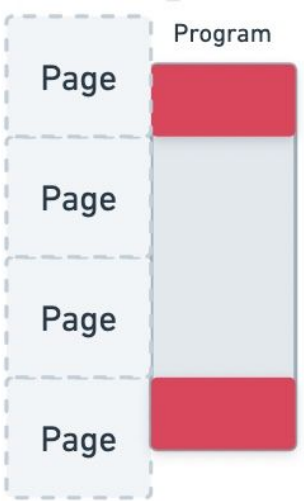
MP2 discussion

Code (Load page)

- fileAddr / vAddr: file address / virtual address of current load
- readSize: the size of current load
- remainSize: the remain size that needs to be loaded
- Need to handle the fragmented page

```
remainSize = noffH.code.size;
fileAddr = noffH.code.inFileAddr;
vAddr = noffH.code.virtualAddr;
readSize = min(PageSize - (vAddr % PageSize), PageSize);
while (remainSize > 0) {
    pAddr = pageTable[vAddr / PageSize].physicalPage * PageSize + (vAddr % PageSize);
    executable->ReadAt(
        &(kernel->machine->mainMemory[pAddr]),
        readSize, fileAddr);
    vAddr += readSize;
    remainSize -= readSize;
    fileAddr += readSize;
    readSize = min(PageSize, remainSize);
}
```



Code (Memory limit exception)

- Notice that you need to comment ASSERT()
  - Insufficient memory for single thread also leads to memory limit exception

Question 1-1

- 請問什麼時候thread會被加到cpu scheduler的readyList？ (至少兩個)
- Thread::Fork, thread剛被建立的時候
- Semaphore::V, semaphore被release的時候
- Thread::Yield
- **Take advantage of search !**

Question 1-2

- 什麼情況下 Sleep() 會 delete thread; 什麼情況不會，請舉例說明。
- Delete: thread->finish() 的時候，因為 thread 已經完成了所以會 delete
- Not delete: 因為在等待硬體 resource 所以去 sleep 的 thread 就不會 delete (E.g. Semaphore::P())

Question 2

- 在 testcase 中，為甚麼兩個 thread 都執行完畢，Nachos 卻不會終止？
- 正常流程下，在所有 thread 都執行結束後，若沒有 pending interrupt 的話，程式在 Interrupt::Idle() 中會呼叫 Halt() 終止 NachOS。
- 然而 testcase 不會終止是因為 Timer 會一直定期 schedule interrupt (Class Timer)

```
Timer::Timer(bool doRandom, CallBackObj *toCall)
{
    randomize = doRandom;
    callPeriodically = toCall;
    disable = FALSE;
    SetInterrupt();
}
```

```
void
Timer::CallBack()
{
    // invoke the Nachos interrupt handler for this device
    callPeriodically->CallBack();

    SetInterrupt(); // do last, to let software interrupt handler
                    // decide if it wants to disable future interrupts
}
```