操作系统xv6概述

介绍：

Xv6是由麻省理工学院(MIT)为操作系统工程的课程（代号6.828）,开发的一个教学目的的操作系统。Xv6是在x86处理器上(x即指x86)用ANSI标准C重新实现的Unix第六版(Unix V6，通常直接被称为V6)。

课程简介：

Xv6是由麻省理工学院(MIT)为操作系统工程的课程（代号6.828）,开发的一个教学目的的操作系统。

Xv6是在x86处理器上(x即指x86)用ANSI标准C重新实现的Unix第六版(Unix V6，通常直接被称为V6)。Unix V6是1975年发布的，基于DEC PDP-11小型机，当时还没有x86系列CPU，而现在PDP的机器已经很少见了，当时使用是在标准ANSI C发布之前的旧式C语言。

与Linux或BSD系统不同，Xv6很简单，可以在一个学期讲完，全部代码只有8千行多，但仍包括了Unix的重要概念和组织结构。由于是基于较早的Unix V6，Xv6的结构与现代操作系统，如Linux，Windows的差距较大。

在MIT以外，很多其它大学也在操作系统课程中使用了Xv6或其变种，如耶鲁，清华等。

另外一个类似的教学用类Unix系统是著名的Minix。

课程教程：

截止2018年9月，Xv6的最新版本是rev8，使用MIT许可证。可以通过git获得最新的源代码

git clone https://github.com/mit-pdos/xv6-public.git

此外还有一份适合打印的pdf文件 ，包括了全部源代码，双面打印的话不到50页。

还有一份Xv6简明的讲义 。

Xv6使用GNU C编译器gcc ，以 ELF可执行文件格式作为支持的可执行文件格式。这有两层含义，一是操作系统的实现代码中有可以识别和执行ELF文件的模块，另一层意思是要有能够生成ELF文件的编译器。如果使用其它的可执行文件格式，就需要重写一个编译器（至少是后端）。使用gcc和ELF可以省下不少工作。

BSD和Linux系统是直接支持这些gcc和ELF的，Mac OS X和Windows 不使用ELF，必须使用交叉编译器才能生成ELF格式可执行文件。

Xv6可以在真实的硬件上启动，但通常使用虚拟机来运行它，如Bochs，QEMU和Vmware player，VirtualBox。

历史背景：

MIT的一个操作系统工程的课程，代号6.828最初使用了Unix v6和John Lion的《Unix V6注释和代码》这本书（通常被称为“狮”书，Lion是作者的姓，这里做为一个双关语，这本书使人们可以得到Unix的源代码）； 而在实验环节，学生们主要是在intel x86 的CPU上实现一个称为Jos的“外核”架构(exokernel)的操作系统。

向学生介绍多个系统（V6，Jos）能帮助他们建立更多操作系统设计的观念。但V6从一开始就导致了教学上的麻烦。很多学生怀疑使用V6这样一个30多年前的，使用旧式C语言（比K&R C还要旧）开发的，并且在过时的PDP-11硬件上运行的系统是否合适。而且学生们还要苦于同时学习PDP-11和Intel x86两种不同架构的底层差别。于是在2006年夏天，课程的老师们决定以V6为基础，使用ANSI C写一个新的在Intel x86多处理器计算机上的系统，也就是Xv6，来代替V6。Xv6使用x86 CPU，更适合学生的经验，而且将6.828课程统一到单一的体系结构上。增加多处理器支持需要使用锁和线程来处理并发（单处理器只要简单的开/关中断即可）。重写一个新系统的过程中，也使V6中比较粗糙的一些部分，如调度和文件系统得到了改进。于是MIT在2006年秋季的6.828课程中就开始使用Xv6了。