

Hw1 Report

Q1.

- 先設置變數pid以及其他運算Fibonacci number所需的變數
- 確認有輸入正確數量的argument、argument的數值 ≥ 0
- 使用pid = fork()製造child process，parent process呼叫wait()等待child process完成
- 如果pid = 0為child process、pid = 1為parent process、pid < 0則表示fork()失敗
- 當pid = 0(即child process)進行Fibonacci number的運算
 - 以變數k承接argv[1]的數值（列印前k個Fibonacci number）
 - 開設一個長度為k、變數名稱為fib的陣列用來紀錄Fibonacci number
 - 將fib陣列index_0的數值設為1
 - 以for loop進行計算，從fib陣列index_1開始到index_(k-1)：
 - 根據題目給的公式：(1)先將fib_0+fib_1的結果寫入fib[current_index]
 - (2)將fib_0更新為fib_1的數值
 - (3)將fib_1更新為fib[current_index]
 - 計算完所需的Fibonacci number，將fib陣列中的數值依序印出
- Child process完成後，pid會從0變成1
- 當pid = 1(即parent process)後，程式結束

Q2.

- 首先根據題目做出shared_data的結構
- 確認有輸入正確數量的argument、argument的數值 ≥ 0
- 配置shared memory segment
- 使用指標shared_mem串接配置好的shared memory空間
- 將shared memory的sequence_size數值設置為argv[1]
- 使用pid = fork()製造child process，parent process呼叫wait()等待child process完成
- 設置變數fib_0 = 0、fib_1 = 1作為後續計算Fibonacci number時使用
- 如果pid = 0為child process、pid = 1為parent process、pid < 0則表示fork()失敗
- 當pid = 0(即child process)進行Fibonacci number的運算
 - 將shared_mem指向的shared_data中的fib_sequence的index_0數值設為1
 - 以for loop進行計算，從fib陣列index_1開始到index_(sequence_size-1)：
 - 將fib_0+fib_1的結果寫入fib_sequence[current_index]
 - 將fib_0更新為fib_1的數值
 - 將fib_1更新為fib_sequence[current_index]

- 計算完所需的Fibonacci number，child process完成，pid從0變成1
- 當pid = 1(即parent process)，依序印出fib_sequence中的Fibonacci number
- 釋放shared_mem所指向的空間後，程式結束