

Java를 알고 C배우기

컴퓨터프로그래밍3

week 6-2 포인터와 배열-포인터 증감연산

2022.1학기
충남대 조은선

포인터의 증감 연산

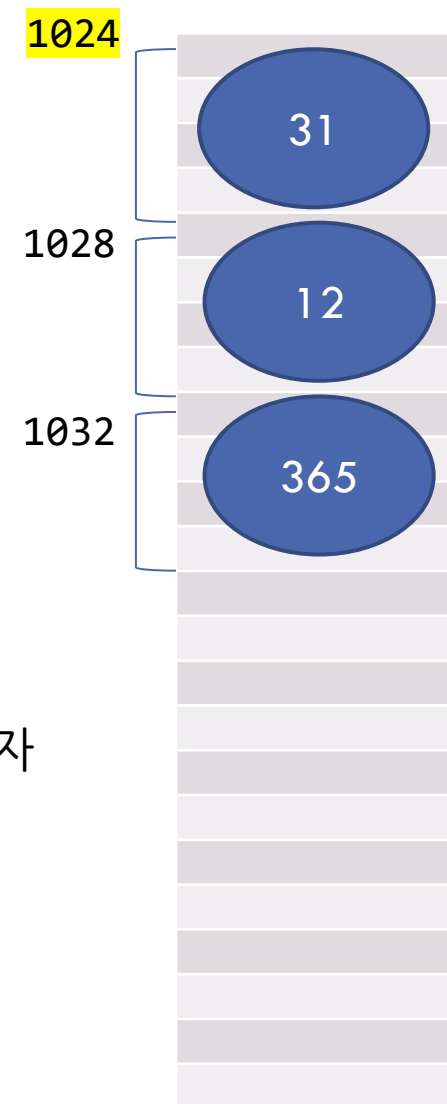
- ▶ 주소 1024를 담고 있는 포인터 변수에 1을 더하면 얼마가 될까?

```
int num = 31;           // 변수 num이 1024번지에 있다고 가정하자
int * p = &num;         // p는 1024를 가진다
printf("%p", p);        // 1024를 출력
printf("%p", p+1);      // 1025를 출력? (X)
```

다른 예

```
int arr[] = {31, 12, 365}; // 배열 arr이 1024번지에 있다고 가정하자
int * p = &arr[0];         // int *p = arr; 도 동일, p는 1024
printf("%p", p);          // 1024를 출력
printf("%p", p+1);        // 1025를 출력? (X)
```

Q) 그럼 무엇을 출력할까?

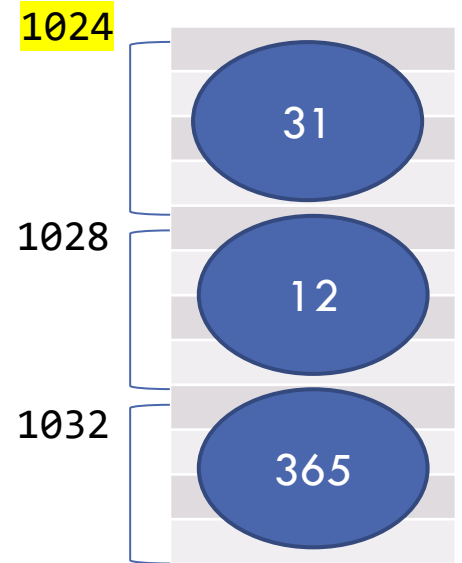


포인터의 증감 연산 (계속)

- ▶ int 포인터는 +1을 하면 sizeof(int) 즉 4만큼 증가한다.

```
int arr[] = {31, 12, 365}; // 배열 arr이 1024번지에 있다고 가정
int * p = &arr[0];          // int * p = arr; p는 1024를 가짐
```

```
printf("%p", p+1);           // 1028를 출력!
printf("%p", p+2);           // 1032를 출력!
printf("%p", p-1);           // 1020을 출력!
```



- ▶ 만일 변수의 타입이 double과 double*이라면? 8씩 증감

```
double darr[] = {3.14, 2.5, 0.1} // 배열 darr이 1024에 있다고 가정
double * p = darr;
printf("%p, %p, %p, %p", p, p+1, p+2, p-1);
// 1024, 1032, 1040, 1016 출력!
```

포인터의 증감연산과 *연산자

실행 결과는?

배열 arr이 1024번지에서 시작한다면

```
int main(void)
{
    int arr[3]={11, 22, 33};
    int * ptr=arr;    // int * ptr=&arr[0]; 과 같은 문장
    printf("%d %d %d \n", *ptr, *(ptr+1), *(ptr+2));

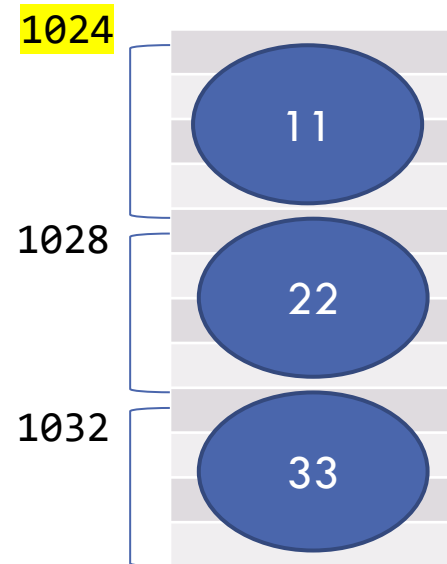
    printf("%d ", *ptr); ptr++;    // printf 함수호출 후, ptr++ 실행
    printf("%d ", *ptr); ptr++;
    printf("%d ", *ptr); ptr--;    // printf 함수호출 후, ptr-- 실행
    printf("%d ", *ptr); ptr--;
    printf("%d ", *ptr); printf("\n");
    return 0;
}
```

arr의 값은 1024, ptr도 1024

*ptr는 11
ptr+1은 1028
*(ptr+1)은?

결과

```
11 22 33
11 22 33 22 11
```



그래서, `*(arr+i)`는 와 똑같다

▶ `*(arr+i)`는 **`arr[i]`**와 똑같다!

```
int main(void)
{
    int arr[3]={11, 22, 33};
    int * ptr=arr;
    printf("%d %d %d \n", *ptr, *(ptr+1), *(ptr+2));
    . . .
}
```

배열이름도 포인터이니, 포인터 변수를 이용한 배열의 접근방식을 배열의 이름에도 사용할 수 있다. 그리고 배열의 이름을 이용한 접근방식도 포인터 변수를 대상으로 사용할 수 있다. 결론은 **`arr`이 포인터 변수의 이름이건 배열의 이름이건**

`arr[i] == *(arr+i)`

```
printf("%d %d %d \n", *(ptr+0), *(ptr+1), *(ptr+2));    // *(ptr+0)는 *ptr과 같다.
printf("%d %d %d \n", ptr[0], ptr[1], ptr[2]);
printf("%d %d %d \n", *(arr+0), *(arr+1), *(arr+2));    // *(arr+0)는 *arr과 같다.
printf("%d %d %d \n", arr[0], arr[1], arr[2]);
```



Quiz

- ▶ 배열 arr의 첫번째 요소 주소가 1000번지라고 할 때 다음 출력값을 적어보시오.

```
int arr[] = {10, 20, 30};  
int * p = arr;  
int * q = p + 1;
```

```
printf("%p", p);  
printf("%p", q-1);  
printf("%p", arr);  
printf("%p", &arr[0]+1);  
printf("%d", *(q-1));  
printf("%d", p[1]);  
printf("%d", a[1]);  
printf("%d", *(arr+1));
```



많이 돌리고 짚어보세요!