# 1、概述

当程序异常退出时，内核会在当前工作目录下生成一个core文件，其本质就是一个关于异常发生时刻内核的映像，同时加上调试信息。

core文件在使用gdb进行调试的时候很有用，可以显示出导致程序出错的代码所在文件和行数。

获取内核转储（core dump）的最大好处是，它能保存问题发生时的状态，只要有问题发生时程序的可执行文件和内核转储，就可以知道进行当时的状态。

内核有问题时的现象：

区分方法 结果

Ps 显示中途停止，状态为D

Ping 不返回响应

键盘 键盘无法输入

Kill -9 无法结束程序

Strace 无法附加attach到进程（无响应）

Gdb 无法附加attach到进程（无响应）

显示内核信息 softlockup等有输出结果

注：这里添加调试信息需要在gcc编译的时候加上-g参数，否则gdb调试时是没有任何调试信息输出的。

# 2、开关

在使用core文件前，需要查看该系统是否已经打开core文件的生成开关，具体指令为：

ulimit –c

该指令表示core文件大小的上限，如果输出结果为0，即core文件大小上限为0，则表示关闭了该功能，不会在指定目录下生成core文件。

打开core文件的方法：

unlimit –a

unlimit –c unlimited

说明：unlimit命令设置后只针对一个终端有效，所以另外启用一个终端时需要重新设置。

generate-core-file 会手动在当前文件生成core文件，不是在系统设置的目录下。

# 3、大小

## 查看开关

使用ulimit –c命令查看core文件的生成开关，若结果为0（即core文件大小上限为0），则表示关闭此功能，不会生成core文件。

## 修改大小

可以通过调整core文件大小来开启上限，具体指令：

unlimit –c filesize

该指令表示限制core文件的大小上限为filesize Kbyte，如果生成的信息超出此大小，core将会被裁剪，最终生成一个不完整的core文件，在调试此core文件的时候，gdb会提示错误。

如果不设置core文件大小的上限，可以这样设置：

unlimit –c unlimited

# 4、配置

## （1）/proc文件系统

/proc文件系统是一个伪文件系统，就是说他是一个实际不存在的目录，因为还是一个非常特殊的目录，它以文件系统的方式为访问系统内核数据的操作提供接口，它并不存在于某个磁盘上，而是由核心在内存中产生，这个目录用于提供关于系统的信息。/proc目录的内容为系统启动时自动生成的，可以通过修改/proc的文件微内核参数。

使用vi无法成功编辑/proc/sys/kernel/core\_patterm，只能使用echo命令修改或者命令sysctl修改（可以通过在/etc/sysctl.conf文件中，对sysctl变量kernel\_core\_patterm的设置）。

说明：core文件是运行时内存的信息，proc存储的正是运行的一些信息。

默认是在当前目录下生成core文件，但是一般我们需要设置一个固定的目录存放core文件，以便于后期调试。对应的配置文件为/etc/sysctl.conf。

目前，有三种配置方法，分别是：

## （2）临时修改

echo ‘/home/project/core/core.%e.%p.%t’ > /proc/sys/kernel/core\_pattern

这种方式只是临时测试使用，因为/proc目录本身是动态加载的，每次主机重启都会重新加载，生产环境不推荐使用这种方式。

## （3）永久修改

修改/etc/sysctl.conf（该配置文件对应/proc/sys目录及其子目录）文件中的配置项，对sysctl变量kernel.core\_pattern进行设置：

# Allow suid programs to dump core

# Dump core in the location

kenel.core\_pattern = /home/project/core/core.%e.%p.%t

kernel.core\_uses\_pid = 0

配置完后手动生效：

sysctl –p /etc/sysctl.conf

说明：

#### kernel.core\_pattern

kernel.core\_pattern配置项用于设置core文件的格式，这里的%e表示可执行文件名称，%p表示被转储进程的PID，%t表示时间，这样就可以唯一确定core文件对应的具体目标进程了。

参数列表包括：

%p 添加PID

%u 添加当前UID

%g 添加当前GID

%s 添加导致产生core的信号

%t 添加core文件生成时的unit时间

%h 添加主机名

%e 添加命令名

举例：

echo “/corefile/core-%e-%p-%t” > /proc/sys/kernel/core\_patterm，可以将core文件统一生成到/corefile目录下，产生的文件命令为core-命令名-pid-时间戳。

#### core\_uses\_pid

core\_uses\_pid可以控制core文件的文件名是否添加PID作为拓展，如果该配置项被配置成1，即使core\_pattern中没有设置%p，最后生成的core dump文件名仍会加上进程ID；如果配置为0，则表示生成的core文件统一命名为core。

举例：

echo “1”> “/proc/sys/kernel/core\_uses\_pid”

## （4）sysctl命令配置生效

sysctl -w kenel.core\_pattern = " /home/project/core/core.%e.%p.%t "

综上所述，我们实际生产环境中推荐使用修改配置文件/etc/sysctl.conf的方式。

# 5、调试

在程序异常终止后生成core文件，这里包含着重要的调试信息，使用gdb可以发现这些异常，具体操作：

gdb 可执行文件 core文件

bt（使用该命令可以看到程序出错的地方）

说明：用gdb同时调试一个运行程序和core文件，core是程序执行后core dump产生的文件。当程序非法崩溃的时候会产生一个core文件，然后使用这个命令会直接定位到发生程序崩溃的地方。

注意有时候需要设置系统命令ulimit –t unlimited才能产生core文件。