

查阅相关资料，动手写测试程序，基于 CentOS 7，给出下列知识点的回答：

【本次作业背景：】假设 test 程序是一个死循环，每 10 秒打印一句话，则编译后我们在#状态下通过 ./test 命令运行程序，如果不按 CTRL+C 终止，#或\$提示符不会再出现，也不能通过键盘在这个终端上输入任何命令

1、每个人的目录结构要求如下（假设学号为 1651234，各人按实修改）：首先建立“学号-000106”子目录（可位于任意子目录下），下面再建立若干空的子目录，示例如下：

```
1651234-000106
```

```
|-- 01  
|-- 02  
|-- 03  
|-- 04  
|-- 05
```

2、正常程序的后台运行及前后台切换

- 在 01 子目录下写 test1-1.c/test1-2.c 两个程序，都是死循环操作，每 5 秒打印一句话（一个程序打印学号，一个程序打印姓名即可），写配套的 makefile 文件，make 后可生成 test1-1 和 test1-2 两个可执行文件
- 用 SecureCRT 登录系统（简称控制台），然后 ./test1-1 运行 test1-1 程序，控制台进入死循环，不再出现#或\$提示符
- 如何将 test1-1 放入后台运行？（此时终端上出现#或\$，可以输入其它命令并执行）
- 如何查看在后台的 test1-1 程序的信息？
- 如何使 test1-1 重新切换到前台来？
- 再次将 test1-1 放入后台，./test1-2 运行 test1-2 程序，然后将 test1-2 也放入后台
- 如何控制将 test1-1/test1-2 中的某一个切换到前台来？
- test1-1/test1-2 的父进程标识是谁？如何查看？
- 用 SecureCRT 再登录一个控制台，用什么命令可以看到 test1-1/test1-2 的信息？此时父进程标识是谁？如何查看？
- 当 test1-1/test1-2 在后台运行时，如果 CTRL+D 退出该控制台登录，在另一个控制台再查看 test1-1/test1-2 的信息，此时父进程标识是谁？
- 能否使 test1-1/test1-2 在终端退出登录后继续运行（不是再次运行）？此时打印信息能否继续出现在新登录的终端上？
- 如何将一个正常程序直接放入后台运行？（不要通过按键切换）
- 本小题的解答，以文字和贴图形式写在文档中，转换为 01.pdf 后放在 01 子目录下提交（01 子目录中应有内容：test1-1.c、test1-2.c、makefile、01.pdf）

3、守护进程的作用、用途、父进程标识的特点

- 在 02 子目录下写 test2.c，功能与 test1-1 相同（死循环操作，每 5 秒打印一次学号即可），写配套的 makefile 文件，make 后可生成 test2 可执行文件
- 用 ./test2 运行后直接成为守护进程（注意：不是用额外的命令把 test2 变成守护进程）
- test2 的进程标识、父进程标识是谁？如何查看？
- 在另一个控制台中能否查看到 test2 进程的相关信息？
- 当 test2 在后台运行时，如果 CTRL+D 退出控制台登录，test2 会怎样？
- 再次登录控制台后，test2 是否仍在运行？此时打印信息能否继续出现在新登录的终端上？
- 引申问题：如果想在进程中打印信息并保证始终能被查看到，应该如何做？
- 本小题的解答，以文字和贴图形式写在文档中，转换为 02.pdf 后放在 02 子目录下提交

(02 子目录中应有内容: test2.c、makefile、02.pdf)

4、守护进程再次分裂子进程 (僵尸进程的处理)

- 在 03 子目录下写 test3-1.c, 循环 10 次, 每隔 3 秒产生一个子进程, 循环完成后, 自己进入死循环操作, 每 5 秒打印“父进程号+自己的进程号+你的学号+main”信息即可; 分裂出的子进程要求每 25 秒打印“父进程号+自己的进程号+你的学号+sub”, 打印三次后子进程退出; 写配套的 makefile 文件, make 后可生成 test3-1 可执行文件
- 分裂出的子进程和守护进程之间的进程 id 有何关联?
- 什么叫僵尸进程? 僵尸进程的产生原因? (test3-1.c 要求在子进程退出后存在僵尸进程)
- 如何杀死僵尸进程?
- 写 test3-2.c, 要求同 test3-1, 但是子进程退出后不能存在僵尸进程
- 本小题的解答, 以文字和贴图形式写在文档中, 转换为 03.pdf 后放在 03 子目录下提交
(03 子目录中应有内容: test3-1.c、test3-2.c、makefile、03.pdf)

5、如何杀死守护进程? 如何杀死守护进程的子进程?

- 在 04 子目录下写 test4-1.c, 循环 10 次, 每隔 3 秒产生一个子进程, 循环完成后, 自己进入死循环操作, 每 5 秒打印“父进程号+自己的进程号+你的学号+main”信息即可; 分裂出的子进程进入死循环操作, 要求每 15 秒打印“父进程号+自己的进程号+你的学号+sub”; 写配套的 makefile 文件, make 后可生成 test4-1 可执行文件
- 如何杀死 test4-1 分裂出来的一个子进程?
- 如何快速杀死 test4-1 分裂出来的全部子进程? (不能用 kill 连续跟 10 个进程号的方法)
- 如果杀死 test4-1, 其子进程会发生哪些变化?
- 写 test4-2.c, 要求与 test4-1 相同, 但是要求杀死 test4-2 后, 它的全部子进程自动退出
- 本小题的解答, 以文字和贴图形式写在文档中, 转换为 04.pdf 后放在 04 子目录下提交
(04 子目录中应有内容: test4-1.c、test4-2.c、makefile、04.pdf)

6、守护进程再次分裂子进程 (极限测试)

- 在 05 子目录下写 test5-1.c, 用 main 函数带参数方式带入一个参数表示循环次数, 然后循环指定参数产生子进程, 每个子进程中定义一个大小为 1024 的字符数组, 任意赋值, 然后进入死循环 (为了屏幕干净, 不用打印信息); 主进程每分裂若干个子进程 (例如: 10 个/100 个等, 可自行决定) 后打印一次“已分裂***个子进程”, 循环结束后/或分裂子进程失败后打印分裂成功的总数, 然后进入死循环; 写配套的 makefile 文件, make 后可生成 test5-1 可执行文件
- 执行方法为 ./test5-1 1000 表示分裂 1000 个子进程
- 子进程样例如下:

```
int sub()
{
    char str[1024];
    //给 str[0..1023]赋一些随机值
    for(;;)
        sleep(1);
}
```

- 虚拟机的内存设置为 512MB, 分裂数量达到多少时, 分裂子进程会失败?
- 虚拟机的内存为 1024MB、2048MB 时, 分裂最大数量又是多少?
- 把 char str[1024]改为 char str[1024*10], 再次测试三种内存下的最大分裂数量
- 写 test5-2.c, 要求与 test5-1 相同, 但是子进程给 str 赋值完后, 不要死循环, 等待 20 秒后子进程退出, 在这种情况下, 如果做到在小内存的情况下分裂完成指定大数量的子进程?
(例如: 在 512MB 内存情况下, 分别 100000 个子进程且必须都分裂成功)

- 在 test5-2.c 中加适当的语句，看分裂的子进程的最大进程号是多少？
- 写 test5-3.c，基本要求同 test5-2，但是由守护进程负责回收每个子进程退出信号，设置两个全局变量做为计数器，一个记录分裂成功的数量，一个记录回收成功的数量，要求全部分裂完成后，且所有子进程都退出后，两个计数器的值要相同（测试数量 ≥ 50000 ）
- 本小题的解答，以文字和贴图形式写在文档中，转换为 05.pdf 后放在 05 子目录下提交（05 子目录中应有内容：test5-1.c、test5-2.c、test5-3.c、makefile、05.pdf）
- 提示：当极限测试导致子进程分裂失败后，Linux 系统还能正常操作吗？

【本次作业目录结构要求及批改方法：】

1、首先建立 16-000106 目录（可位于任意目录下）

2、每个人的目录结构要求如下（假设学号为 1651234，**各人按实修改**）：首先建立“学号-000106”子目录（可位于任意子目录下），下面再建立 01-05 的子目录，示例如下：

```
1651234-000106
|-- 01
|-- 02
|-- ...
`-- makefile （每位同学的总 makefile 文件，make 后能生成所有子目录下的可执行文件）
```

3、提交作业时，每位同学上交一个 linux-daemon.tar.bz2 文件，解压后能得到上述的完整目录结构，截止时间到后，会从每人的交作业目录中复制出来，全部放在 16-000106 目录中

示例如下：

```
16-000106
|-- 1651234-linux-daemon.tar.bz2 （第 1 位同学的作业压缩包）
...
`-- 1654321-linux-daemon.tar.bz2 （最后 1 位同学的作业压缩包）
```

依次解压后，能得到如下目录结构：

```
16-000106
|-- 1651234-000106 （第 1 位同学的作业目录）
...
`-- 1654321-000106 （最后 1 位同学的作业目录）
```

4、进入 16-000106 目录，进行一次 make，就能生成所有可执行文件，示例如下：

```
16-000106
|-- 1651234-000106 （第 1 位同学的作业目录）
...
|-- 1654321-000106 （最后 1 位同学的作业目录）
`-- check.sh （老师事先建好的 shell 文件，准备编译所有同学的本次作业，具体的实现方式是进入到每个学号对应的目录后调用该目录下的总 makefile）
```

5、无法顺利编译则不能得分，对应学号及子目录名错则不能得分

6、作业提交时清除所有的中间文件及生成的可执行文件、源程序备份文件等

【作业要求：】

- 1、将上述问题的解答及对应的简单测试程序写成文档，**10 月 17 日前**网上提交
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业则不得分