

110710532 資工二 廖晏霆

系統程式

OPERATING SYSTEMS: THREE EASY PIECES 中文版 心得報告

June 27 2020

前言

爲什麼選擇這本書？這本書在說什麼？這本書在說明有關程式的運作的方式寫很清楚，也說明了程式爲何這麼寫，什麼原因程式要這麼設計，加上本書的內容以談諧方式去描述，用日常的說話方式反而比較易懂，這也是我選擇這本書的原因。

本報告大幅整理來自該書第五章內容

學習筆記

在第五章在說明API 與 UNIX系統，說明 `fork()` ,

`wait()` 和 `exec()` 的強大：

`fork()`: 他被用來新增進程

`wait()`: 調整進程先後

`exec()`: 輕鬆寫兩個以上進程

fork():

他被用來新增進程，利用大於或等於 0 來新增進程

```
int a = fork();
```

```
if( a <0) 如果小於零就離開程式
```

```
{
```

```
    fprintf (stderr, "fork failed\n");
```

```
    exit(1);
```

```
}
```

```
else if (a == 0) printf("hello, I am child (pid:%d)\n", (int) getpid());
```

```
如果等於零會新增一個程式(我們稱為子進程(child)) 返回值為0
```

```
else printf("hello, I am parent of %d (pid:%d)\n",a, (int) getpid());
```

```
這個稱為父進程(parent) 返回值為子進程的pid
```

```
return 0;
```

小幅修改自該書的例子

運行結果：

hello, I am parent of 29147 (pid:29146)

hello, I am child (pid:29147)

藉由傳回值可知，兩個進程不是完全一樣的，這就是 **fork()**強大的地方，可以一次運行兩個沒差很多的進程，就差在父進程比子進程先運行。下一個方法可以讓子進程先運行：

wait():

等待子進程運行完成再執行程式:

```
int a = fork();
```

```
f( a <0)
```

```
{
```

```
    fprintf (stderr, "fork failed\n");
```

```
    exit(1);
```

```
}
```

```
else if (a == 0) printf("hello, I am child);
```

```
else
```

```
{
```

```
    int b = wait(NULL); 父進程會禮讓子進程先運行
```

```
    printf("hello, I am parent);
```

```
}
```

```
return 0;
```

小幅修改自該書的例子

運行結果：

hello, I am child

hello, I am parent

利用**fork()** 還有**wait()**就可以應付很多程式了，後面有一個更強大的**exec()**。

exec():

小幅修改自該書的例子

```
int a = fork();
if (a < 0) {
    fprintf(stderr, "fork failed\n");
    exit(1);
} else if (rc == 0) {
    printf("hello, I am child\n");
    char *myargs[3];
    myargs[0] = strdup("wc"); 執行wc(word count)(舉例程序名稱)
    myargs[1] = strdup("asd"); asd file(舉例參數)
    myargs[2] = NULL;
    execvp(myargs[0], myargs); 執行 wc (word count) 在 asd file
    printf("this shouldn't print out");
} else {
    int b = wait(NULL);
    printf("hello, I am parent\n");
}
return 0;
```


`fork()`是用來方便複製程式而`exec()`是用來寫不同程式的。`exec()`還有一個特點，就是打你重新運行程式時，他會用覆寫的方式執行，把原本的程式清除再執行，讓程式運作順利。`fork()`、`exec()`再加上`wait()`的應用就很強大了，程式的分別也比較容易寫。

結語

以上都只是這本書的一小部分，裡面還介紹很多值得學習的東西，真的很推薦大家去讀，他其實都會把重點框起來(例如一定要看man手冊之類的)而且也蠻好懂的，裡面也有作業還有習題只要你有看懂就會寫了，習題跟作業解答網路上也找的到，我自己有時間也會看這本書一兩次，增進自己的能力與邏輯。