Sleep(),Finish(),Yield():Thread.cc

Accumulate time:在running state跑多久

Onetick():interrupt.cc

Ready to Run:

接受Terminal端的參數作為execfile,為每個execfile建立一個thread,宣告一個Address space.Fork這個procedure,其中會set register value和load page table register(Execute()),分配Stack空間(StackAllocate),把interrupt關掉後就把Thread放到Read to Run state.

完成Forked procedure,最後current->Finish(),call Sleep(),放棄CPU.

Run to Ready:

將User Program的一個指令抓過來Decode,執行OneTick(),將System跟User都tick++,CheckIfDue會檢查是否有到期的pending interrupt,Yield()找下一個在ready list的thread(FindNextToRun),並把現在這個執行的thread放回ready list重新排隊(ReadyToRun),並做context switch(Sceduler::Run)

Running to Waiting:

發生在I/O或system call.

Read a character typed at the keyboard, Semaphore::P():類似wait(),semaphore value--, ,會把Thread放進queue中(Append), 並call Sleep()把thread Block住,這時候CPU空掉了,要找下一個在readyqueue的thread(FindNextToRun),找不到就idle,最後分配CPU給下一個在ready queue的thread(Run())

Waiting->Ready:

Semaphore::V() 類似Signal() semaphore value++, 並且把Waiting queue的front thread取出來,放回去Ready queue(ReadyToRun())

Running to Terminating:

Exception Handler的SC\_EXIT 執行Finish(),他會call,Sleep(),並且把finishing設成True,這時候找ready queue最前面的thread(FindNextToRun()),最後分配CPU(Run()),前面的fininshing=True在Run()代表最後會刪除這個Thread

Running to Ready

從Ready queue挑選下一個thread(FindNextToRun()).

Switch.s:

T1 change to T2

t1把所有register相關的資訊儲存起來,t2從memory 把相關Register的資訊Load Register,到達ret時set CPU program counter to the memory address pointed by the value of register esp,也就是抓取ThreadRoot位置,ThreadRoot會call

StartupRC:ThreadBegin()

InitialPC ForkExecute()

WhenDonePC ThreadFinish().

Switch回來後,

如果thread1是被Block的(waiting state)且Finish=True,Run執行完後會把thread1 delete掉,並執行新的user program(execfile),若Finish=false,會挑選下一個在ready queue的thread.

Run():將instruction Decode,並且用OneTick模擬Clock執行時間

Implement:

2-2

Running->Ready 計算時間放在Yield(),因為FindNextToRun已經在Ready

(a):放在ReadyToRun

(b):放在FindNextToRun