

GUN/Linux核心书系

Linux

函数库参考手册

徐千洋 著



- ◆ 超过 400 个常用函数的详细说明
- ◆ 依函数用途分类介绍，方便查询
- ◆ gcc 常用函数介绍
- ◆ 函数与宏的差异
- ◆ 常见错误原因列表




中国青年出版社

旗标



旗标出版股份有限公司

 GUN/Linux核心书系

 旗标丛书

Linux C

函数库参考手册

徐千洋 著



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书由旗标出版股份有限公司授权中国青年出版社独家出版。未经出版者书面许可,任何单位和个人均不得以任何形式复制或传播本书的部分或全部。

版权贸易合同登记号: 01-2001-4223

图书在版编目(CIP)数据

Linux C 函数库参考手册 / 徐千洋著. - 北京: 中国青年出版社, 2002

ISBN 7-5006-4653-4

I. L... II. 徐... III. C 语言 - 程序设计 - 手册 IV. TP312.62

中国版本图书馆 CIP 数据核字(2001)第 092822 号

策 划: 胡守文

王修文

郭 光

责任编辑: 吴 源

黄 谊

责任校对: 肖新民

丛 书 名: GUN/Linux 核心书系

书 名: **Linux C 函数库参考手册**

著 者: 徐千洋

出版发行: 中国青年出版社

地址: 北京市东四 12 条 21 号 邮政编码: 100708

电话: (010) 84015588 传真: (010) 64053266

印 刷: 北京机工印刷厂

开 本: 787 × 1092 1/16 印张: 26 字数: 497 千字

版 次: 2002 年 1 月北京第 1 版

印 次: 2002 年 1 月第 1 次印刷

印 数: 1-5000

书 号: ISBN 7-5006-4653-4/TP · 243

定 价: 39.00 元

目 录

第 1 章 字符测试函数

isalnum (测试字符是否为英文字母或数字) ...	2
isalpha (测试字符是否为英文字母)	3
isascii (测试字符是否为 ASCII 码字符)	4
isblank (测试字符是否为空格字符)	5
iscntrl (测试字符是否为 ASCII 码的控制 字符)	5
isdigit (测试字符是否为阿拉伯数字)	6
isgraph (测试字符是否为可打印字符)	7
islower (测试字符是否为小写英文字母)	8
isprint (测试字符是否为可打印字符)	8
isspace (测试字符是否为空格字符)	9
ispunct (测试字符是否为标点符号或特殊 符号)	10
isupper (测试字符是否为大写英文字母)	11
isxdigit (测试字符是否为 16 进制数字)	12

第 2 章 数据转换函数

atof (将字符串转换成浮点型数)	15
atoi (将字符串转换成整型数)	15
atol (将字符串转换成长整型数)	16
ecvt (将浮点型数转换成字符串, 取四舍 五入)	17
fcvt (将浮点型数转换为字符串, 取四舍 五入)	18
gcvt (将浮点型数转换为字符串, 取四舍 五入)	19
strtod (将字符串转换成浮点型数)	20
strtol (将字符串转换成长整型数)	21
strtoul (将字符串转换成无符号长整型数) ...	22

toascii (将整型数转换成合法的 ASCII 码 字符)	23
tolower (将大写字母转换成小写字母)	24
toupper (将小写字母转换成大写字母)	25

第 3 章 内存配置函数

alloca (配置内存空间)	27
brk (改变数据字节的范围)	27
calloc (配置内存空间)	27
free (释放原先配置的内存)	28
getpagesize (取得内存分页大小)	28
malloc (配置内存空间)	29
mmap (建立内存映射)	29
munmap (解除内存映射)	32
realloc (更改已配置的内存空间)	33
sbrk (增加程序可用的数据空间)	34

第 4 章 时间函数

asctime (将时间和日期以字符串格式表示) ...	36
clock (取得进程占用 CPU 的大约时间)	36
ctime (将时间和日期以字符串格式表示) ...	37
difftime (计算时间差距)	38
ftime (取得目前的时间和日期)	38
gettimeofday (取得目前的时间)	39
gmtime (取得目前的时间和日期)	41
localtime (取得当地目前的时间和日期)	42
mktime (将时间结构数据转换成经过的秒数) ...	43
settimeofday (设置目前的时间)	44
strftime (格式化日期和时间)	45
time (取得目前的时间)	47
tzset (设置时区以供时间转换)	48

第 5 章 字符串处理函数

bcmp (比较内存内容)	50
bcopy (拷贝内存内容)	50
bzero (将一段内存内容全清为零)	51
ffs (在一整型数中查找第一个值为真的位) ..	52
index (查找字符串中第一个出现的指定字 符)	53
memccpy (拷贝内存内容)	53
memchr (在某一内存范围中查找一特定字 符)	54
memcmp (比较内存内容)	55
memcpy (拷贝内存内容)	56
memfrob (对内存区域编码)	57
memmove (拷贝内存内容)	58
memset (将一段内存空间填入某值)	59
rindex (查找字符串中最后一个出现的指定 字符)	59
strcasemp (忽略大小写比较字符串)	60
strcat (连接两字符串)	61
strchr (查找字符串中第一个出现的指定字 符)	62
strcmp (比较字符串)	63
strcoll (采用目前区域的字符排列次序来比 较字符串)	64
strcpy (拷贝字符串)	64
strcspn (返回字符串中连续不含指定字符串 内容的字符数)	65
strdup (复制字符串)	66
strfry (随机重组字符串内的字符)	67
strlen (返回字符串长度)	68
strncasemp (忽略大小写比较字符串)	68
strncat (连接两字符串)	69
strncmp (比较字符串)	70
strcpy (拷贝字符串)	71
strpbrk (查找字符串中第一个出现的指定字	

符)	71
strrchr (查找字符串中最后一个出现的指定 字符)	72
strspn (返回字符串中连续不含指定字符串 内容的字符数)	73
strstr (在一字符串中查找指定的字符串)	74
strtok (分割字符串)	75

第 6 章 数学计算函数

abs (计算整型数的绝对值)	78
acos (取反余弦函数值)	78
asin (取反正弦函数值)	79
atan (取反正切函数值)	80
atan2 (取得反正切函数值)	81
ceil (取不小于参数的最小整型数)	81
cos (取余弦函数值)	82
cosh (取双曲线余弦函数值)	83
div (取得两整型数相除后的商及余数)	84
exp (计算指数)	85
fabs (计算浮点型数的绝对值)	85
frexp (将浮点型数分为底数与指数)	86
hypot (计算直角三角形斜边长)	87
labs (计算长整型数的绝对值)	88
ldexp (计算 2 的次方值)	89
ldiv (取得两长整数相除后的商及余数)	89
log (计算以 e 为底的对数值)	90
log10 (计算以 10 为底的对数值)	91
modf (将浮点型数分解成整数与小数)	92
pow (计算次方值)	93
sin (取正弦函数值)	94
sinh (取双曲线正弦函数值)	95
sqrt (计算平方根值)	95
tan (取正切函数值)	96
tanh (取双曲线正切函数值)	97

第 7 章 用户和组函数

cuserid (取得用户帐号名称)	99
--------------------------	----

- endgrent (关闭组文件) 99
- endpwent (关闭密码文件) 100
- endutent (关闭 utmp 文件) 100
- fgetgrent (从指定的文件来读取组格式) 101
- fgetpwent (从指定的文件来读取密码格式) ... 103
- getegid (取得有效的组织别码) 104
- geteuid (取得有效的用户识别码) 105
- getgid (取得真实的组织别码) 106
- getgrent (从组文件文件中取得帐号的数
据) 106
- getgrgid (从组文件中取得指定 gid 的数
据) 108
- getgrnam (从组文件中取得指定组的数据) ... 109
- getgroups (取得组代码) 110
- getlogin (取得登录的用户帐号名称) 111
- getpw (取得指定用户的密码文件数据) 112
- getpwent (从密码文件中取得帐号的数据) ... 113
- getpwnam (从密码文件中取得指定帐号的
数据) 115
- getpwuid (从密码文件中取得指定 uid 的
数据) 116
- getuid (取得真实的用户识别码) 117
- getutent (从 utmp 文件中取得帐号登录数
据) 117
- getutid (从 utmp 文件中查找特定的记录) ... 119
- getutline (从 utmp 文件中查找特定的记录) ... 120
- initgroups (初始化组清单) 121
- logwtmp (将一登录数据记录到 wtmp 文
件) 122
- pututline (将 utmp 记录写入文件) 122
- setegid (设置有效的组织别码) 123
- seteuid (设置有效的用户识别码) 124
- setfsgid (设置文件系统的组织别码) 124
- setfsuid (设置文件系统的用户识别码) 125
- setgid (设置真实的组织别码) 125
- setgrent (从头读取组文件中的组数据) 126
- setgroups (设置组代码) 126
- setpwent (从头读取密码文件中的帐号数
据) 127
- setregid (设置真实及有效的组织别码) 128
- setreuid (设置真实及有效的用户识别码) ... 128
- setuid (设置真实的用户识别码) 129
- setutent (从头读取 utmp 文件中的登录数
据) 129
- updwtmp (将一登录数据记录到 wtmp 文
件) 130
- utmpname (设置 utmp 文件路径) 131
- ## 第 8 章 数据加密函数
- crypt (将密码或数据编码) 133
- getpass (取得一密码输入) 134
- ## 第 9 章 数据结构函数
- bsearch (二元搜索) 137
- hcreate (建立哈希表) 139
- hdestroy (删除哈希表) 139
- hsearch (哈希表搜索) 139
- insque (加入一项目至队列中) 140
- lfind (线性搜索) 141
- lsearch (线性搜索) 141
- qsort (利用快速排序法排列数组) 143
- remque (从队列中删除一项目) 144
- tdelete (从二叉树中删除数据) 145
- tfind (搜索二叉树) 145
- tsearch (二叉树) 146
- twalk (走访二叉树) 146
- ## 第 10 章 随机数函数
- drand48 (产生一个正的浮点型随机数) 149
- erand48 (产生一个正的浮点型随机数) 150
- initstate (建立随机数状态数组) 151
- jrand48 (产生一个长整型数随机数) 151
- lcong48 (设置 48 位运算的随机数种子) ... 152

lrand48 (产生一个正的长整型随机数)	154
mrnd48 (产生一个长整型随机数)	155
nrnd48 (产生一个正的长整型随机数)	156
rand (产生随机数)	157
random (产生随机数)	158
seed48 (设置 48 位运算的随机数种子)	158
setstate (建立随机数状态数组)	160
srand (设置随机数种子)	160
srand48 (设置 48 位运算的随机数种子)	161
srandom (设置随机数种子)	161

第 11 章 初级 I/O 函数

close (关闭文件)	164
creat (建立文件)	164
dup (复制文件描述词)	165
dup2 (复制文件描述词)	166
fcntl (文件描述词操作)	166
flock (锁定文件或解除锁定)	168
fsync (将缓冲区数据写回磁盘)	169
lseek (移动文件的读写位置)	169
mkstemp (建立唯一的临时文件)	170
open (打开文件)	171
read (由已打开的文件读取数据)	175
sync (将缓冲区数据写回磁盘)	175
write (将数据写入已打开的文件内)	176

第 12 章 标准 I/O 函数

clearerr (清除文件流的错误旗标)	178
fclose (关闭文件)	178
fdopen (将文件描述词转为文件指针)	178
feof (检查文件流是否读到了文件尾)	179
fflush (更新缓冲区)	180
fgetc (由文件中读取一个字符)	180
fgetpos (取得文件流的读取位置)	181
fgets (由文件中读取一字符串)	181
fileno (返回文件流所使用的文件描述词)	182
fopen (打开文件)	183

fputc (将一指定字符写入文件流中)	184
fputs (将一指定的字符串写入文件内)	185
fread (从文件流读取数据)	186
freopen (打开文件)	187
fseek (移动文件流的读写位置)	187
fsetpos (移动文件流的读写位置)	189
ftell (取得文件流的读取位置)	189
fwrite (将数据写至文件流)	190
getc (由文件中读取一个字符)	191
getchar (由标准输入设备内读进一字符)	192
gets (由标准输入设备内读进一字符串)	192
mktemp (产生唯一的临时文件文件名)	193
putc (将一指定字符写入文件中)	194
putchar (将指定的字符写到标准输出设备)	194
puts (将指定的字符串写到标准输出设备)	195
rewind (重设文件流的读写位置为文件开头)	195
setbuf (设置文件流的缓冲区)	196
setbuffer (设置文件流的缓冲区)	196
setlinebuf (设置文件流为线性缓冲区)	197
setvbuf (设置文件流的缓冲区)	197
tmpfile (建立临时文件)	198
ungetc (将一指定字符写回文件流中)	198

第 13 章 进程及流程控制

abort (以异常方式结束进程)	201
assert (若测试的条件不成立则终止进程)	201
atexit (设置程序正常结束前调用的函数)	202
execl (执行文件)	203
execle (执行文件)	203
execlp (从 PATH 环境变量中查找文件并执行)	204
execv (执行文件)	205
execve (执行文件)	206

execvp (执行文件)	208	vscanf (格式化字符串输入)	239
exit (正常结束进程)	208	vsprintf (格式化字符串复制)	240
_exit (结束进程执行)	209	vsscanf (格式化字符串输入)	241
fork (建立一个新的进程)	209	第 15 章 文件及目录函数	
getpgid (取得进程组识别码)	210	access (判断是否具有存取文件的权限)	243
getpgrp (取得进程组识别码)	211	alphasort (依字母顺序排序目录结构)	244
getpid (取得进程识别码)	212	chdir (改变当前的工作目录)	245
getppid (取得父进程的进程识别码)	213	chmod (改变文件的权限)	246
getpriority (取得程序进程执行优先权)	213	chown (改变文件的所有者)	248
longjmp (跳转到原先 setjmp 保存的堆栈 环境)	214	chroot (改变根目录)	249
nice (改变进程优先顺序)	215	closedir (关闭目录)	250
on_exit (设置程序正常结束前调用的函数)	216	fchdir (改变当前的工作目录)	250
ptrace (进程追踪)	217	fchmod (改变文件的权限)	251
setjmp (保存目前堆栈环境)	219	fchown (改变文件的所有者)	252
setpgid (设置进程组识别码)	220	fstat (由文件描述词取得文件状态)	253
setpgrp (设置进程组识别码)	221	ftruncate (改变文件大小)	254
setpriority (设置程序进程执行优先权)	221	ftw (遍历目录树)	255
siglongjmp (跳转到原先 sigsetjmp 保存的 堆栈环境)	222	get_current_dir_name (取得当前的工作目 录)	256
sigsetjmp (保存目前堆栈环境)	222	getcwd (取得当前的工作目录)	257
system (执行 shell 命令)	223	getwd (取得当前的工作目录)	258
wait (等待子进程中断或结束)	224	lchown (改变文件的所有者)	259
waitpid (等待子进程中断或结束)	225	link (建立文件连接)	260
第 14 章 格式化输入输出函数		lstat (由文件描述词取得文件状态)	261
fprintf (格式化输出数据至文件)	229	nftw (遍历目录树)	261
fscanf (格式化字符串输入)	229	opendir (打开目录)	263
printf (格式化输出数据)	230	readdir (读取目录)	263
scanf (格式化字符串输入)	233	readlink (取得符号连接所指的文件)	265
snprintf (格式化字符串复制)	234	realpath (将相对目录路径转换成绝对路 径)	266
sprintf (格式化字符串复制)	235	remove (删除文件)	267
sscanf (格式化字符串输入)	236	rename (更改文件名称或位置)	267
vfprintf (格式化输出数据至文件)	237	rewinddir (重设读取目录的位置为开头位 置)	268
vscanf (格式化字符串输入)	237		
vprintf (格式化输出数据)	238		

scandir (读取特定的目录数据)	270
seekdir (设置下回读取目录的位置)	272
stat (取得文件状态)	273
symlink (建立文件符号连接)	277
telldir (取得目录流的读取位置)	278
truncate (改变文件大小)	279
umask (设置建立新文件时的权限遮罩)	280
unlink (删除文件)	280
utime (修改文件的存取时间和更改时间)	281
utimes (修改文件的存取时间和更改时间)	281

第 16 章 信号函数

alarm (设置信号传送闹钟)	284
kill (传送信号给指定的进程)	285
pause (让进程暂停直到信号出现)	286
psignal (列出信号描述和指定字符串)	287
raise (传送信号给目前的进程)	288
sigaction (查询或设置信号处理方式)	288
sigaddset (增加一个信号至信号集)	291
sigdelset (从信号集里删除一个信号)	291
sigemptyset (初始化信号集)	292
sigfillset (将所有信号加入至信号集)	292
sigismember (测试某个信号是否已加入至 信号集里)	292
signal (设置信号处理方式)	293
sigpause (暂停直到信号到来)	294
sigpending (查询被搁置的信号)	294
sigprocmask (查询或设置信号遮罩)	294
sigsuspend (暂停直到信号到来)	295
sleep (让进程暂停执行一段时间)	295
isdigit (测试字符是否为阿拉伯数字)	296

第 17 章 错误处理函数

ferror (检查文件流是否有错误发生)	299
perror (打印出错误原因信息字符串)	299
strerror (返回错误原因的描述字符串)	300

第 18 章 管道相关函数

mkfifo (建立具名管道)	303
pclose (关闭管道 I/O)	304
pipe (建立管道)	305
popen (建立管道 I/O)	306

第 19 章 Socket 相关函数

accept (接受 socket 连线)	309
bind (对 socket 定位)	309
connect (建立 socket 连线)	311
endprotoent (结束网络协议数据的读取)	313
endservent (结束网络服务数据的读取)	313
gethostbyaddr (由 IP 地址取得网络数据)	314
gethostbyname (由主机名称取得网络数据)	315
getprotobyname (由网络协议名称取得协议 数据)	317
getprotobynumber (由网络协议编号取得协 议数据)	317
getprotoent (取得网络协议数据)	318
getservbyname (依名称取得网络服务的数 据)	320
getservbyport (依 port 号码取得网络服务的 数据)	321
getservent (取得主机网络服务的数据)	321
getsockopt (取得 socket 状态)	323
herror (打印出网络错误原因信息字符串)	324
hstrerror (返回网络错误原因的描述字符 串)	324
htonl (将 32 位主机字符顺序转换成网络 字符顺序)	325
htons (将 16 位主机字符顺序转换成网络 字符顺序)	326
inet_addr (将网络地址转成网络二进制的数 字)	326
inet_aton (将网络地址转成网络二进制的数 字)	326
inet_ntoa (将网络二进制的数字转换成网络	

地址)	327	shmget (配置共享内存)	360
listen (等待连接)	328	第 21 章 记录函数	
ntohl (将 32 位网络字符顺序转换成主机 字符顺序)	331	closelog (关闭信息记录)	362
ntohs (将 16 位网络字符顺序转换成主机 字符顺序)	331	openlog (准备做信息记录)	362
recv (经 socket 接收数据)	332	syslog (将信息记录至系统日志文件)	363
recvfrom (经 socket 接收数据)	333	第 22 章 环境变量函数	
recvmsg (经 socket 接收数据)	335	getenv (取得环境变量内容)	366
send (经 socket 传送数据)	336	putenv (改变或增加环境变量)	366
sendmsg (经 socket 传送数据)	336	setenv (改变或增加环境变量)	367
sendto (经 socket 传送数据)	338	unsetenv (清除环境变量内容)	368
setprotoent (打开网络协议的数据文件)	340	第 23 章 正则表达式	
setservent (打开主机网络服务的数据文 件)	340	regcomp (编译正则表达式字符串)	371
setsockopt (设置 socket 状态)	340	regerror (取得正则搜索的错误原因)	372
shutdown (终止 socket 通信)	341	regex (进行正则表达式的搜索)	374
socket (建立一个 socket 通信)	342	regfree (释放正则表达式使用的内存)	375
第 20 章 进程通信 (IPC) 函数		第 24 章 动态函数	
ftok (将文件路径和计划代号转为 System V IPC key)	345	dldclose (关闭动态函数库文件)	378
msgctl (控制信息队列的运作)	345	dlderror (动态函数错误处理)	378
msgget (建立信息队列)	348	dlopen (打开动态函数库文件)	379
msgrcv (从信息队列读取信息)	349	dlsym (从共享对象中搜索动态函数)	380
msgsnd (将信息送入信息队列)	350	第 25 章 其他函数	
semctl (控制信号队列的操作)	351	getopt (分析命令行参数)	383
semget (配置信号队列)	353	isatty (判断文件描述词是否是终端机)	384
semop (信号处理)	354	select (I/O 多工机制)	385
shmat (attach 共享内存)	356	ttyname (返回一终端机名称)	386
shmctl (控制共享内存的操作)	357	附录 A 编译程序-gcc	
shmdt (detach 共享内存)	359	附录 B 宏与函数	
		附录 C 不定参数	
		附录 D Linux 信号列表	
		附录 E 常见错误代码及原因	

1

CHAPTER

字符测试函数

isalnum (测试字符是否为英文字母或数字)

相关函数 : isalpha, isdigit, islower, isupper

表头文件 : #include <ctype.h>

定义函数 : int isalnum (int c);

函数说明 : 检查参数 c 是否为英文字母或阿拉伯数字, 在标准 C 中相当于使用 (isalpha (c) || isdigit (c)) 做测试。

返回值 : 若参数 c 为字母或数字, 则返回 TRUE, 否则返回 NULL (0)。

附加说明 : 此为宏定义, 非真正函数。

范 例

```
/* 找出 str 字符串中为英文字母或数字的字符 */
#include <ctype.h>
main ()
{
    char str[]="123c@#FDsP[e?";
    int i;
    for (i = 0; str[i] != 0;i++)
        if (isalnum (str[i]) ) printf ("%c is an alphanumeric character\n", str[i]) ;
}
```

执行结果

```
1 is an alphabetic character
2 is an alphabetic character
3 is an alphabetic character
c is an alphabetic character
F is an alphabetic character
D is an alphabetic character
s is an alphabetic character
P is an alphabetic character
e is an alphabetic character
```

②

isalpha (测试字符是否为英文字母)

相关函数 : isalnum, islower, isupper

表头文件 : #include <ctype.h>

定义函数 : int isalpha (int c);

函数说明 : 检查参数 c 是否为英文字母, 在标准 C 中相当于使用 (isupper(c) || islower(c)) 做测试。

返回值 : 若参数 c 为英文字母, 则返回 TRUE, 否则返回 NULL (0)。

附加说明 : 此为宏定义, 非真正函数。

范 例

```
/* 找出 str 字符串中为英文字母的字符 */
#include <ctype.h>
main ()
{
    char str[]="123c@#FDsP[e?";
    int i;
    for (i = 0; str[i] != 0;i++)
        if (isalpha (str[i]) ) printf ("%c is an alphabetic character\n", str[i]) ;
}
```

执行结果

```
c is an alphabetic character
F is an alphabetic character
D is an alphabetic character
s is an alphabetic character
P is an alphabetic character
e is an alphabetic character
```

①

isascii (测试字符是否为 ASCII 码字符)

相关函数 : iscntrl

表头文件 : #include <ctype.h>

定义函数 : int isascii (int c);

函数说明 : 检查参数 c 是否为 ASCII 码字符, 也就是判断 c 的范围是否在 0 到 127 之间。

返回值 : 若参数 c 为 ASCII 码字符, 则返回 TRUE, 否则返回 NULL (0)。

附加说明 : 此为宏定义, 非真正函数。

范 例

```
/* 判断 int i 是否具有对映的 ASCII 码字符 */
#include <ctype.h>
main ()
{
    int i;
    for (i = 125; i < 130; i++)
        if (isascii (i) )
            printf ("%d is an ascii character : %c\n", i, i) ;
        else
            printf ("%d is not an ascii character\n", i) ;
}
```

执行结果

```
125 is an ascii character : }
126 is an ascii character : ~
127 is an ascii character :
128 is not an ascii character
129 is not an ascii character
```

①

isblank (测试字符是否为空格字符)**相关函数** : isspace**表头文件** : #include <ctype.h>**定义函数** : int isblank (int c);**函数说明** : 检查参数 c 是否为空格字符, 也就是判断是否为空格 (space) 或是定位字符 (tab)。空格 (space) 的 ASCII 码为 32, 定位字符 (tab) 的 ASCII 码则为 9。**返回值** : 若参数 c 为空格字符, 则返回 TRUE, 否则返回 NULL (0)。**附加说明** : 此为宏定义, 非真正函数。**范 例**

```

/* 将字符串 str[] 中内含的空格字符找出, 并显示空格字符的 ASCII 码*/
#include <ctype.h>
main ()
{
    char str[]="123c @# FD      sP[e?";
    int i;
    for (i = 0; str[i] != 0;i++)
        if (isblank (str[i])) printf ("str[%d] is blank character: %d\n", i, str[i]);
}

```

执行结果

```

str[4] is blank character: 32
str[7] is blank character: 32
str[10] is blank character: 9

```

①

isctrl (测试字符是否为 ASCII 码的控制字符)**相关函数** : isascii

表头文件： #include <ctype.h>

定义函数： int iscntrl (int c);

函数说明： 检查参数 c 是否为 ASCII 控制码，也就是判断 c 的范围是否在 0 到 31 之间。

返回值： 若参数 c 为 ASCII 控制码，则返回 TRUE，否则返回 NULL (0)。

附加说明： 此为宏定义，非真正函数。

isdigit (测试字符是否为阿拉伯数字)

相关函数： isxdigit

表头文件： #include <ctype.h>

定义函数： int isdigit (int c);

函数说明： 检查参数 c 是否为阿拉伯数字 0 到 9。

返回值： 若参数 c 为阿拉伯数字，则返回 TRUE，否则返回 NULL (0)。

附加说明： 此为宏定义，非真正函数。

范 例

```
/* 找出 str 字符串中为阿拉伯数字的字符 */
#include <ctype.h>
main ()
{
    char str[]="123c@#FDsP[e?";
    int i;
    for (i = 0; str[i] != 0;i++)
        if (isdigit (str[i]) ) printf ("%c is an digit character\n", str[i]) ;
}
```

执行结果

```
1 is an digit character
```



```
2 is an digit character
```

```
3 is an digit character
```

①

isgraph (测试字符是否为可打印字符)

相关函数 : isprint

表头文件 : #include <ctype.h>

定义函数 : int isgraph (int c);

函数说明 : 检查参数 c 是否为可打印字符, 若 c 所对映的 ASCII 码可打印, 且非空格字符则返回 TRUE。

返回值 : 若参数 c 为可打印字符, 则返回 TRUE, 否则返回 NULL (0)。

附加说明 : 此为宏定义, 非真正函数。

范 例

```
/* 判断 str 字符串中那些为可打印字符 */
#include <ctype.h>
main ()
{
    char str[]="a5 @;";
    int i;
    for (i = 0; str[i] != 0;i++)
        if (isgraph (str[i]) )
            printf ("str[%d] is printable character. %d\n", i, str[i]) ;
}
```

执行结果

```
str[0] is printable character: a
str[1] is printable character: 5
str[3] is printable character: @
str[4] is printable character: ;
```