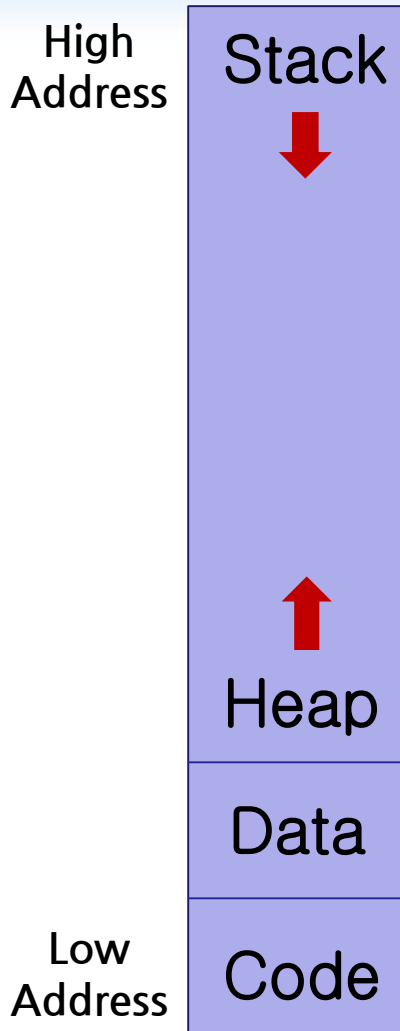


메모리 구조



- ▶ **Stack - 지역 변수 및 매개변수가 저장되는 영역**
 - 함수가 호출되면 메모리에 생성되고, 함수가 종료되면 메모리에서 삭제됨.
 - 프로그램 작성 시 모두 결정되고, 프로그램 운용중에는 이용 할 수 없다.
- ▶ **Heap - 프로그램 운용 중에 사용자의 필요에 의해서 이용할 수 있는 영역**
 - 동적할당 처리 과정을 통해 생성 된다.
 - 변수의 이름이 아니라 주소로 접근해야 하므로 포인터가 사용된다.
 - 프로그램 종료 전 또는 함수가 종료 되어도 영역이 사라지지 않는다.
그러므로, 사용하지 않는 부분은 영역을 해제(삭제) 처리를 해주어야 한다.
- ▶ **Data - 전역 변수 및 정적변수가 저장되는 영역**
 - 프로그램이 종료되기 전까지 유지된다.
- ▶ **Code - 프로그래머가 작성한 언어들을 기계어로 변환하여 보유하고 있는 영역**

동적 할당

❖ 동적 할당이란?

- 프로그램 운용 중 추가로 만들어 저장해야 하는 변수(데이터)가 발생하는 경우, Heap 메모리에 필요한 영역을 만들어 데이터를 저장하고 만들어진 영역은 이름이 아닌 주소를 통해 접근하는 처리 과정
- 프로그램이 종료되기 전까지는 사라지지 않고 영역 및 데이터를 계속 유지하는 특징이 있다. 따라서 사용을 완료한 영역은 반드시 해제 처리를 해야 한다.
- malloc()함수를 통해 생성하고, 포인터로 접근하며, free()함수로 해제한다.

동적 할당 - malloc()함수와 free()함수

❖ malloc() 함수

- 동적할당을 생성하는 함수
- void 포인터라는 자료형을 리턴하므로, 사용하려는 자료형으로 형변환하여 사용한다.
- Header File : <stdlib.h> 또는 <malloc.h>
- 형식

(자료형*) malloc(sizeof(자료형));

자료형 형변환

동적할당 자료형의 크기

❖ free() 함수

- 동적할당을 해제하는 함수
- Header File : <stdlib.h> 또는 <malloc.h>

free(해제할 영역의 주소);

C 동적 할당 - 자료형에 따른 malloc() 의 사용

int 변수 공간 필요 시

```
int *p = (int*)malloc( sizeof(int) );
```

double 변수 공간 필요 시

```
double *p = (double*)malloc(sizeof(double));
```

char 변수 공간 필요 시

```
char *p = (char*)malloc(sizeof(char));
```

문자열 변수 공간 필요 시

```
char *p =  
    (char*)malloc( sizeof(char)*(strlen( " 문자열 " )+1) );
```

구조체 변수 공간 필요 시

```
구조체 *p = (구조체*)malloc(sizeof(구조체));
```

예제 : 각 자료형의 동적할당

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct peo{
    char  name[20];
    int   age;
}PE0;
```

```
void  main() {
    int   *pa = (int*)malloc(sizeof(int));
    float *pb = (float*)malloc(sizeof(float));
    double *pc = (double*)malloc(sizeof(double));
    char  *pd = (char*)malloc(sizeof(char));
    char *pt =
        (char*)malloc((strlen( " 가나다라 " )+1)
                        *sizeof(char));

    PE0 *pp = (PE0*)malloc(sizeof(PE0));

    *pa = 10;      *pb = 1.11;
    *pc = 8.88;    *pd = ' a ' ;
                    - 뒷면 계속 -
```

예제 : 각 자료형의 동적할당

```
strcpy(pt, "가나다라");
strcpy(pp->name, "구조체이름");
pp->age = 100;

printf(" *pa = %d \n", *pa );
printf(" *pb = %f \n", *pb );
printf(" *pc = %f \n", *pc );
printf(" *pd = %c \n", *pd );
printf(" *pt = %s \n", pt);
printf("이름 : %s, 나이 : %d \n", pp->name,
                                           pp->age);

free(pa);
free(pb);
free(pc);
free(pd);
free(pt);
free(pp);
}
```