1. 系统调用和库函数

系统调用：由系统直接提供的函数接口

库函数：由开发人员提供的函数接口

1. 标准IO和文件IO

标准IO：由库函数提供的函数接口

优点：减少对硬件的频繁操作，有利于保护硬件，适应性更强，

缺点：缓存区的数据可能丢失，开发人员写的库源码有问题，影响程序员的操作

文件IO：由系统调用提供的函数接口

优点：数据实时写入硬件，代码稳定

缺点：频繁操作硬件，不利于保护硬件

1. 标准IO缓存区

全缓存：默认的情况是全缓存

刷新条件：

1. 程序正常结束
2. 缓存区满了
3. fflush

行缓存：默认和终端有关系的缓存

刷新条件

1. 字符’\n’
2. 程序正常结束
3. 缓存区满了
4. fflush

无缓存

stderr

1. 流：文件指针
   1. FILE指针：每个被使用的文件都在内存中开辟一个区域，用来存放文件的有关信息，这些信息是保存在一个结构体类型的变量中，该结构体类型是由系统定义的，取名为FILE。
   2. 标准I/O库的所有操作都是围绕流(stream)来进行的，在标准I/O中，流用FILE \*来描述。
   3. 标准I/O库是由Dennis Ritchie在1975年左右编写的
2. ctags

1．cd /usr/include

2.sudo ctags –R

3.检测vi –t “FILE”

追加 ctrl + 】

返回上有一层 ctrl + t

1. 获得流

#include <stdio.h>

FILE \*fopen(const char \*path, const char \*mode);

功能：获得流

参数：path 文件的路径

mode 打开方式

r：只读的方式打开，要求文件必须存在

r+：可读可写的方式打开，要求文件必须存在

w：只写的方式的打开，如果文件存在清空，如果不存在创建

w+：可读可写的方式打开，如果文件存在清空，如果不存在创建

a:追加的方式打开如果不存在创建

a+：可读可写的追加方式，如果不存在创建

返回值：成功：返回流，失败NULL

默认打开三个流：标准输入 标准输出 标准出错

stdin stdout stderr

1. fclose

#include <stdio.h>

int fclose(FILE \*fp);

功能：关闭一个流

参数：fp 要关闭的流

返回值：成功0 失败EOF

注意：不要多重释放

1. 每次一个字符的读写

getchar

putchar

#include <stdio.h>

int fgetc(FILE \*stream);

同下

int getc(FILE \*stream);

功能：从流中读入一个字符

参数：流

返回值：成功读到的字符 失败EOF

getc（宏） fgetc（函数）

int getchar(void);

#include <stdio.h>

int fputc(int c, FILE \*stream);

功能：输出一个字符

参数：c 要输出的字符

stream 流

返回值：成功输出的值 失败EOF

int putc(int c, FILE \*stream);

同上

1. 每次一行字符的读写

#include <stdio.h>

char \*fgets(char \*s, int size, FILE \*stream);

功能：从流中读入一行字符(遇到‘\n’读结束)

参数：s 读到数据的存放地址

size 期望值

stream 流

返回值：成功s 失败NULL

注意：一次最多读取size -1 由于结尾自动补‘\0’

#include <stdio.h>

int fputs(const char \*s, FILE \*stream);

功能：输出一行字符(第一个’\0’,之前的内容)

参数：s 要输出的字符首地址，

stream 流

返回值：成功：成功写入的个数，失败EOF

1. 每次按记录的读写

#include <stdio.h>

size\_t fread(void \*ptr, size\_t size, size\_t nmemb, FILE \*stream);

功能：读记录

参数：ptr 读到数据存放位置

size 一条记录的大小

nmemb 期望读多少条

stream 流

返回值：成功读到记录的条数

size\_t fwrite(const void \*ptr, size\_t size, size\_t nmemb,FILE \*stream);

功能：向流中写入几条记录

参数：ptr 要写入的内容

size 一条记录的大小

nmemb 记录的条数

stream 流

返回值：实际写入的记录条数

【11】

#include <stdio.h>

void clearerr(FILE \*stream);

功能：清空错误状态

参数：流

int feof(FILE \*stream);

功能：判断是否到文件结尾

参数：流

返回值：到结尾为真，没有到为假

int ferror(FILE \*stream);

功能：判断流是否出错

参数：流

返回值：为真表示出错，为假表示正常