系統程式設計 System Programming 2021

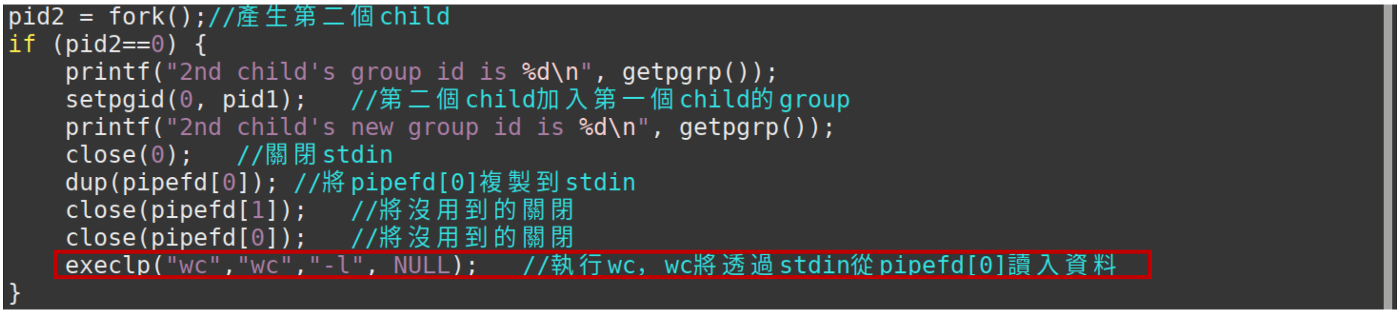
電機四 406415089 宋 O 天

Hw12 Pipe

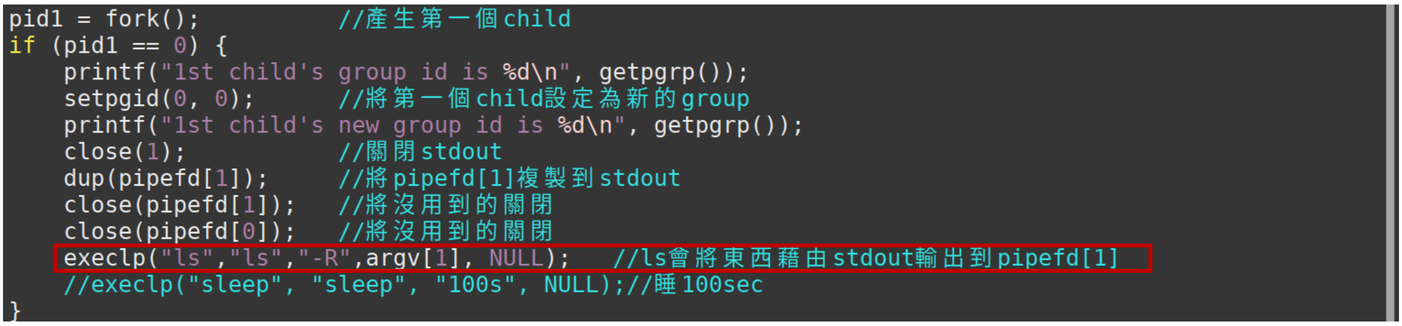
註

<1> 設計 fileCount.c，執行./filecount /YYY 如同執行 ls R /YYY | wc -l 計算 /YYY 目錄下大約有多少字元

完成，依照 pipe4-3.c 去改，主要就更改 execlp 的參數



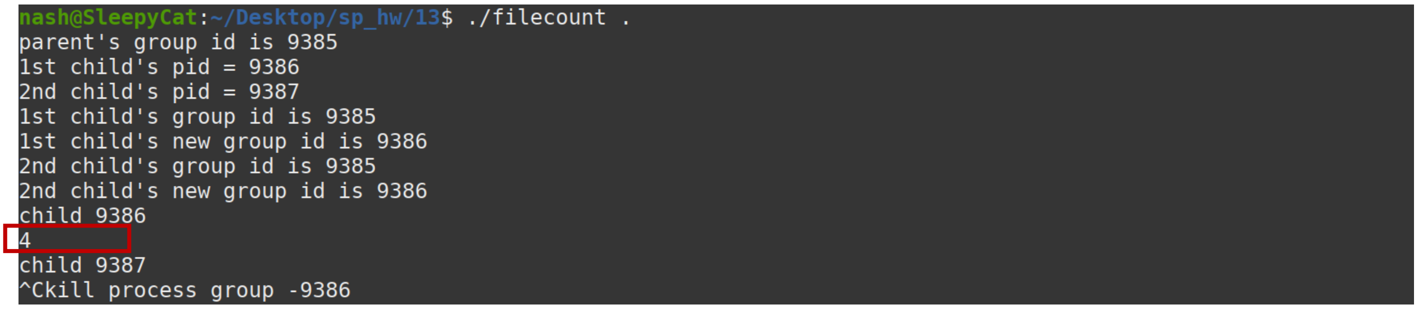
```
pid2 = fork();//產生第二個child
if (pid2==0) {
    printf("2nd child's group id is %d\n", getpgrp());
    setpgid(0, pid1);    //第二個child加入第一個child的group
    printf("2nd child's new group id is %d\n", getpgrp());
    close(0);   //關閉stdin
    dup(pipefd[0]); //將pipefd[0]複製到stdin
    close(pipefd[1]);    //將沒用到的關閉
    close(pipefd[0]);    //將沒用到的關閉
    execlp("wc","wc","-l", NULL);    //執行wc, wc將透過stdin從pipefd[0]讀入資料
}
```



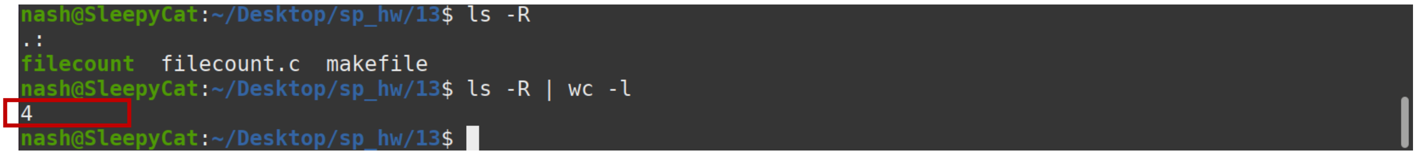
```
pid1 = fork();          //產生第一個child
if (pid1 == 0) {
    printf("1st child's group id is %d\n", getpgrp());
    setpgid(0, 0);      //將第一個child設定為新的group
    printf("1st child's new group id is %d\n", getpgrp());
    close(1);           //關閉stdout
    dup(pipefd[1]);     //將pipefd[1]複製到stdout
    close(pipefd[1]);   //將沒用到的關閉
    close(pipefd[0]);   //將沒用到的關閉
    execlp("ls","ls","-R",argv[1], NULL);    //ls會將東西藉由stdout輸出到pipefd[1]
    //execlp("sleep", "sleep", "100s", NULL);//睡100sec
}
```

ls | wc -l	統計當前目錄下的非隱藏目錄和非隱藏檔案的個數
ls -a | wc -l	統計當前目錄下的所有檔案和目錄的個數實際數目

執行結果：



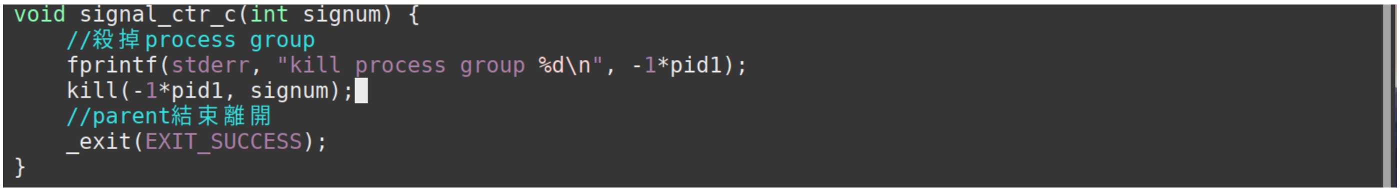
```
nash@SleepyCat:~/Desktop/sp_hw/13$ ./filecount .
parent's group id is 9385
1st child's pid = 9386
2nd child's pid = 9387
1st child's group id is 9385
1st child's new group id is 9386
2nd child's group id is 9385
2nd child's new group id is 9386
child 9386
4
child 9387
^Ckill process group -9386
```



```
nash@SleepyCat:~/Desktop/sp_hw/13$ ls -R
.:
filecount  filecount.c  makefile
nash@SleepyCat:~/Desktop/sp_hw/13$ ls -R | wc -l
4
nash@SleepyCat:~/Desktop/sp_hw/13$ 
```

<2>　確保主程式 fileCount 收到 ctr+c 以後，可以讓 ls -R /YYY | wc -l 立即中止

完成，一樣修改一下 pipe4-3.c



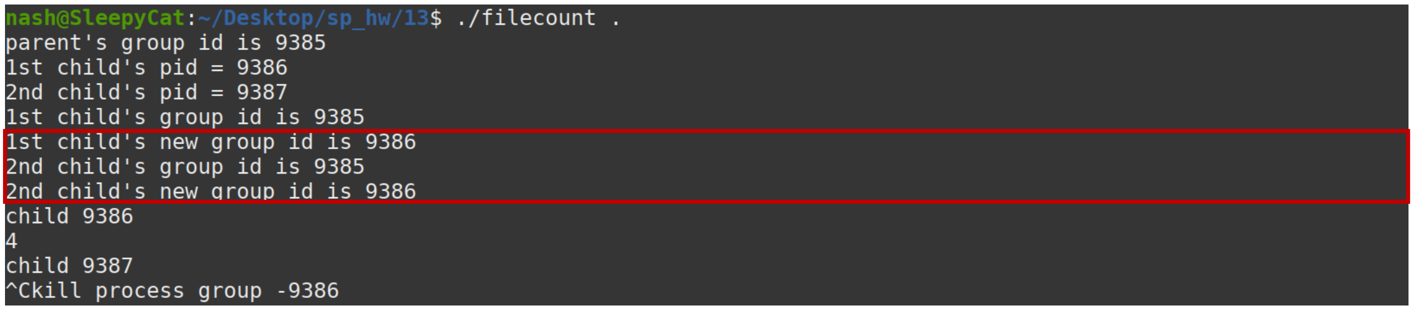
```
void signal_ctr_c(int signum) {
    //殺掉process group
    fprintf(stderr, "kill process group %d\n", -1*pid1);
    kill(-1*pid1, signum);
    //parent結束離開
    _exit(EXIT_SUCCESS);
}
```

man page 可以看見以下內容，所以才寫成 kill (-1*pid,...)這種形式

If *pid* is less than -1, then *sig* is sent to every process in the process group whose ID is *-pid*.

從執行結果來看，Child 1 和 2 都屬於同一個 Group，Parent 則和他們不一樣，

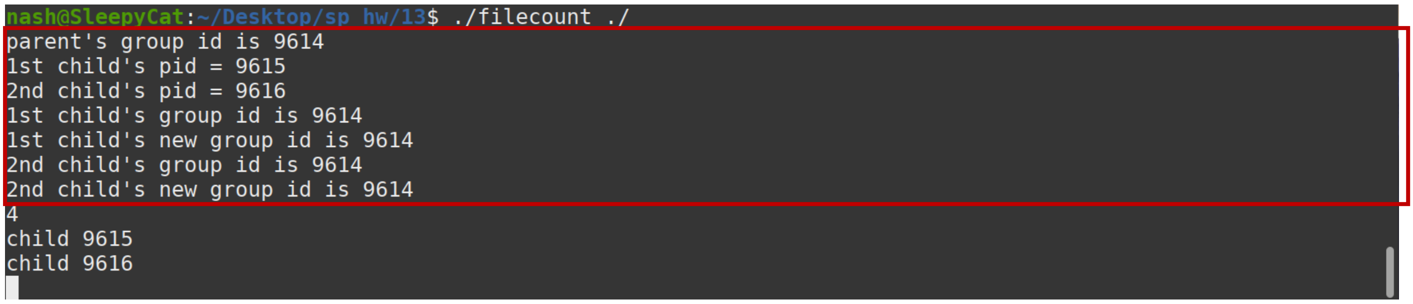
兩個 child 是藉由 setpgid()去設定的



```
nash@SleepyCat:~/Desktop/sp_hw/13$ ./filecount .
parent's group id is 9385
1st child's pid = 9386
2nd child's pid = 9387
1st child's group id is 9385
1st child's new group id is 9386
2nd child's group id is 9385
2nd child's new group id is 9386
child 9386
4
child 9387
^Ckill process group -9386
```

<3>你是否可以在不修改 SIGINT 的 signal handler 的情況下完成第二小題呢？

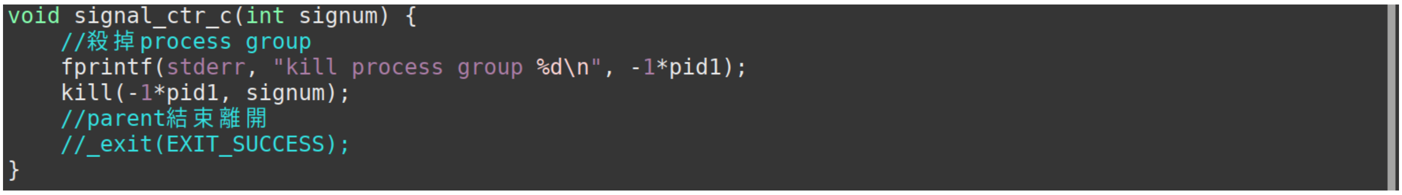
把 setpgid()註解掉的話，parent 和 child 都會是相同的 group，此時 ctrl+c 就可以都砍掉



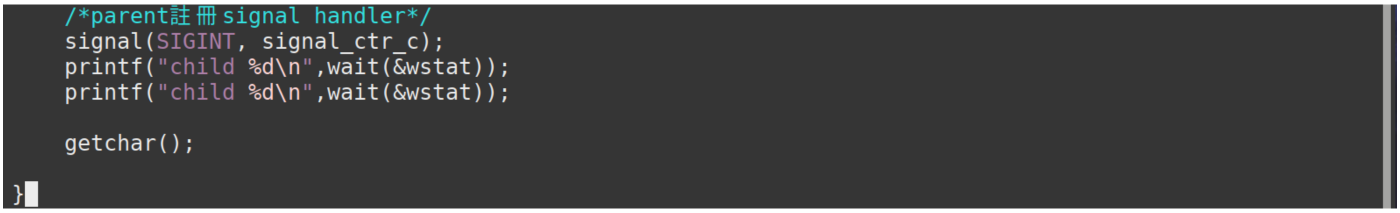
```
nash@SleepyCat:~/Desktop/sp_hw/13$ ./filecount ./
parent's group id is 9614
1st child's pid = 9615
2nd child's pid = 9616
1st child's group id is 9614
1st child's new group id is 9614
2nd child's group id is 9614
2nd child's new group id is 9614
4
child 9615
child 9616
```

如果我在 signal handler 裡面將 parent process 的_exit 註解掉，然後在程式後面加上 getchar()，

就會發現 ctrl+c 只會砍掉兩個 child(同一個 process group)，parent 在 stdin 輸入之後才會停止



```
void signal_ctr_c(int signum) {
    //殺掉process group
    fprintf(stderr, "kill process group %d\n", -1*pid1);
    kill(-1*pid1, signum);
    //parent結束離開
    //_exit(EXIT_SUCCESS);
}
```



```
    /*parent註冊signal handler*/
    signal(SIGINT, signal_ctr_c);
    printf("child %d\n",wait(&wstat));
    printf("child %d\n",wait(&wstat));

    getchar();

}
```

<參考資料>

https://blog.gtwang.org/linux/linux-wc-command-tutorial-examples/

Linux 使用 wc 指令計算字數、行數教學與範例

https://www.itread01.com/content/1549205837.html

ls — 列出目標目錄中所有的子目錄和檔案

<致謝>

摯友 博禕