

# 字符输入输出

# putchar

- `int putchar(int c);`
- 向标准输出写一个字符
- 返回写了几个字符，EOF (-1) 表示写失败

# getchar

- `int getchar(void);`
- 从标准输入读入一个字符
- 返回类型是int是为了返回EOF (-1)
  - Windows—>Ctrl-Z
  - Unix—>Ctrl-D

# 标准库中的字符串函数

# string.h

- strlen
- strcmp
- strcpy
- strcat
- strchr
- strstr

# strlen

- `size_t strlen(const char *s);`
- 返回s的字符串长度（不包括结尾的0）

# strcmp

- `int strcmp(const char *s1, const char *s2);`
- 比较两个字符串，返回：
  - `0:s1==s2`
  - `>0:s1>s2`
  - `<0:s1<s2`

# strcpy

- `char * strcpy(char *restrict dst, const char *restrict src);`
- 把src的字符串拷贝到dst
  - restrict表明src和dst不重叠（C99）
- 返回dst
  - 为了能链起代码来



# 复制一个字符串

```
char *dst = (char*)malloc(strlen(src)+1);  
strcpy(dst, src);
```

# strcat

- `char * strcat(char *restrict s1, const char *restrict s2);`
- 把s2拷贝到s1的后面，接成一个长的字符串
- 返回s1
- s1必须具有足够的空间

# 安全问题

- strcpy和strcat都可能出现安全问题
  - 如果目的地没有足够的空间？

# 安全版本

- `char * strncpy(char *restrict dst, const char *restrict src, size_t n);`
- `char * strncat(char *restrict s1, const char *restrict s2, size_t n);`
- `int strncmp(const char *s1, const char *s2, size_t n);`

# 字符串中找字符

- `char * strchr(const char *s, int c);`
- `char * strrchr(const char *s, int c);`
- 返回NULL表示没有找到
- 如何寻找第2个?

# 字符串中找字符串

- `char * strstr(const char *s1, const char *s2);`
- `char * strcasestr(const char *s1, const char *s2);`