Scheduler::Run( Thread \* nextThread, bool finishing)

1. Check if finishing, if true, it destroys the current thread
2. Check if thread is user program, if so, save CPU regs

(main thread的space是null，所以不是user program，不需要保存資料)

1. CheckOverflow()
2. Change to nextThread and set state to Running
3. Call SWITCH to stop current thread and start new thread  
   **SWITCH在switch.S中被Implement  
     
   From** [**https://www.itread01.com/content/1541489407.html**](https://www.itread01.com/content/1541489407.html)主執行緒進入SWITCH最後一句話執行時CPU的返回值指向ThreadRoot函式，子執行緒進入SWITCH最後一句話執行時CPU的返回值指向Scheduler::Run函式中的SWITCH函式語句。
4. Call CheckToBeDestroyed if there is any toBeDestroyed thread, if so, delete **IT**並沒有在第一時間就刪除，是為了完成SWITCH的程序。
5. Try to restore available address spaces. (弄懂thread的RestoreUserState和space的RestoreState)

AddrSpace :: AddrSpace()

1. Create Entry for a new thread in PageTable
2. Set the initial values for the Thread
3. Zero out the entire address space (bzero : set N bytes to zero)

Thread::Fork()

1. Two arguments func and arg, func is the procedure to be forked, arg is the parameter to be passed to the procedure
2. Call StackAllocate with the same arguments
3. Set interrupt to off
4. Put current thread into scheduler for run **(ReadyToRun)**
5. Set interrupt back to the original value

Thread::StackAllocate()

1. Call AllocBoundedArray
2. #ifdef directive **allows for conditional compilation**
3. Sets the correct Machine state

Scheduler::ReadyToRun()

1. Set this thread to Ready state
2. Append it on the ready list ( Fork的時候被呼叫)

Kernel::Exec

1. Create new thread by the 檔案名稱與執行緒編號
2. Allocate the space for phy to virtual memory translation
3. Fork
4. Why is the return value Threadnum -1

(回傳當前占用的最後一個執行緒編號，可用來防止kernel建立過多thread)

(上限10個，見kernel.h，”t[10]”)

Kernel :: ExecAll

1. Call Exec for each file that needs execution
2. Call finish when the loop ends (Finishing = TRUE)

(讓當前的Thread (即main thread) 進入睡眠，以觸發**S**W**I**T**C**H)

AddrSpace:: Execute()

1. Assumes the program is already loaded into the address space
2. Set register by calling Init Registers and RestoreState
3. Call Machine::Run to jump to the user program to run the program.

AddrSpace:: Load

1. Use Filesystem to Open the file

Kernel :: Kernel

1. Use strcmp to compare different input values and do the according actions
2. Execfile[ ++execfilenum ]會記錄 -e 參數的後面一個參數的名稱

Kernel::ForkExecute

1. Call Execute if the executable is found

Thread::CheckOverflow()

1. 如果Stack overflowed，則程式中斷(assertion failed)

SWITCH

1. 以Assembly Language寫成
2. 移動Stack pointer，讓他從舊的thread指向新的 (t1 -> t2)

ThreadRoot

1. 以Assembly Language寫成
2. 會呼叫startup function (即Thread::Begin)和讀取初始參數(在StackAllocate時候被記錄，即ForkExecute和指向該thread的指標)，以及讀取最後該呼叫的Thread::Finish。  
   (示意圖，實際執行情形不一定會是這個 #ifdef 的 #else)一張含有 文字 的圖片

   自動產生的描述
3. 在main thread SWITCH之後，會以此函式為開端，讓OS繼續執行其他thread。(若沒有這個動作，則會讓main thread一路return，從Sleep -> Finish -> ExceAll -> main -> ASSERTNOTREACHED，而中止程式。)

要實作的東西在machine.h和addrspace.cc