdlib.h>   // calloc, free

#define ROW     5       // 행
#define COL     3       // 열

int main(void)
{
        // int    table[ROW][COL];
        int       **pTable = NULL;

        // 동적 메모리 할당: 2차원 동적 배열
        pTable = (int **)calloc(ROW+1, sizeof(int *));
        for(int i=0; i<ROW; i++)
                pTable[i] = (int*)calloc(COL, sizeof(int));

        for(int i=0; i<ROW; ++i)      {
                for(int j=0; j<COL; ++j)
                        printf("%3d", pTable[i][j]+i);
                printf("\n");
        }

        // 메모리 할당 해제
        for(int i=0; i<ROW; ++i)
                free(pTable[i]);
        free(pTable);
        return 0;
}
```

Microsoft Visual Studio 디버그

```
  0  0  0
  1  1  1
  2  2  2
  3  3  3
  4  4  4

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

CC BY NC ND

2차원 배열과 동적 메모리 할당

main 함수: 명령행 인자

main 함수: 명령행 인자 (1/3)

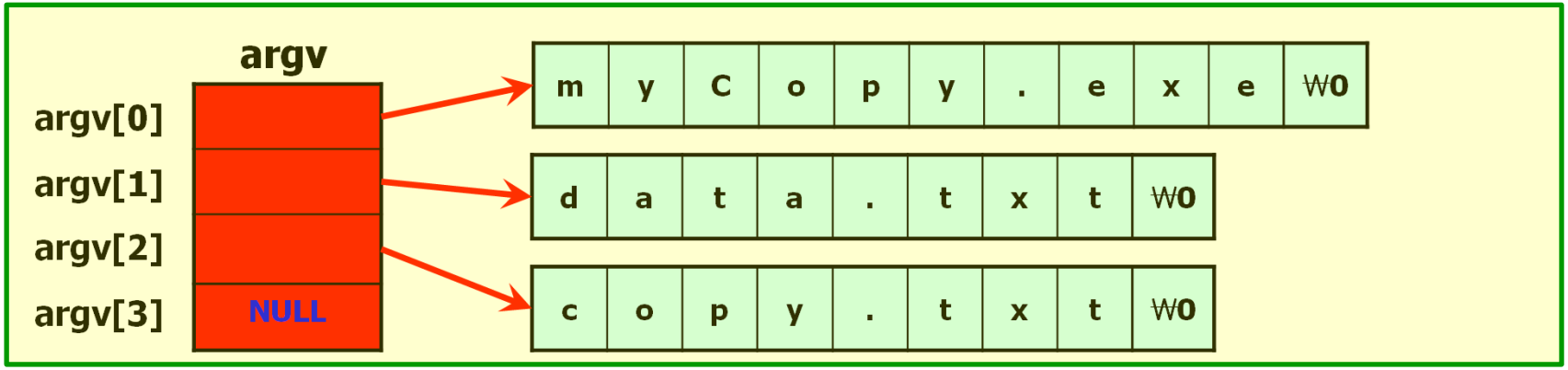
● 명령행 인자

```
int  main(int  argc, char *argv[], char **env)
int  main(int  argc, char **argv, char **env)
```

- **argc** : 명령행 인자의 개수
- ****argv** : 명령행 인자(문자열)가 있는 메모리의 시작 주소
- ****env** : 현재 시스템에 설정되어 있는 환경 변수

argc | **3**

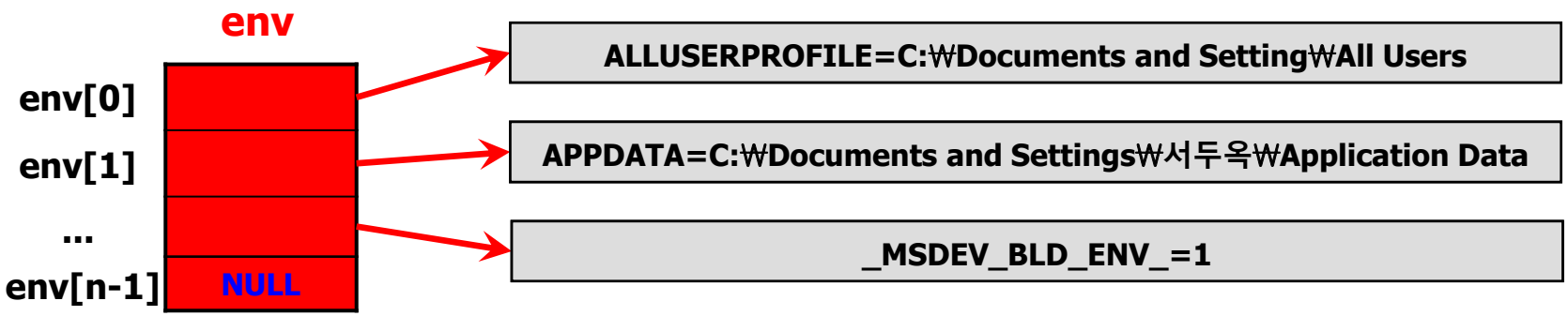
```
C:₩> myCopy  data.txt  copy.txt
```



main 함수: 명령행 인자 (2/3)

- **명령행 인자:** 환경 변수
  - ****env** : 현재 시스템에 설정되어 있는 환경 출력

```
#include <stdio.h>
int main(int argc, char **argv, char **env)
{
        while(*env)
                puts(*env++);

        return 0;
}
```



main 함수: 명령행 인자 (3/3)

예제 12-4: 명령행 인자 -- 사칙 연산

```
#include <stdio.h>
#include <stdlib.h>   // exit, atoi

// int main(int argc, char **argv)
int main(int argc, char *argv[])
{
        if (argc != 4)     {
                printf("프로그램 사용법: Clickseo 10 + 20 \n");
                exit(1);
        }

        char    op;
        int     a, b, res;
        a = atoi(argv[1]);
        b = atoi(argv[3]);
        op = argv[2][0];
        if(b==0)        {
                printf("Error: 나눗셈 입력 오류!!! \n");
                printf("\t 0 으로 나눌 수 없습니다. \n");
                exit(1);
        }
        switch(op)      {
                case '+': res = a + b; break;
                case '-': res = a - b; break;
                case '*': res = a * b; break;
                case '/': res = a / b; break;
                default: printf("지원하지 않는 연산자 입니다!!! \n");
                         exit(1);
        }
        printf("계산 결과: %d %c %d = %d \n", a, op, b, res);
        return 0;
}
```

Microsoft Visual Studio 디버그

```
프로그램 사용법: Clickseo 10 + 20

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

참고문헌

[1] 서현우, “혼자 공부하는 C 언어: 1:1 과외 하듯 배우는 프로그래밍 자습서”, 한빛미디어, 2023.

[2] Paul Deitel, Harvey Deitel, "C How to Program", Global Edition, 8/E, Pearson, 2016.

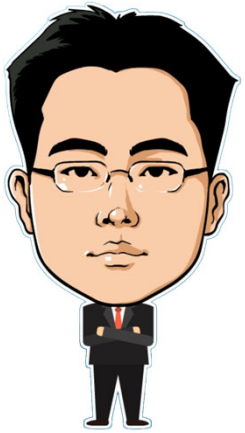
[3] Kamran Amini, 박지윤 번역, “전문가를 위한 C : 동시성, OOP부터 최신 C, 고급 기능까지!”, 한빛미디어, 2022.

[4] 서두옥, “(열혈강의) 또 하나의 C : 프로그래밍은 셀프입니다”, 프리렉, 2012.

[5] Behrouz A. Forouzan, Richard F. Gilberg, 김진 외 7인 공역, “구조적 프로그래밍 기법을 위한 C”, 도서출판 인터비젼, 2004.

[6] Brian W. Kernighan, Dennis M. Ritchie, 김석환 외 2인 공역, “The C Programming Language”, 2/E, 대영사, 2004.

[7] "C reference", cppreference.com, 2024 of viewing the site, https://en.cppreference.com/w/c.



이 강의자료는 저작권법에 따라 보호받는 저작물이므로 무단 전제와 무단 복제를 금지하며,
내용의 전부 또는 일부를 이용하려면 반드시 저작권자의 서면 동의를 받아야 합니다.

Copyright © Clickseo.com. All rights reserved.