

# APUE - 6

阅读目的：了解 UNIX 系统环境中的进程环境以及 C 程序的典型存储空间布局。

阅读时间：3 小时

阅读概况：第 7 章

## 第 7 章 进程环境

### 1. main 函数

```
int main(int argc, char *argv[]);
```

当内核执行 C 程序时，在调用 main 前先调用一个特殊的启动例程（exec 函数）。可执行程序文件将此启动例程指定为程序的起始地址，这是由连接编辑器设置的，而连接编译器则由 C 编译器调用。

启动例程从内核取得命令行参数和环境变量值。

### 2. 进程终止

有 8 种方式使进程终止（termination），其中 5 种为正常终止。

正常终止：

1. 从 main 返回；
2. 调用 exit；
3. 调用 \_exit 或 \_Exit；
4. 最后一个线程从其启动例程返回；
5. 从最后一个线程调用 pthread\_exit。

异常终止：

1. 调用 abort；
2. 接到一个信号；
3. 最后一个线程对取消请求作出响应。

main 函数返回后立即调用 exit 函数。形如：

```
exit (main(argc, argv));
```

### 退出函数

```
#include <stdlib.h>

void exit(int status);

void _Exit(int status);

#include <unistd.h>

void _exit(int status);
```

- \_exit 和 \_Exit 立即进入内核，exit 则先执行一些清理处理，然后返回内核。
- exit 函数总是执行一个标准 I/O 库的清理关闭操作。

对于所有打开流调用 fclose 函数。这会造成输出缓冲中的所有数据都被冲洗（写到文件上）。

- 三个函数都返回一个整形参数，称为**终止状态（exit status）**。如果调用函数时不带终止状态，则该进程的终止状态是未定义的。

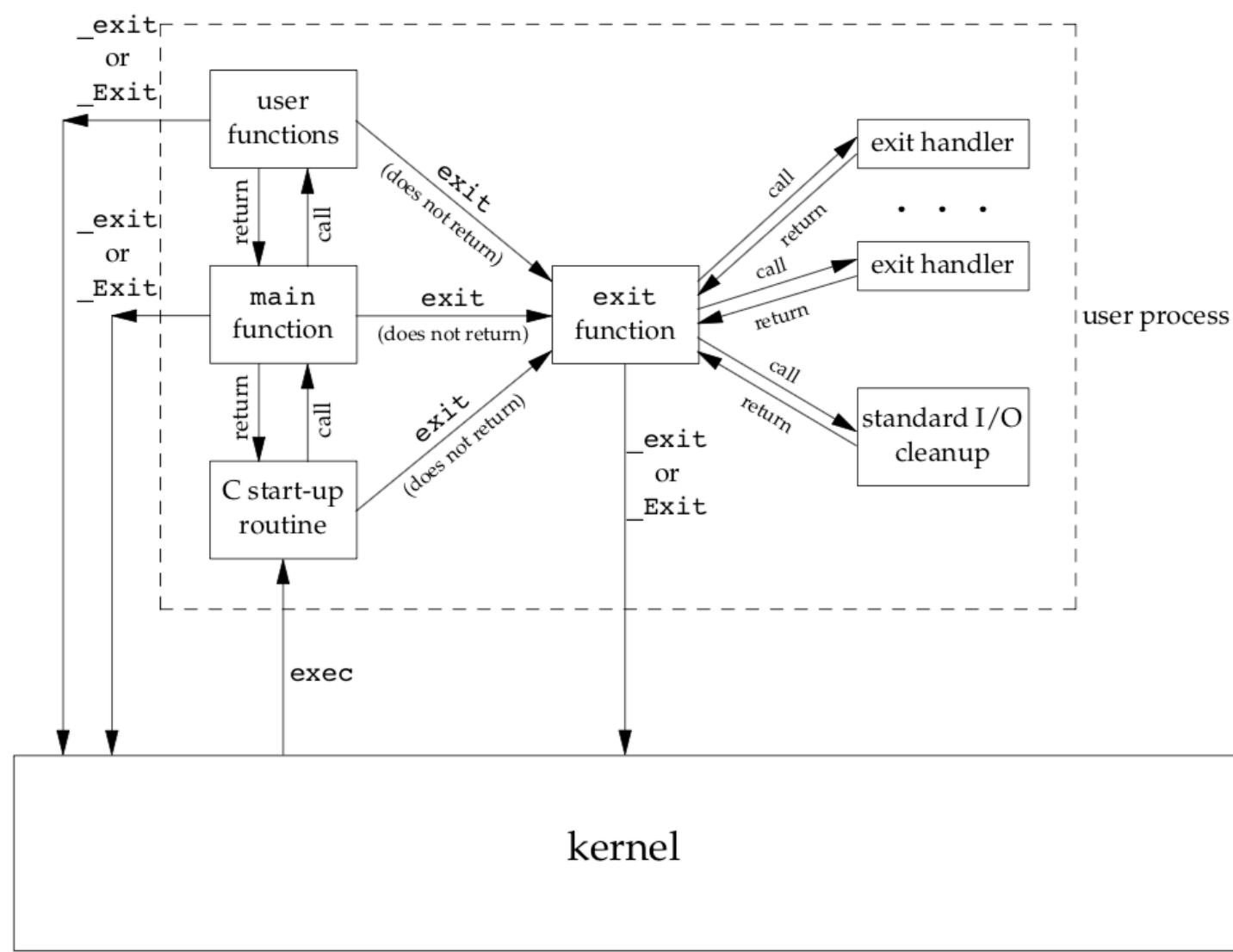
- 若 main 的返回类型是整形，并且 main 执行到最后一条语句时返回（隐式返回），那么该进程的终止状态是 0。（C99）

### atexit 函数

```
#include <stdlib.h>

int atexit(void (&func)(void));
```

atexit 的参数是一个函数地址，当调用此函数时无需向它传递任意参数，也不期望它返回一个值。exit 调用着这些函数的顺序与它们登记时候的顺序相反。同一函数如若登记多次。也会被调用多次。



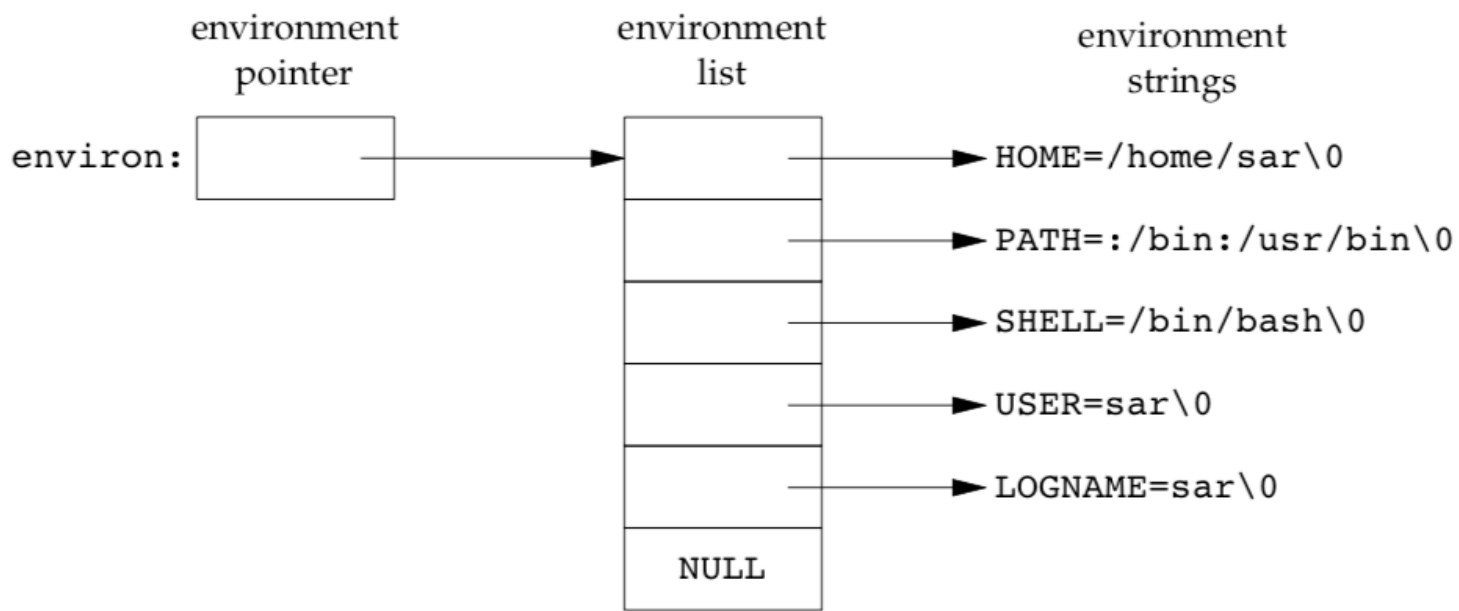
内核使程序执行的唯一方法是调用一个 exec 函数。进程自愿终止的唯一方式是显示或隐式地（通过调用 exit）调用 \_exit 或 \_Exit。进程也可非自愿地由一个信号使其终止。

### 3. 环境表

每个程序都接收到一张环境表。与参数表一样，环境表也是一个字符指针数组，其中每个指针包含一个以 null 结束的 C 字符串的地址。全局变量 environ 则包含了该指针数组的地址：

```
extern char **environ;
```

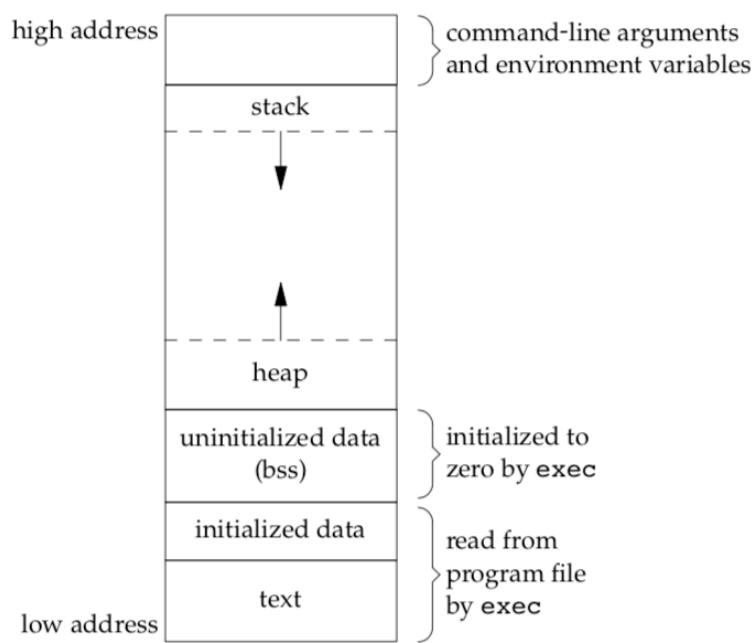
其中每个字符串的结尾处都显示地有一个 null 字节。我们称 environ 为环境指针（environment pointer），指针数组为环境表，其中各指针指向的字符串为环境字符串。



环境由 `name = value` 这样的字符串组成。

通常用 `getenv` 和 `putenv` 函数来访问特定的环境变量，而不是用 `environ` 变量。但是，如果要查看整个环境，则必须使用 **environ** 指针。

## 4. C 程序的存储空间布局



- **正文段**：CPU 执行的机器指令部分。
- **初始化数据段**：程序中需明确地赋初值的变量。
- **未初始化数据段**：在程序开始执行之前，内核将此段中的数据初始化为 0 或空指针。
- **栈**：自动变量以及每次函数调用时所保存的信息。
- **堆**：进行动态存储分配。

## 5. 存储空间分配

此处内容较多，单独整理一份文档