第 00 章作业 - Linux 知识补充 - 守护进程的编写及使用方法 05

1652817　钟钰琛　计算机科学与技术

2018 年 10 月 17 日

1. 512MB 的时候，分裂了 3347 个进程

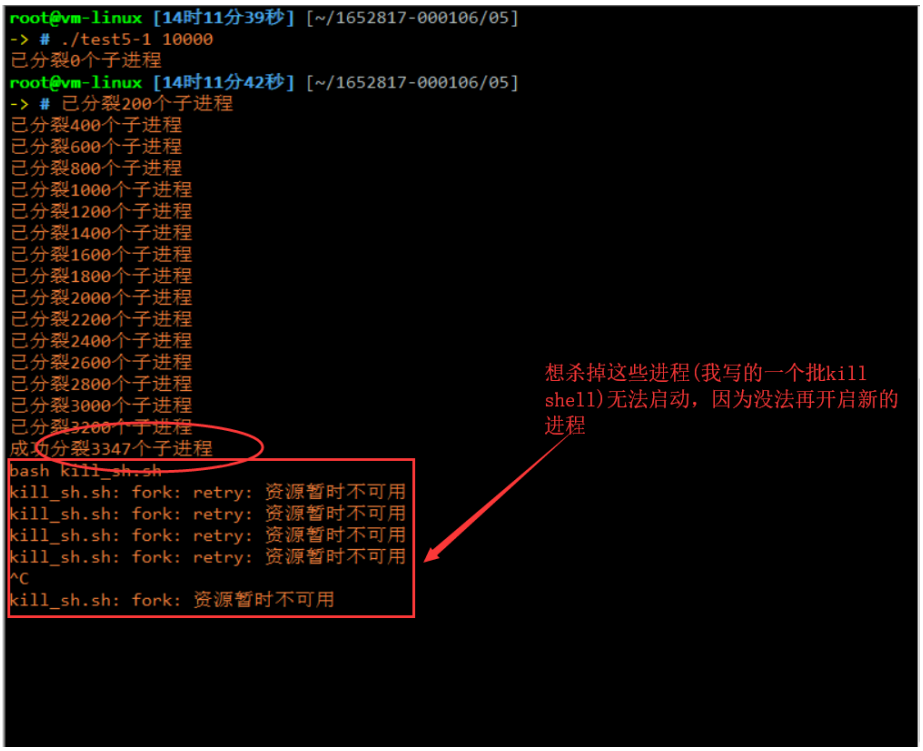

图 1: 512MB 的情况

而后我想 kill 掉这些进程，但是因为不能 fork 新的进程而不能 kill! 所

以现在 linux 什么事也做不了!

2. 1024MB 的时候，分裂了 7737 个进程

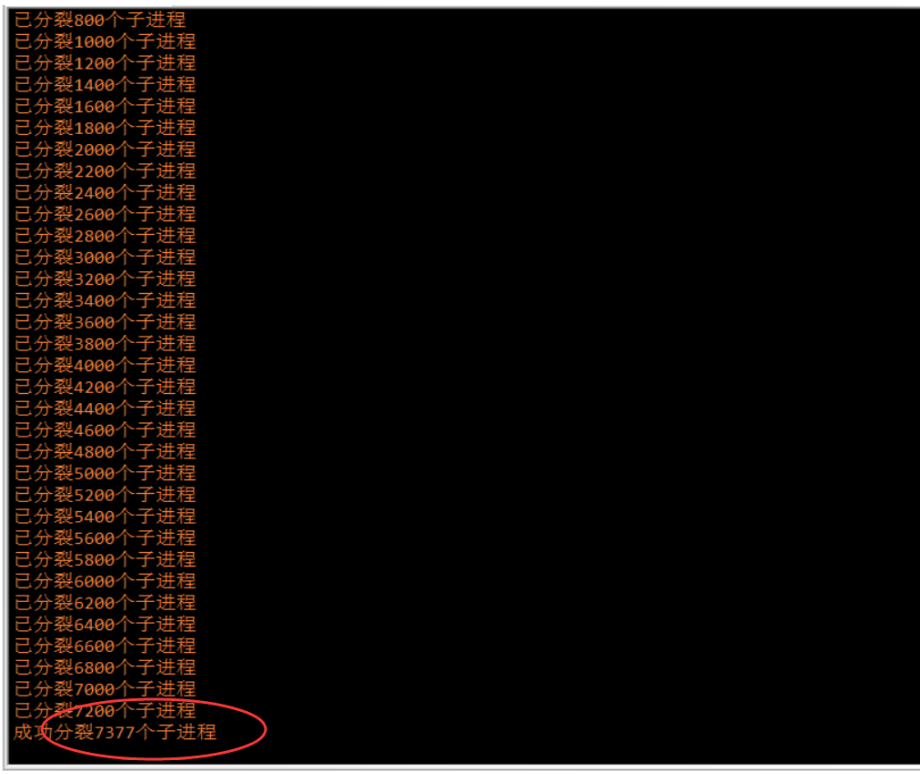

图 2: 1024MB 的情况

3. 2048MB 的时候，分裂了 14154 个进程

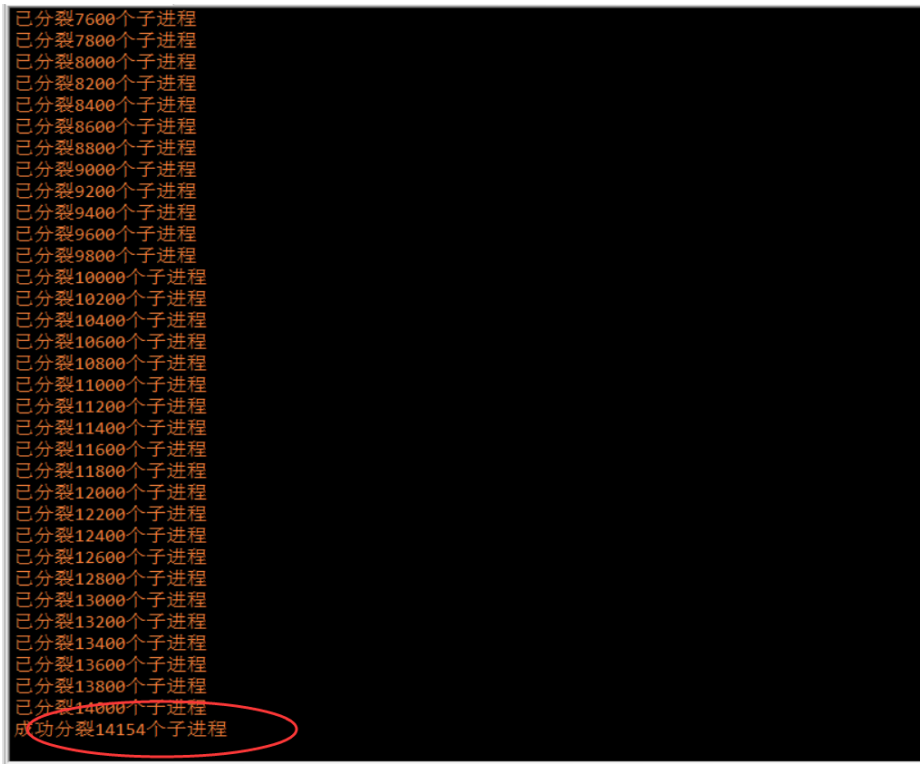

图 3: 2048MB 的情况

从上面三个实验来看，分裂的进程和内存大小基本服从线性关系.

4. 修改为 char str[1024*10], 分别测试 512MB, 1024MB, 2048MB 下和之前相比，分裂的进程数几乎没有变化，只是略微减少了一些.

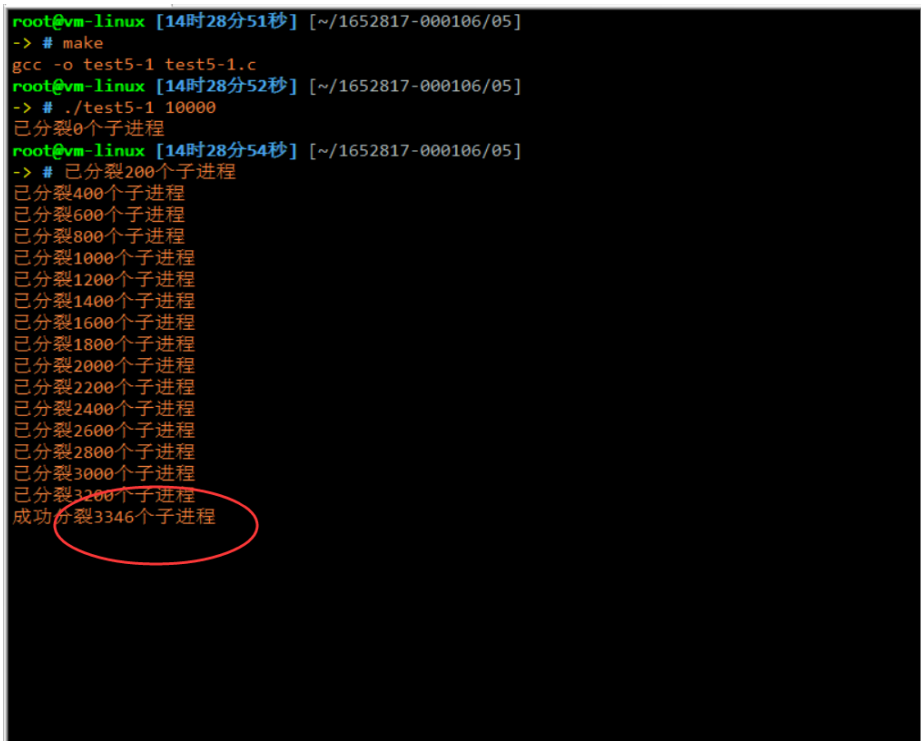

图 4: 512MB 的情况

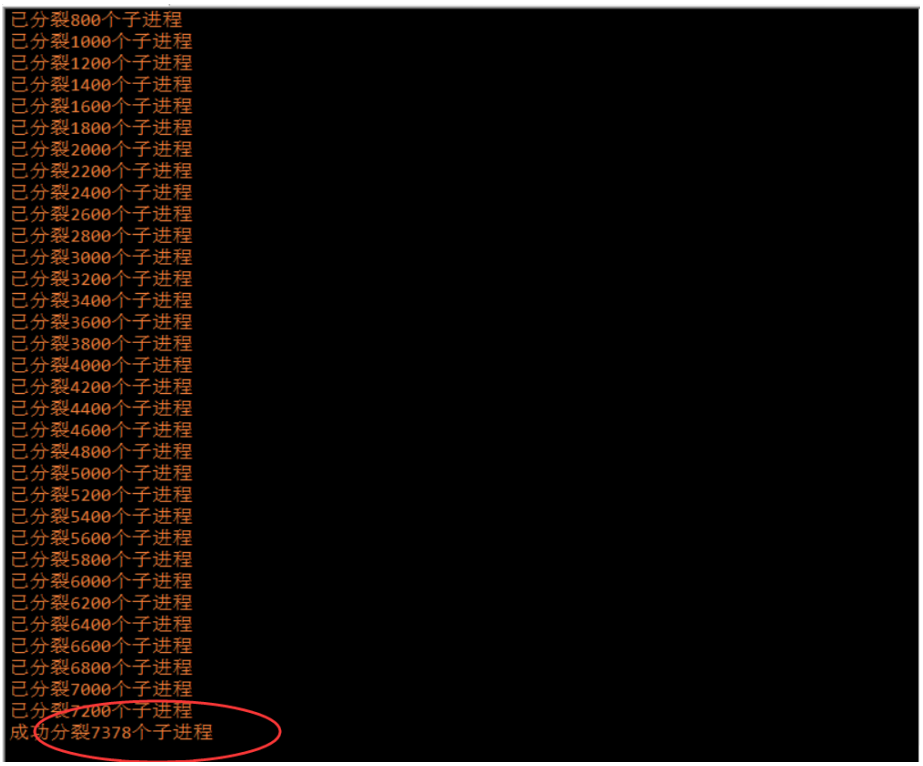

图 5: 1024MB 的情况

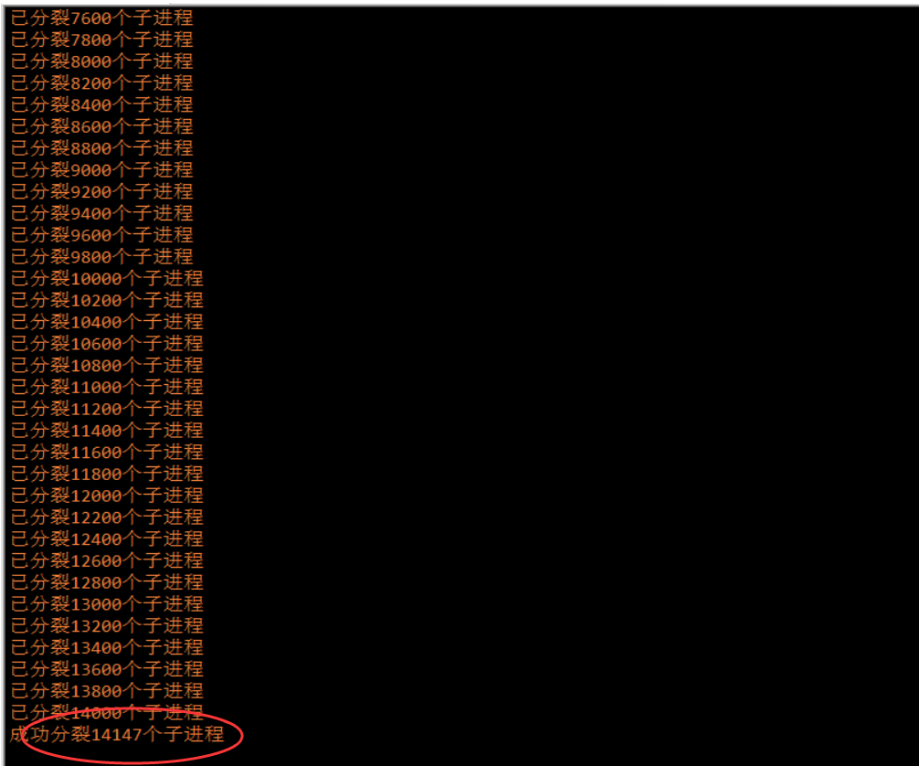

图 6: 2048MB 的情况

5. test5-2.c

我的做法是子进程退出后让 init 进程回收，然后可以创建新的进程.

```
54 int main(int argc, char* argv[]){
55         if(argc != 2){
56                 printf("Usage: ./test4-1 number\n");
57                 exit(EXIT_FAILURE);
58         }
59         int n = atoi(argv[1]);
60         //printf("loop %d times\n", n);
61
62         create_deamon();
63
64         int i;
65         int flag = 1;
66         for (i = 0;i < n; ++i){
67                 pid_t pid = fork();
68
69                 // loop until fork sucessfully
70                 while(pid < 0){
71                         sleep(1);
72                         printf(".");
73                         pid = fork();
74                 }
75
76                 // child
77                 if (pid == 0){
78                         char str[1024];
79                         str[0] = 'z';
80                         str[511] = 'y';
81                         str[1023] = 'c';
82
83                         sleep(20);
84                         exit(EXIT_SUCCESS);
85                 }
                                                    54,25         82%
```

图 7: test5-2.c 部分代码

为了让子进程退出后自动回收,和 T4 最后一问一样,添加 signal(SIGCHLD, SIG_IGN).

打印这么多点是因为父进程在等待的时候打点

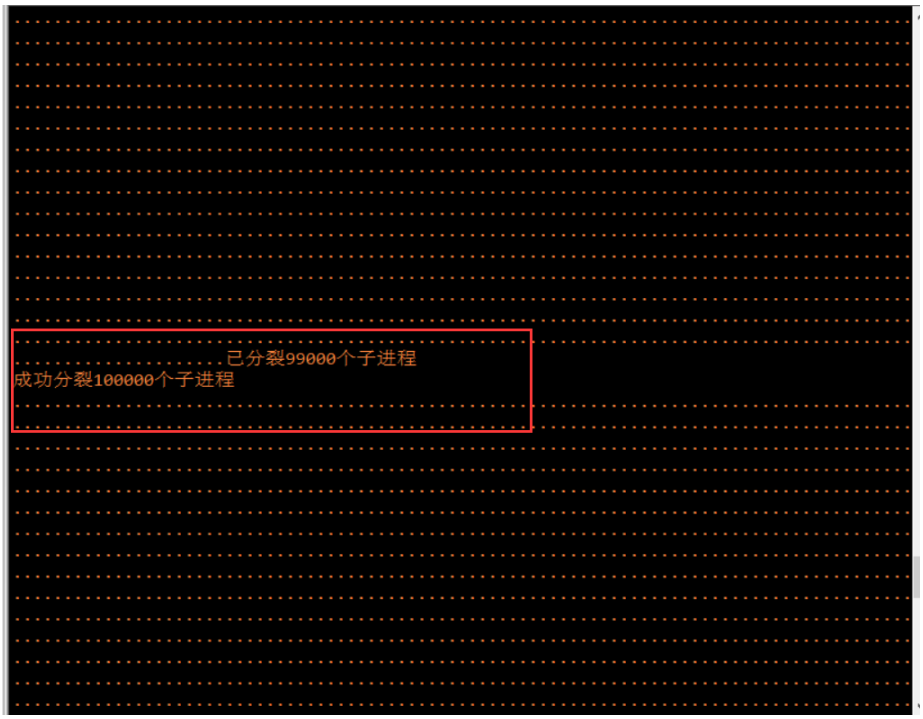

图 8: 512MB 内存下执行 test5-2.c, 成功 fork100000 个进程

6. 最大进程号
(为了清楚起见把打印. 删了) 最大进程号 131071

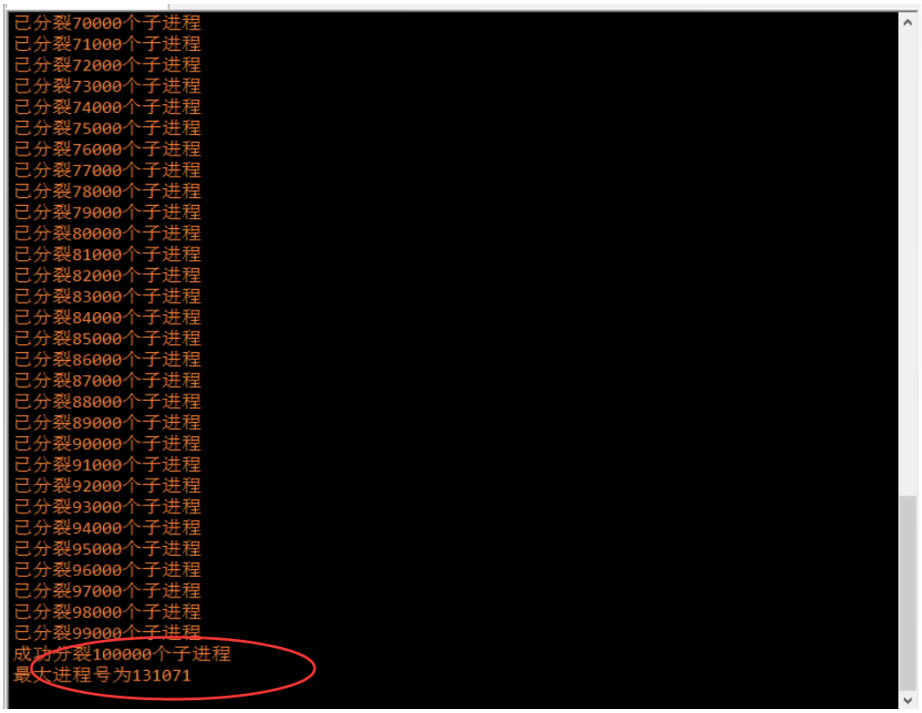

图 9: 最大进程号

7. test5-3.c

如果强行要求父进程来回收，就必须要用 wait/waitpid 了. 同时删掉之前 signal(SIGCHLD, SIG_IGN) 具体可以这么使用

图 10: waitpid

结果如下

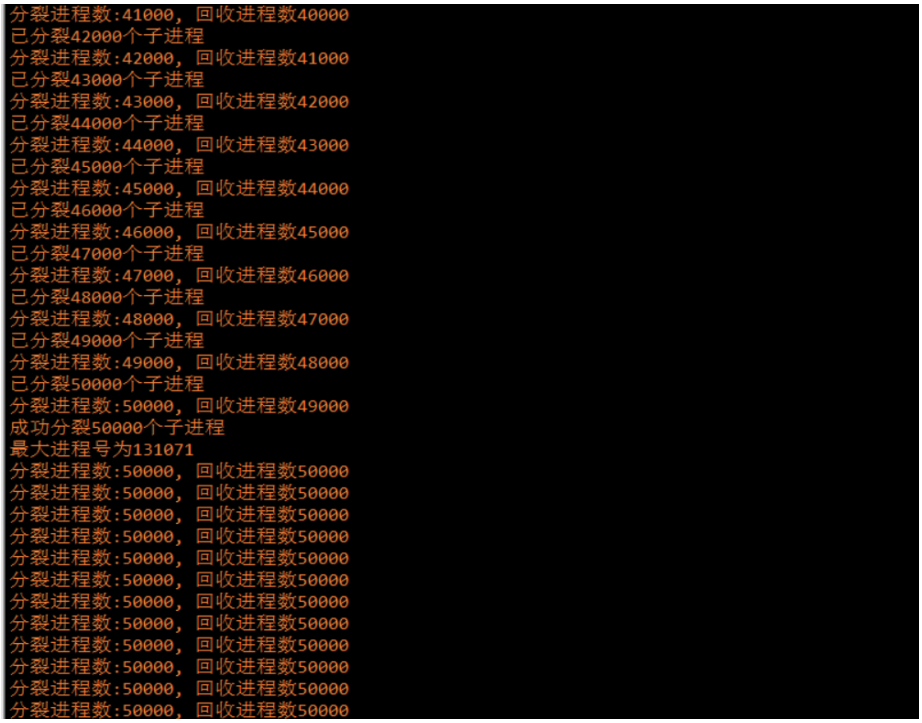

图 11: 执行 test5-3