**设置日志回调函数**

#define EVENT\_LOG\_DEBUG 0

#define EVENT\_LOG\_MSG 1

#define EVENT\_LOG\_WARN 2

#define EVENT\_LOG\_ERR 3

/\* Deprecated; see note at the end of this section \*/

#define \_EVENT\_LOG\_DEBUG EVENT\_LOG\_DEBUG

#define \_EVENT\_LOG\_MSG EVENT\_LOG\_MSG

#define \_EVENT\_LOG\_WARN EVENT\_LOG\_WARN

#define \_EVENT\_LOG\_ERR EVENT\_LOG\_ERR

typedef void (\*event\_log\_cb)(int severity, const char \*msg);

void event\_set\_log\_callback(event\_log\_cb cb);

在写入日志的时候event\_log\_cb将会被调用

日志回调函数示例

static FILE \*logfile = ...;

static void write\_to\_file\_cb(int severity, const char \*msg){

const char \*s;

if (!logfile)

return;

switch (severity) {

case \_EVENT\_LOG\_DEBUG: s = "debug"; break;

case \_EVENT\_LOG\_MSG: s = "msg"; break;

case \_EVENT\_LOG\_WARN: s = "warn"; break;

case \_EVENT\_LOG\_ERR: s = "error"; break;

default: s = "?"; break; /\* never reached \*/

}

fprintf(logfile, "[%s] %s\n", s, msg);

}

**设置内存管理使用函数**

Libevent默认使用malloc、realloc和free进行内存管理

如下可替换掉这些函数

void event\_set\_mem\_functions(void \*(\*malloc\_fn)(size\_t sz),

void \*(\*realloc\_fn)(void \*ptr, size\_t sz),

void (\*free\_fn)(void \*ptr));