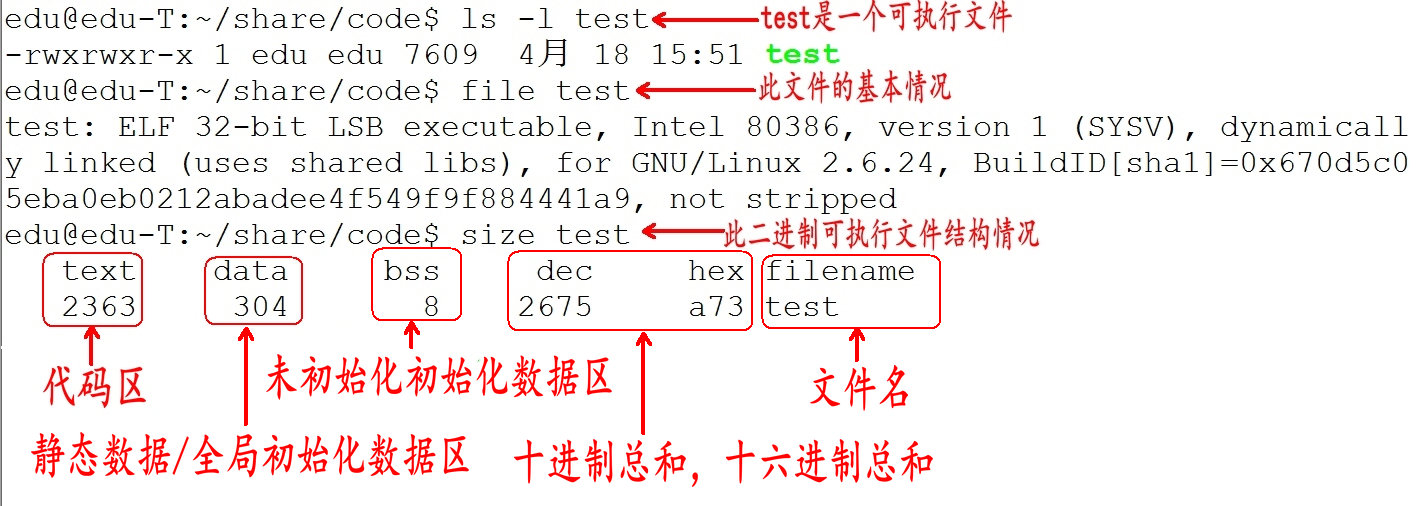
# 一、Linux可执行文件结构

在 Linux 下，程序是一个普通的可执行文件，以下列出一个二进制可执行文件的基本情况：



可以看出，此可执行文件在存储时（没有调入到内容）分为代码区（text）、数据区（data）和未初始化数据区（bss）3 个部分。各段基本内容说明如下：

## ****代码区****：

存放 CPU 执行的机器指令。

## ****全局初始化数据区/静态数据区（数据段，data）****：

该区包含了在程序中明确被初始化的全局变量、已经初始化的静态变量和常量数。

## ****未初始化数据区（ BSS ）****：

存入的是全局未初始化变量和未初始化静态变量。

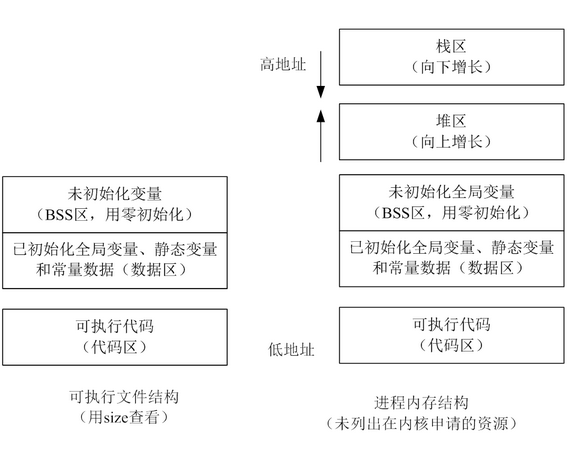
例如，一个不在任何函数内声明的未初始化变量。

**long** sum[1000];

将 sum 存储到未初始化数据

# 二、Linux进程结构

下图为可执行文件存储结构和 Linux 进程基本结构（部分）的对照图。



一个进程主要包括如下：

（1）**代码区（text segment）**。加载的是可执行文件代码段  
（2）**全局初始化数据区/静态数据区（Data Segment）**。加载的是可执行文件的数据段  
（3）**未初始化数据区（BSS）**。加载的是可执行文件BSS段  
（4）**栈区（stack）**。由编译器自动分配释放，存放函数的参数值、返回值、局部变量等。  
（5）**堆区（heap）**。用于动态内存分配。堆在内存中位于BSS区和栈区之间。一般由程序员分配和释放，若程序员不释放，程序结束时有可能由OS回收。

## C 各存储类型比较

