**cpu-api - Homework (Code)**

**Q1.**

1. Write a program that calls fork(). Before calling fork(), have the main process access a variable (e.g., x) and set its value to something (e.g., 100). What value is the variable in the child process?

What happens to the variable when both the child and parent change

the value of x?

Answer :

　　首先依照題意，在第6行，宣告了一個X變數，並將他的值設為100，接著在第8行calls fork()，分別在14~20行以及21~27行顯示了子進程和父進程的程式。在第16~17行以及23~24行輸出子進程以及父進程的X值，如執行結果所示，Old parent,Old child的X值皆為100。結果相同的原因是他們都是用相同的內容創建出來的。接下來依照題目的第二個問題，在18行以及25行分別將子進程的X改為500，將父進程的X改為1000，更改完之後分別在19以及26行輸出他們更改後的值，如執行結果所示，New parent x = 1000,New child x = 500。由以上過程可知，雖然他們在創建的時候擁有相同的內容，但創建後他們卻是2個獨立的。

Code :

一張含有 文字 的圖片

自動產生的描述

執行結果 :一張含有 文字, 時鐘, 綠色, 黑暗 的圖片

自動產生的描述

**Q2.**

2. Write a program that opens a file (with the open() system call) and then calls fork() to create a new process. Can both the child and parent access the file descriptor returned by open()? What happens when they are writing to the file concurrently, i.e., at the same time?

Answer :

　　首先依照題意，在第10行使用open()打開一個文件，該文件為test.txt後面給他們權限，分別為O\_CREAT若該文件不存在則自動建立該文件、O\_WRONLY以只寫入的方式打開文件、O\_TRUNC若該文件存在並且可以寫入，此指令會將文件長度清空，原來的文件也會消失、S\_IRWXU則是該文件所有者具有可讀、可寫、可執行的權限，接下來在第11行calls fork()，然後在第17~18行以及第20~21行分別在子進程以及父進程寫入文件，如結果所示，他們都可以分別寫入文件，當他們同時寫入文件時，如果先進入父進程，他會等待子進程處理完之後，再換他開始做，所以不會同時寫入。

Code :

一張含有 文字 的圖片

自動產生的描述

執行結果 :

一張含有 文字 的圖片

自動產生的描述

**Q3.**

3. Write another program using fork(). The child process should print “hello”; the parent process should print “goodbye”. You should try to ensure that the child process always prints first; can you do this without calling wait() in the parent?

Answer :

首先依照題意，在第8行calls fork()，因題目說在不使用wait()的情況下執行，所以我在第15行使用了waitpid()來確保子進程先輸出 ，最後在第13行以及第16行分別在子進程輸出hello和父進程輸出Goodbye，結果如執行輸出所示。

Code :一張含有 文字 的圖片

自動產生的描述

執行結果 :

一張含有 文字 的圖片

自動產生的描述

**Q4.**

4. Write a program that calls fork() and then calls some form of exec() to run the program /bin/ls. See if you can try all of the variants of exec(), including (on Linux) execl(), execle(),execlp(), execv(), execvp(), and execvpe(). Why do you think there are so many variants of the same basic call?

Answer :

依照題意，首先在第8行calls fork()，接著在子程序中16~22行中分別使用了excl()、execlp()、execle()、execv()、execvp()、execvpe()，L跟V的差別在於是否要將參數傳遞給執行程序，字尾如果有E的可以允許傳遞另一個char\*，是在執行程序啟動之前添加到新的進程中的一組字符串，類似一種傳遞參數的方式，字尾如果有P的，可以用來搜索可執行的文件，如果當前的目錄中沒有可執行文件，則沒有P版本的需要在可執行文件的文件名前加上絕對或相對路徑。

一張含有 文字 的圖片

自動產生的描述

執行結果 :



**Q5.**

5. Now write a program that uses wait() to wait for the child process to finish in the parent. What does wait() return? What happens if you use wait() in the child?

Answer :

首先在第8行calls fork()，創建子進程跟父進程，接下來第9行使用wait()，將子進程以及父進程一起使用wait()並將他的值存到WC，由於父進程一定會先等待子進程做完所以同時使用wait()不會有影響，接下來第20行由子進程先執行，但因為他本身並沒有子進程，所以輸出的WC返回-1，然後輪到父進程執行，由執行結果可知子進程的PID為1891，所以父進程的WC值繼承子進程的PID輸出1891。

一張含有 文字 的圖片

自動產生的描述 執行結果 : 一張含有 文字, 時鐘 的圖片

自動產生的描述

**Q6.**

6. Write a slight modification of the previous program, this time using waitpid() instead of wait(). When would waitpid() be useful?

Answer :

過程同Q5，將第9行的wait() 改成waitpid()，結果也相同。Waitpid()只在進程本身有子進程的時候有用，在一個子進程終止前，waitpid()可以選擇調配使用者的特定進程。一張含有 文字 的圖片

自動產生的描述

執行結果 :一張含有 文字, 時鐘, 儀錶 的圖片

自動產生的描述

**Q7.**

7. Write a program that creates a child process, and then in the child closes standard output (STDOUT FILENO). What happens if the child calls printf() to print some output after closing the descriptor?

Answer :

首先依照題意，在第7行calls fork()，然後在第12行子進程中關閉標準輸出close(STDOUT\_FILENO)，由於輸出的功能已經被關閉，第13行做了printf()輸出一些訊息也沒辦法輸出，如執行結果所示。

一張含有 文字 的圖片

自動產生的描述

執行結果 :



**Q8.**

8. Write a program that creates two children, and connects the standard output of one to the standard input of the other, using the pipe() system call.

Answer :  
 依照題意，使用pipe()將一個標準輸出連接到另一個標準輸入，第11行宣告pipe，接下來第16行calls fork()，創建第一個子進程第20~24行，然後在父進程中第20~44行中的第27行，再進行一次fork()來創建一個子進程，為第32~35行。第23行第一個子進程使用了dup2把stdout指向pipe的寫入端，並將第24行的字串內容傳遞到pipe的寫入端。接下來第35行第二個子進程使用了dup2使stdin指向pipe的讀取端，第37~39行為第一個子進程指向後將pipe內的資料存在一個叫做BUFF的地方，然後把pipe裡面的內容輸出出來，經過以上的過程，第一個子進程的標準輸出成功連接到第二個子進程的標準輸入。

一張含有 文字 的圖片

自動產生的描述

執行結果 :

一張含有 文字, 時鐘, 黑暗, 儀錶 的圖片

自動產生的描述