memcpy memmove区别和实现

memcpy与memmove的目的都是将N个字节的源内存地址的内容拷贝到目标内存地址中。

但当源内存和目标内存存在重叠时，memcpy会出现错误，而memmove能正确地实施拷贝，但这也增加了一点点开销。

memmove的处理措施：

（1）当源内存的首地址等于目标内存的首地址时，不进行任何拷贝

（2）当源内存的首地址大于目标内存的首地址时，实行正向拷贝

（3）当源内存的首地址小于目标内存的首地址时，实行反向拷贝

-- memcpy实现

```
void* memcpy(void* dest, const void* src, size_t n)
{
    char*       d = (char*) dest;
    const char*  s = (const char*) src;
    while(n--)
       *d++ = *s++;
    return dest;
}
```

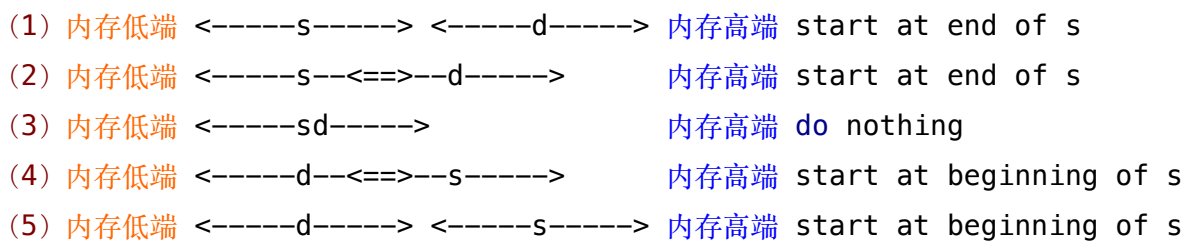
-- memmove实现

```
void* memmove(void* dest, const void* src, size_t n)
{
    char*     d  = (char*) dest;
    const char*  s = (const char*) src;

    if (s>d)
    {
         // start at beginning of s
         while (n--)
            *d++ = *s++;
    }
    else if (s<d)
    {
        // start at end of s
        d = d+n-1;
        s = s+n-1;

        while (n--)
           *d-- = *s--;
    }
    return dest;
}
```

示意图：


（1）内存低端 <-----s-----> <-----d-----> 内存高端 start at end of s
（2）内存低端 <-----s--<==>--d-----> 内存高端 start at end of s
（3）内存低端 <-----sd-----> 内存高端 do nothing
（4）内存低端 <-----d--<==>--s-----> 内存高端 start at beginning of s
（5）内存低端 <-----d-----> <-----s-----> 内存高端 start at beginning of s