



深圳大学  
Shenzhen University

# 操作系统

操作系统大作业： part I-2

谭舜泉

计算机与软件学院

# 大作业Part I

- 大作业**Part I** 由六部分组成。六部分均为阅读xv6中文文档，理解对应的xv6源代码，并回答大作业中提出的代码理解问题。
  1. （第二周）第0章：操作系统接口；
  2. （第四周）第1章：第一个进程；
  3. （第六周）附录A/B：PC硬件及引导加载器；
  4. （第八周）第2章：页表；
  5. （第十周）第4章：锁；
  6. （第十二周）第3章：陷入、中断和驱动程序。



# 大作业Part I

- 注意上半学期的大作业分为两部分汇总提交。
- 其中**1-3**提交一份实验报告，而**4-6**提交另一份实验报告。



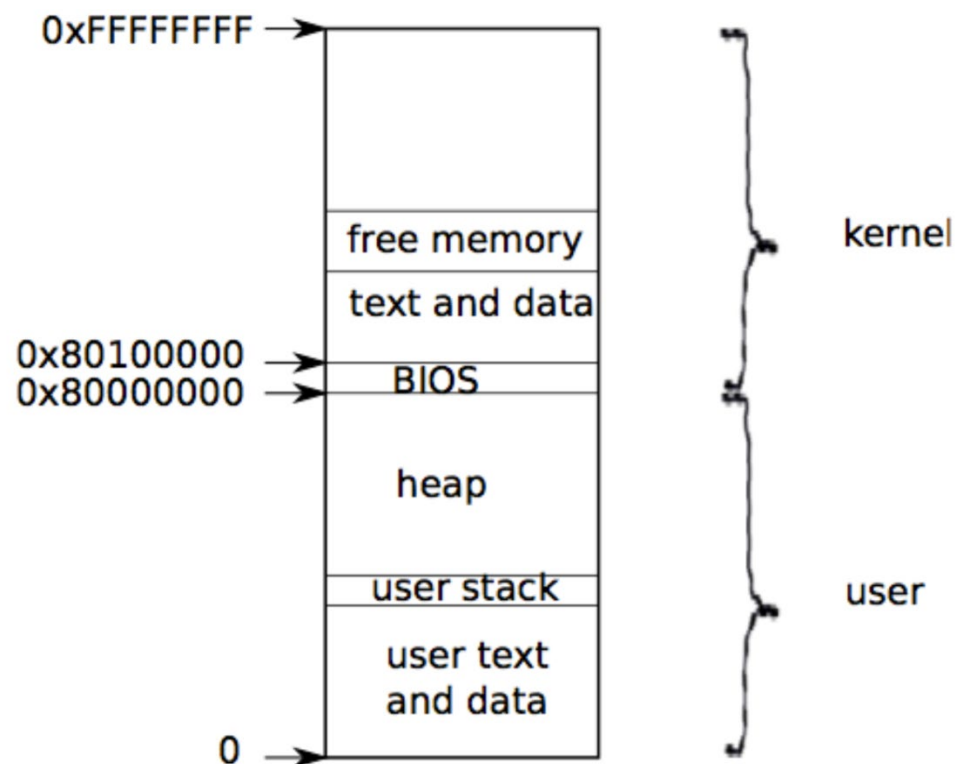
# 实验报告

- 实验报告要求使用深大实验报告模板填写。
- 允许两至三人组成一个小组，撰写实验报告。
- 请按照实验要求，完成实验并撰写实验报告。  
把实验报告电子版上传到**BLACKBOARD**。 本实验分组完成。每个实验小组完成一份实验报告即可。但要求小组中每个成员都把实验报告的副本上传到**BLACKBOARD**，并注明同组成员的名字和学号。



# 第1章 第一个进程

- 进程是一个抽象概念。
  - 它让一个程序可以假设它**独占**一台机器。
  - 进程向程序提供“看上去”**私有的**，其他进程无法读写的（或地址空间）；
  - 以及一颗“看上去”**仅执行该程序的CPU**。



# 第1章 第一个进程

- xv6 为每个进程维护了不同的页表。
- 一片地址空间包含了从虚拟地址0开始的用户内存。
  - 地址最低处放置进程的指令；
  - 接下来则是全局变量， 栈区；
  - 以及一个用户可按需拓展的“堆”区（malloc 用）。
- 内核的指令和数据也会被进程映射到每个进程的地址空间中。
  - xv6 将内核映射到了地址空间的高地址处， 即从 0x80100000 开始。
  - BIOS代码放在0x80000000。



# 第1章 代码：第一个地址空间

- 当 PC 开机时， 它会初始化自己然后从磁盘中载入 boot loader 到内存并运行。
- 然后， boot loader 把 xv6 内核从磁盘中载入并从 entry（1040）开始运行。
- x86 的分页硬件在此时还没有开始工作；所以这时的虚拟地址是直接映射到物理地址上的。
- boot loader 把 xv6 内核装载到物理地址 0x100000 处。



# 第1章 代码：第一个地址空间

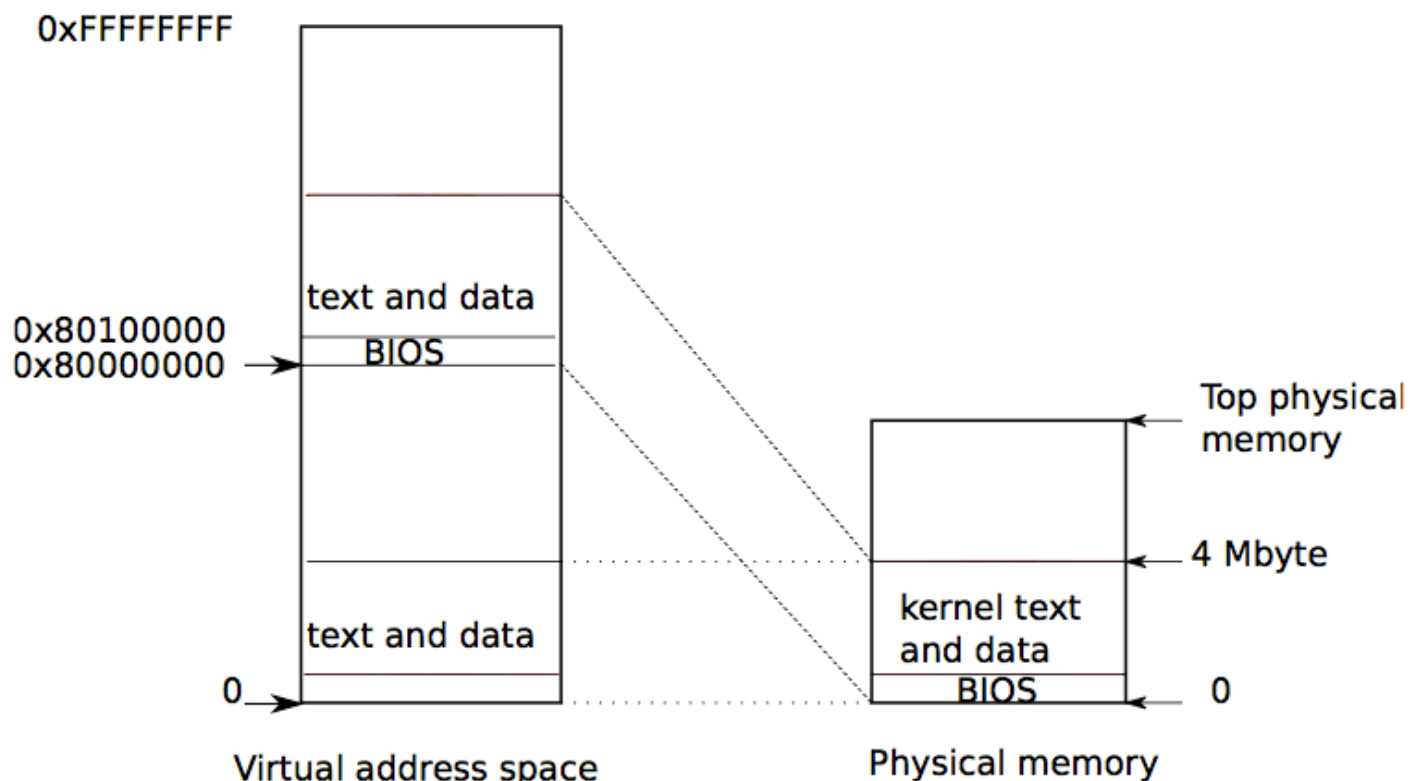


Figure 1-2. Layout of a virtual address space



# 第1章 代码：第一个地址空间

- entry 中的页表的定义在 main.c (1311) 中。 (p15最后一段，读一下)

```
pde_t entrypghdir[NPDENTRIES] = {  
    // Map VA's [0, 4MB) to PA's [0, 4MB)  
    [0] = (0) | PTE_P | PTE_W | PTE_PS,  
    // Map VA's [KERNBASE, KERNBASE+4MB) to PA's [0, 4MB)  
    [KERNBASE>>PDXSHIFT] = (0) | PTE_P | PTE_W | PTE_PS,  
};
```



# 第一章代码：创建及运行第一个进程

- (p16) main->userinit->allocproc (读一下其代码)
- 第一个进程内存中的初始内容是由 **initcode.S** 汇编得到的。
- 在 main 调用了 **userinit** 之后， **mpmain** 调用 **scheduler** 开始运行进程（1267）。
- **initcode.S** 干的第一件事是触发 **exec** 系统调用。



# 代码理解问题

- “xv6 的地址空间结构有一个缺点，即无法使用超过 2GB 的物理 RAM”——请给出解释。为什么xv6的内存空间只能有2GB？
- （p15）“这个映射就限制内核的指令+代码必须在 4mb 以内。”——请给出解释。为什么？
- 请问initcode.S 所触发 exec 系统调用执行了哪个程序，而那个程序又是实现什么功能的呢？



# 代码理解问题

- 这些问题，如果不能理解的话也不用担心。我将在两周后的课堂上（在实验报告提交前）给予解答。

