atexit 与 on_exit 的区别：

atexit 是 退出前 执行的函数，不带参数

on_exit 是 退出时 执行的函数 ， 带 结束码 ， 以及自己传进去的参数

函数名：**atexit**

功　能：注册终止函数

用　法：int **atexit**(**atexit**_t func);

个人理解：终止函数放在一个栈里，每个注册的终止函数都会执行，后进入的先执行

程序例：

```
#include <stdio.h>
#include <stdlib.h>

void exit_fn1(void)
{
   printf("Exit function #1 called\n");
}

void exit_fn2(void)
{
   printf("Exit function #2 called\n");
}

int main(void)
{
   /* post exit function #1 */
   atexit(exit_fn1);
   /* post exit function #2 */
   atexit(exit_fn2);
   return 0;
}
```

原型：extern void **exit**(int retval);

功能：结束程序

说明：返回值将被忽略

头文件：#include <stdlib.h>

定义函数：int on_exit(void (* function) (int void*), void *arg);

函数说明：on_exit()用来设置一个程序正常结束前调用的函数. 当程序通过调用exit()或从main 中返回时, 参数function 所指定的函数会先被调用, 然后才真正由exit()结束程序. 参数arg 指针会传给参数function函数, 详细情况请见范例.

返回值：如果执行成功则返回0, 否则返回-1, 失败原因存于errno 中.

范例

```
#include <stdlib.h>
void my_exit(int status, void *arg)
{
      printf("before exit()!\n");
      printf("exit (%d)\n", status);
      printf("arg = %s\n", (char*)arg);
}
main()
{
      char * str = "test";
      on_exit(my_exit, (void *)str);
      exit(1234);
}
```

执行：

```
before exit()! exit (1234) arg = test
```