

System programming homework #10

王興彥 408410113

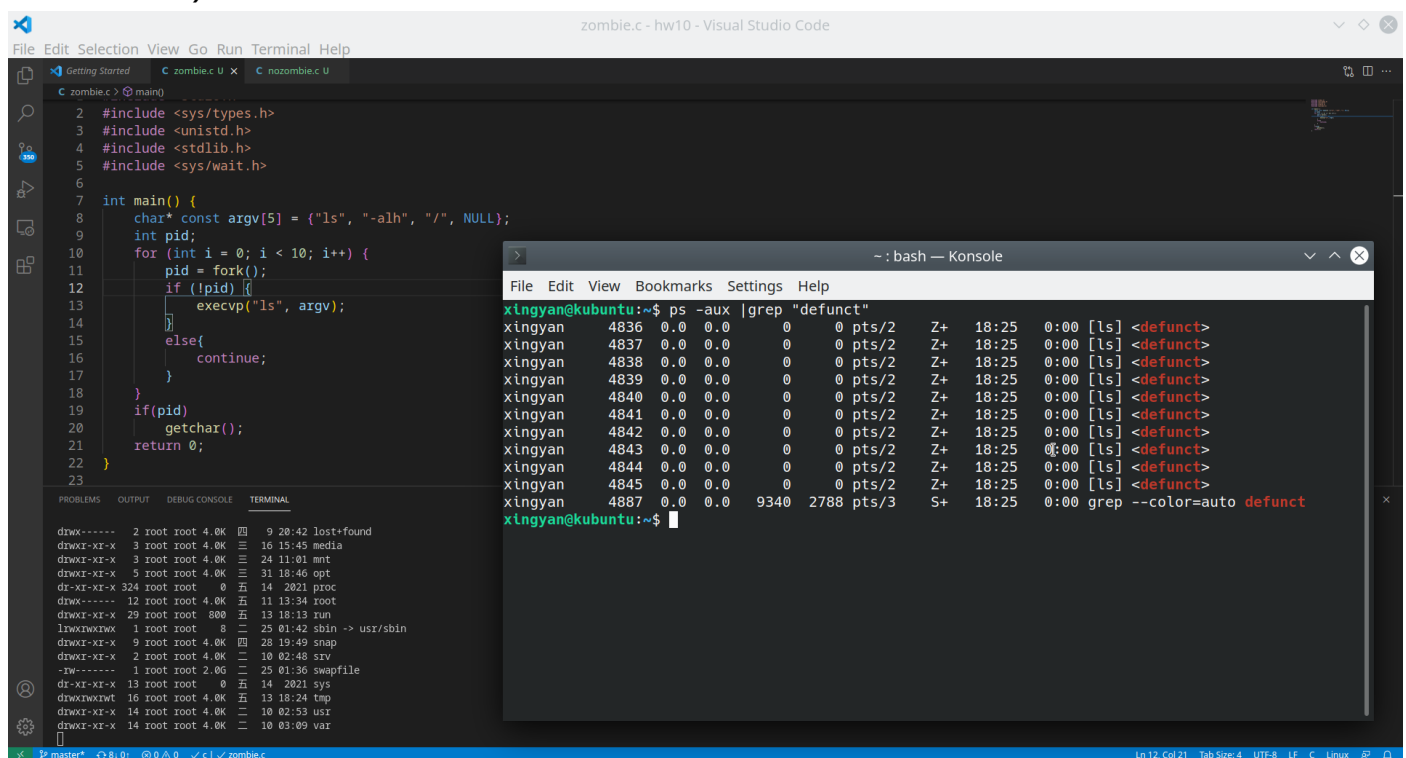
1. 產生 zombie 以後是否可以使用 kill 指令將 zombie 殺掉？

不行，因為他已經死了，如果要 kill 掉 zombie 必須從他的 parent 下手，將 parent kill 掉後，zombie 會被 pid1 給繼承，接著會 wait zombie 然後清理掉它。

參考資料: <https://stackoverflow.com/questions/16944886/how-to-kill-zombie-process>

2. 請附上截圖證明『zombie』的確會產生 10 個 zombie。

由下圖可以看到，執行程式時 parent 還沒執行完畢，child 已經執行完畢，確實會產生 zombie(pid 4836 ~4845)。



```
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5 #include <sys/wait.h>
6
7 int main() {
8     char* const argv[5] = {"ls", "-alh", "/", NULL};
9     int pid;
10    for (int i = 0; i < 10; i++) {
11        pid = fork();
12        if (!pid) {
13            execvp("ls", argv);
14        }
15        else{
16            continue;
17        }
18    }
19    if(pid)
20        getchar();
21    return 0;
22 }
23
```

```
xingyan@kubuntu:~$ ps -aux | grep "defunct"
xingyan  4836  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4837  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4838  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4839  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4840  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4841  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4842  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4843  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4844  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4845  0.0  0.0      0   0 pts/2    Z+   18:25   0:00 [ls] <defunct>
xingyan  4887  0.0  0.0    9340 2788 pts/3    S+   18:25   0:00 grep --color=auto defunct
```

3. 請附上截圖證明『nozombie』幾乎不會產生 zombie

因為先建立了一個假的 parent，並使用 wait 因此過程中不會產生 zombie。

※下圖中 sleep(i) 應該更正為 sleep(i+1) 才符合作業要求，程式碼部分以更改為正確內容，但實際上不影響測試過程。

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/wait.h>
6
7 int main() {
8     int pid = fork();
9     if (pid == 0) {
10         for (int i = 0; i < 100; i++) {
11             int pid2 = fork();
12             if (pid2 == 0) { //child
13                 printf("child %2d 開始等待\n", i);
14                 sleep(i);
15                 printf("child %2d 告辭了\n", i);
16                 exit(0);
17             } else {
18                 //parent
19                 continue;
20             }
21         }
22         exit(0);
23     } else {
24         int wstatus;
25         wait(&wstatus);
26     }
27     return 0;
28 }
29
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
child 93 告辭了
child 94 告辭了
child 95 告辭了
child 96 告辭了
child 97 告辭了
child 98 告辭了
child 99 告辭了

```

```

File Edit View Bookmarks Settings Help
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5148  0.0  0.0  9340  660 pts/3  S+  18:28  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5158  0.0  0.0  9340  736 pts/3  S+  18:29  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5161  0.0  0.0  9340  728 pts/3  S+  18:29  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5168  0.0  0.0  9340  728 pts/3  S+  18:29  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5170  0.0  0.0  9340  732 pts/3  S+  18:29  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5172  0.0  0.0  9340  2744 pts/3  S+  18:29  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5205  0.0  0.0  9340  732 pts/3  S+  18:30  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5209  0.0  0.0  9340  728 pts/3  S+  18:30  0:00  grep --color=auto defunct
xingyan@kubuntu:~$ ps -aux |grep "defunct"
xingyan  5211  0.0  0.0  9340  724 pts/3  S+  18:30  0:00  grep --color=auto defunct
xingyan@kubuntu:~$

```

4. 請說明你的系統中『task_struct』到底有多大，並附上截圖證明你的說法

Slabinfo 其格式為

```
# name <active_objs> <num_objs> <objsize> <objperslab> <pagesperslab> :
tunables <limit> <batchcount> <sharedfactor> : slabdata <active_slabs>
<num_slabs> <sharedavail>
```

第三欄位為 size，而經由 man slabinfo 可知其單位為 bytes，故 task_struct 大小為 6144bytes。

```

File Edit View Bookmarks Settings Help
xingyan@kubuntu:~$ sudo cat /proc/slabinfo | grep task_struct
task_struct 1427 1560 6144 5 8 : tunables 0 0 0 : slabdata 312 312 0
xingyan@kubuntu:~$

```

5. 在網路上找一張你覺得最能描述「辛棄疾」的『風景畫』。用 10 到 20 個字說明你選這張圖的理由。

辛棄疾詞風「激昂豪邁，風流豪放」，感覺很符合這張風景畫所展現出的意象。

