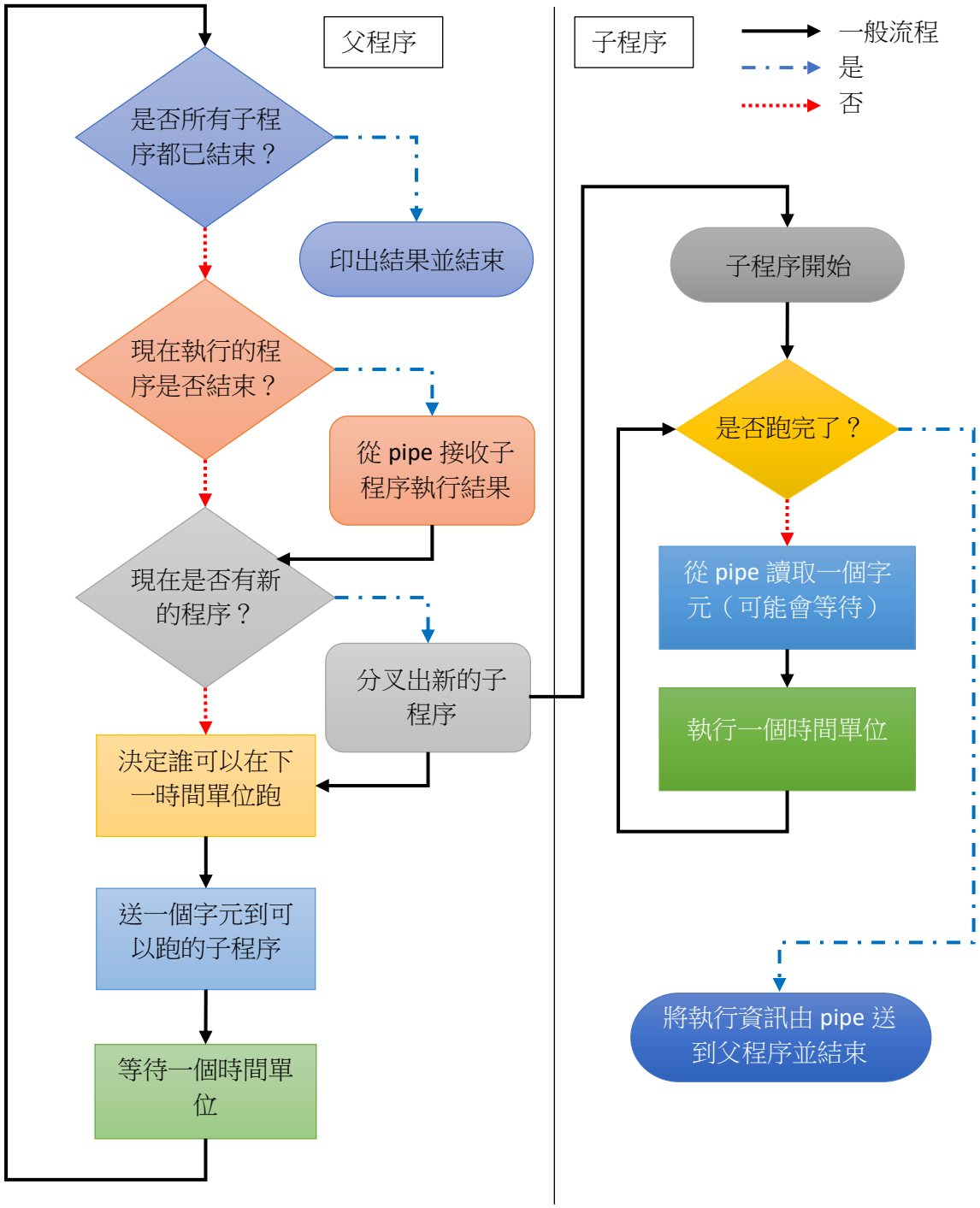
作業系統 Project 1 B07902135 黃申昌

排程系統的設計：

這套排程系統利用了 read 是 blocking system call 的特性。

父程序在生出子程序前，會先建立 pipe 和子程序溝通。當子程序要跑一個時間單位時，它會先從 pipe 讀取一個字元。因為 read 是 blocking system call，子程序會被送到 waiting queue 等待父程序從 pipe 給它一個字元。因此，只要父程序在一個時間單位內，只送字元給一個子程序，在該時間單位內就只有一個子程序可以執行。如此就能達到子程序中同一時間只有一個能跑的效果。



決定誰可以在下一時間單位跑：

這個排程系統支援四種可能的排程機制，而這四種排程機制有不同的方法選擇下一時間單位要讓哪個子程序執行：

- 首先因為 FIFO 和 SJF 是 non-preemptive 的，所以如果已經在執行的子程序還要跑下一時間單位的話，就讓那個子程序繼續跑下去。

接下來：

- 如果是 FIFO，就讓 ready time 最小的子程序執行。
- 如果是 SJF 或 PSJF，就讓 remaining time 最小的子程序執行。其中 remaining time 的計算方法是，一開始所有子程序的 remaining time 會設為其 execution time，而父程序在子程序在跑一個時間單位時，就會將其 remaining time 減一。當現在在執行的程序 remaining time 是 0，就表示子程序該結束了，此時父程序會去 wait 那個子程序，並從 pipe 接收子程序的執行時間。
- 如果是 RR：
    - 維護一個 linked list，當有程序要跑時，就把它加在 linked list 中記錄其編號。
    - 如果現在有程序在執行，但它還沒跑到 500 個時間單位，就讓它繼續跑。
    - 如果現在有程序在執行，而且它跑到 500 個時間單位了，就挑 linked list 的下一個子程序執行。此時紀錄下時間，以便追蹤下一個子程序的執行時間並打斷。

核心版本：

使用的系統是 Ubuntu 20.04，核心版本為 Linux kernel 5.4。在 Oracle VirtualBox 6.1.6 上運行，模擬的機器具有單個 CPU。

討論：

使用 pipe 做排程以及使用 sched_setscheduler 的差異：

- 使用 pipe 做排程的話，所有排程工作都在用戶空間進行，因此程式就只需要 I/O 的系統呼叫，而不需要請 linux 的排程器協助排程。所以不需要自己定義系統呼叫，也就不需要編譯核心（我在做這份 Project 時沒有編譯核心），而且可以在不使用 sudo 也不自己定義系統呼叫下正確地進行排程（但我還是有用 sudo，這樣才能寫東西到 dmesg 去，反正要 dmesg –clear 的話一定要有 sudo 權限）。
- 然而因為所有東西都在用戶空間進行，所以程序的優先度就會和其他無關的程序一樣，而不會像使用 sched_setscheduler 會把程序的優先度拉到其他程序之上。我在程式碼的一開始，有讓程序對自己呼叫 sched_setscheduler，就是為了將自己的優先度拉得比其他程序高，以免和其他與作業無關的程序搶資源。
- 因為卡住子程序的機制是用 pipe 而非 SCHED_IDLE，就算子程序被分發到不同的 CPU 內，它還是會被完全卡住，因為子程序會跑到 waiting queue 中而非 ready queue 中。所以子程序在它從 pipe 收到來自父程序的訊息前，完全沒有被 CPU 選中的機會，也就不需要 sched_setaffinity 了。
- 同理，也不需要擔心父子的時間單位不同步：如果子的時間單位比父快，子在

跑下一時間單位前要先等待父跑完一時間單位並給它訊息；如果父的時間單位比子快，訊息依然會在 pipe 裡面讓子去接收，所以子不會錯過父的訊息。因為父要做的事比子多，所以子通常會跑得比父快，所以使用 pipe 的機制就相當於讓父子在每執行完一次時間單位就做一次同步。

- 然而，因為子程序在每跑一個時間單位前，就要進行一次 I/O，這個方法的效率會較使用 sched_setscheduler 排程來得差。

可能影響執行時間的因素以及如何排除：

- 使用 pipe 的話，如果在某個時間單位中沒有可以跑的子程序，父程序就不用把東西寫進 pipe 內，時間單位的長度就會縮短。為了排除此問題，這個排程器在讀進所有子程序的資訊後，會先 fork 出一個子程序，在沒有可以跑的子程序的時候跑。這個子程序在其他方面都和其他子程序相同，只是其 execution time 是無限，而且只會在沒有可以跑的子程序時被選中。當父程序要結束時，它就會送 SIGKILL 給那個多出的子程序。
- 因為程式的執行環境是在虛擬機上，所以虛擬機也要和 host 上的其他程序搶 CPU 時間，如此一來，就會導致虛擬機每個時間單位的時間會視 host 的繁忙程度而定。

其他注意事項：

- 在實際在跑程式時，會有以「(數字): 」開頭的東西印出來，那些是印到 stderr 的部分。
- 影片的拍攝方式是直接用手機對著螢幕拍，不然如果用電腦自己的螢幕錄影功能的話，會拖慢程序執行的速度。同時，影片有加速 1.5 倍，不然其時間會超過五分鐘。