qsort()

2017. Fall

국민대학교 컴퓨터공학부 최준수

qsort()

qsort() :

- implementation of quick sorting algorithm
- defined in C standard library, "stdlib.h"
- generic function that can sort arrays of any size, containing any kind of objects and using any kind of comparison predicate
- if objects are not the same in size, pointers have to be used

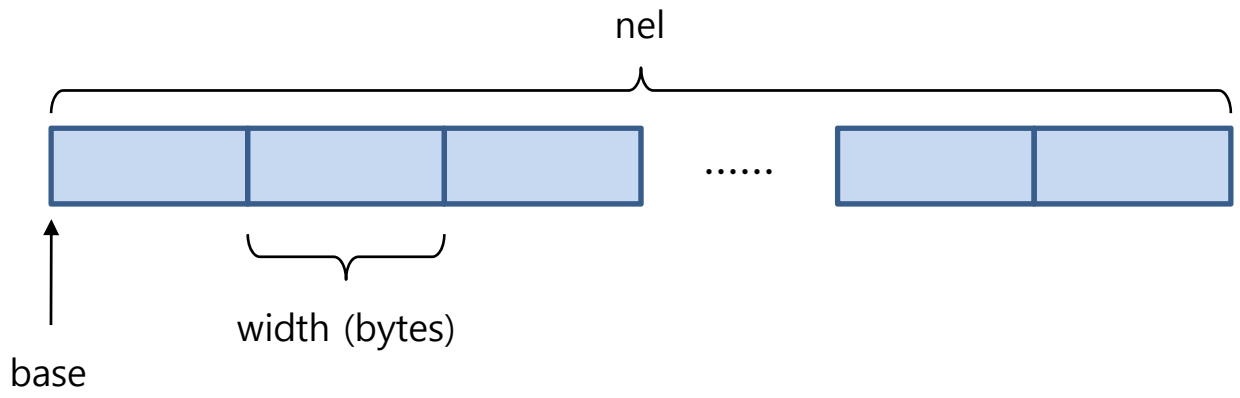
```
#include <stdlib.h>

void qsort( void *base, size_t nel, size_t width,
            int (*compare)(const void *, const void *));
```

qsort()

```
void qsort( void *base, size_t nel, size_t width,
            int (*compare)(const void *, const void *));
```

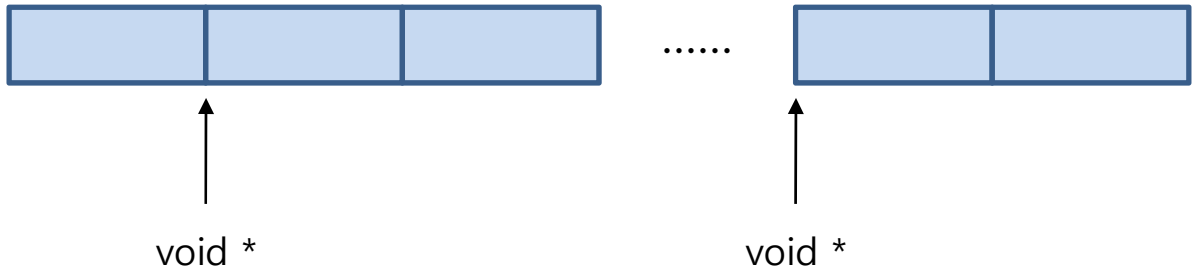
- base : 정렬될 자료가 저장된 1차원 배열의 시작 주소
- nel : 배열에 저장된 자료의 개수
- width : 배열에 저장된 한 개의 자료가 차지하는 메모리의 크기 (바이트 수)
- compare : 배열에 저장된 두 개의 자료의 크기를 비교하는 함수 포인터



qsort()

```
int (*compare)(const void *, const void *)
```

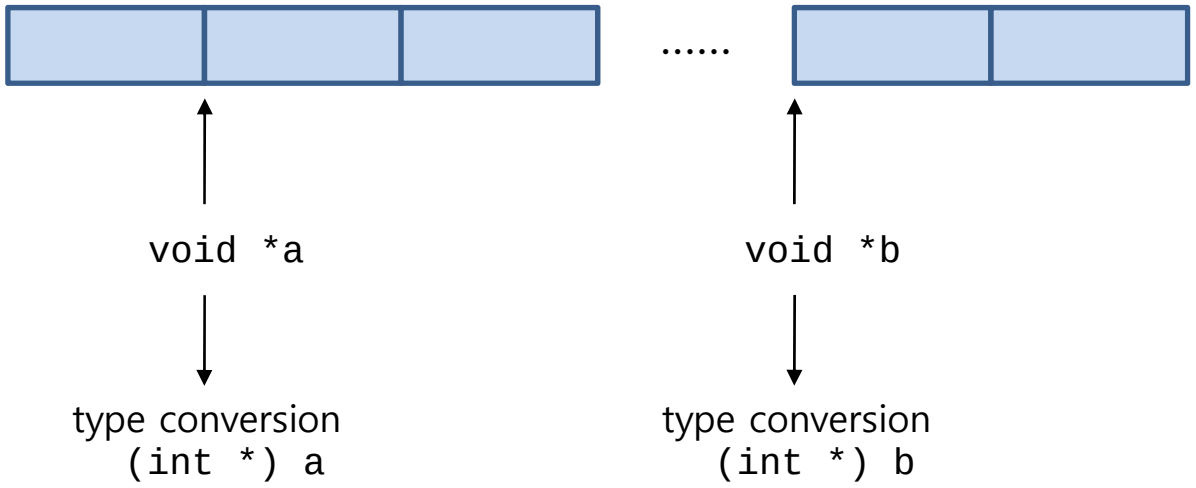
배열에 저장된 두 개의 자료의 크기를 비교하는 함수 포인터
매개변수 : 크기를 비교할 두 자료의 주소 (void *)
Return value:
음수 : 첫 번째 자료가 두 번째 자료보다 작은 경우
0 : 첫 번째 자료가 두 번째 자료와 같은 경우
양수 : 첫 번째 자료가 두 번째 자료보다 큰 경우



qsort()

qsort() 를 사용하는 예 : 정수 배열을 정렬하는 프로그램

```
int icompare(const void *a, const void *b)
{
        return *(int *)a - *(int *)b;
}
```



qsort()

qsort() 를 사용하는 예 : 정수 배열을 정렬하는 프로그램

```
int icompare(const void *a, const void *b)
{
        return *(int *)a - *(int *)b;
}
```

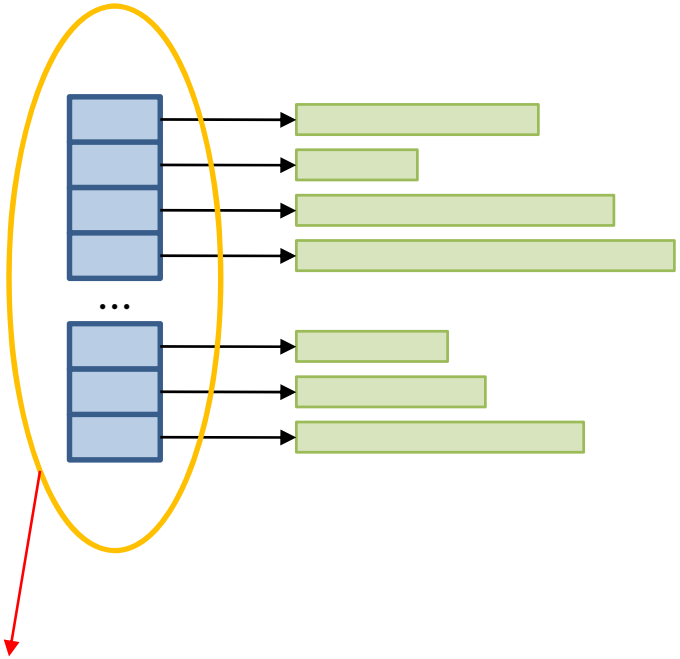
```
void main()
{
        int i;
        int ints[]={6, 9, 5, 1, 3, 2, 4, 8, 7};

        qsort(ints, sizeof(ints)/sizeof(ints[0]), sizeof(int), icompare);

        for (i=0; i<sizeof(ints)/sizeof(ints[0]); i++)
        {
                printf("%d\n", ints[i]);
        }
}
```

qsort()

qsort() 를 사용하는 예 :
크기가 다른 자료를 정렬하는 경우에는 어떻게 하여야 하나?



qsort() 에서 정렬할 배열 (실제 자료의 포인터를 원소로 하는 배열)