

## 实验二：字符串函数的设计

### 一、实验目标

1. 第二章-字符串
2. 掌握常用的字符串函数的设计；
3. 理解字符串处理函数中存在的安全缺陷。

### 二、内容与要求

1. 所在模块：StrSafe

2. 编写函数：**int gets\_safe( char \*str, rsize\_t n );**

功能：从 stdin 读入用户输入，直到用户敲入 Enter 终止。

(1) 如果用户输入的字符个数大于等于 n，则对输入字符串进行截断后保存前(n-1)个字符，最后一个位置置为空字符，以保证 str 指向的字符串符合 C 语言定义。同时，要求对 stdin 中缓冲的多余字符进行清空。

(2) 如果输入的字符个数小于 n，则保存所有字符，最后附加空字符，以保证 str 指向的字符串符合 C 语言定义。

(3) 本函数不保存回车换行符。

(4) 如果读取中出现错误（getchar 返回错误），则 str[0] 置零（如果确定 str 非空），返回-2。

(5) 如果 str 为空指针，返回-1.

参数 str: 指向本地缓冲区的指针，用户输入的数据将被保存在该缓存区内，str 应非空。

参数 n: 本地缓冲区的大小，n 等于 0 时，只是将 str[0] 置零。

返回值：读入的字符个数，不包括空终止符。出现读取错误，返回-2，参数错误返回-1.

提示：使用 getchar()获取用户输入的单个字符，ferror 函数测试读取错误。

3. 编写函数：**int strcpy\_safe(char \* dest, rsize\_t destsz, const char \* src);**

功能：复制源字符串 src 到目的缓冲区 dest。

(1) 如果 src 字符串长度大于等于 destsz，则只复制前 (destsz-1) 个字符，目的缓冲区最后一个字符置零，保证 dest 字符串符合 C 语言对字符串的要求。

(2) 如果 **src** 字符串长度小于 **destsz**，则复制整个源字符串，最后附加空字符，以保证 **dest** 字符串符合 C 语言对字符串的要求。

(3) 如果 **dest** 为空指针，返回-1.

参数 **dest**: 指向目的缓冲区的指针，**dest** 应该为非空指针。

参数 **destsz**: 目的缓冲区的大小，**destsz** 等于 0，函数不执行复制操作，函数返回 0。

参数 **src**: 指向源字符串的指针，如果 **src** 为空指针，函数返回 0。

返回值: 实际复制的字符个数，不包括空终止符。出现错误返回-1.

提示: 使用指针实现单个字符逐一复制，源字符串要符合 C 语言字符串定义规则，如果给入字符串不是以空字符终止，将出现越界复制的情况，直到达到(**destsz**-1)个数为止。

4. 在 **main** 函数中对所编写函数进行测试，说明所使用的测试用例。

5. MSDN: <https://msdn.microsoft.com/en-us/library/hh875057.aspx>

6. MSDN: 微软提供的 **Strsafe** 函数，

<https://docs.microsoft.com/en-us/windows/desktop/menurc/strsafe-ovw>

7. 建议学时: 2 学时。

### 三、参考测试用例

序号	<b>int gets_safe( char *str, rsize_t n );</b>
1	<b>Char buf[8];</b> <b>gets_safe( buf, sizeof(buf));</b> Stdin: “123456”, “1234567”, “12345678”, “\n”
2	<b>Char* buf = 0;</b> <b>gets_safe(buf, sizeof(buf));</b>
3	<b>Char* buf[8];</b> <b>gets_safe(buf, 0);</b>

序号	<b>int strcpy_safe(char * dest, rsize_t destsz, const char * src);</b>
1	<b>Char dest[8];</b> <b>Char *src = “12345”;</b> <b>strcpy_safe( dest, sizeof(dest), src );</b>

2	Char dest[8]; Char *src = “1234567”; strcpy_safe( dest, sizeof(dest), src );
3	Char dest[8]; Char *src = “12345678”; strcpy_safe( dest, sizeof(dest), src );
4	Char *src = “1234567”; strcpy_safe( 0, 8, src );
5	Char dest[8]; strcpy_safe( dest, sizeof(dest), 0);
6	Char dest[8]; Char *src = “12345678”; strcpy_safe( dest, 0, src );

## 附 2：实验报告模板

实验名称	实验一：字符串函数的使用
<p>报告内容应包括：</p> <ol style="list-style-type: none"><li>1. 攻击面分析：所写函数可能会碰到的异常场景</li><li>2. 设计思路：理清函数处理流程，可以用流程图或列表形式</li><li>3. 心得体会</li></ol>	