به نام خدا

پروژه اول آزمایشگاه سیستم عامل

محمد سعادتی پویا صادقی پاییز 1402

'XV6 چیست؟

- 1. یک سیستم عامل سبک
- 2. طراحی شده در MIT برای اهداف آموزشی
 - 3. دارای ساختار آسان و قابل فهم

مراحل پروژه

- 1. چند سوال در ابتدا برای آشنایی شما با مفاهیم کلی
 - 2. نصب و راه اندازی xv6
 - .. پیاده سازی یک قابلیت دستگرمی
 - 4. اضافه کردن چند قابلیت به کنسول
 - 5. اجرای برنامه سطح کاربر
 - 6. آشنایی با اشکال زدایی

| xv6 چگونه اجرا می شود؟|

- 1. xv6 می تواند به عنوان یک سیستم عامل جداگانه اجرا شود. (پیشنهاد نمی شود)
 - 2. xv6 می تواند در محیط یک emulator اجرا شود. (پیشنهاد می شود)

از Qemu به عنوان emulator برای اجرای xv6 استفاده می کنیم.

نصب امولاتور Qemu

/https://www.qemu.org/download



Download QEMU

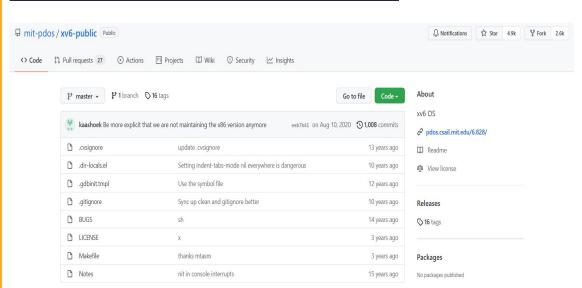
Source code Linux macOS Windows

QEMU is packaged by most Linux distributions:

- Arch: pacman -S qemu
- Debian/Ubuntu: apt-get install gemu
- Fedora: dnf install @virtualization
- Gentoo: emerge --ask app-emulation/qemu
- RHEL/CentOS: yum install qemu-kvm
- SUSE: zypper install qemu

اجرای سیستم عامل xv6

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



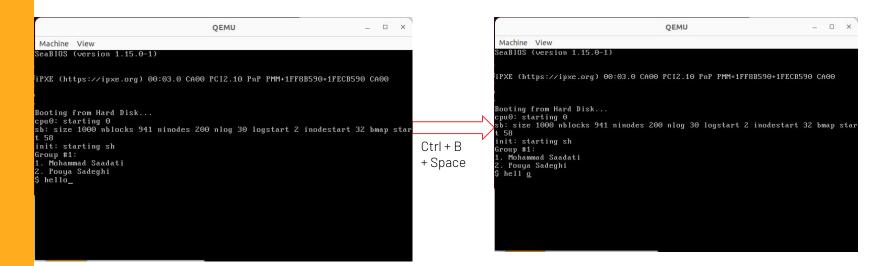
چاپ نام اعضای گروه

```
QEMU - Press Ctrl+Alt+G to release grab
Machine View
SeaBIOS (version 1.15.0-1)
iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B590+1FECB590 CA00
Booting from Hard Disk...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
Group #1:
1. Mohammad Saadati
2. Pouya Sadeghi
```

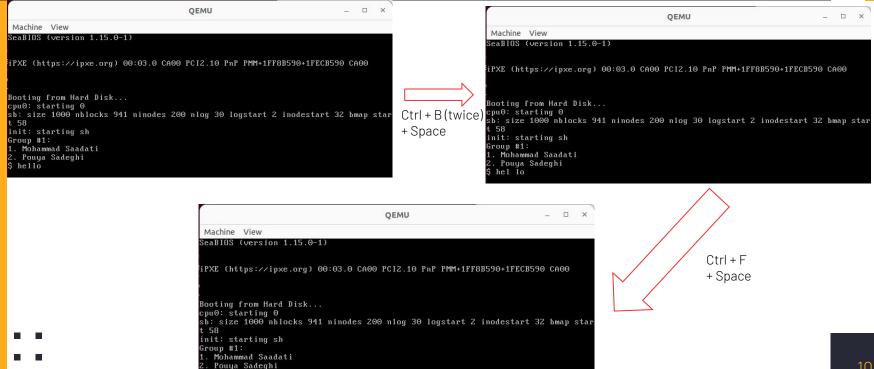
قابلیت های اضافه شده به کنسول

- (چپ) انتقال اشاره گریک خانه به سمت عقب (پپ) $\operatorname{Ctrl} + \operatorname{B}$
- انتقال اشاره گریک خانه به سمت جلو (راست) $\operatorname{Ctrl} + \operatorname{F}$
 - ایاک کردن صفحه ترمینال: Ctrl + L
 - 4. Arrow UP: دستور وارد شده قبلی (در صورت وجود)
- 5. Arrow DOWN: دستور وارد شده بعدی (در صورت وجود)

CTRL + B

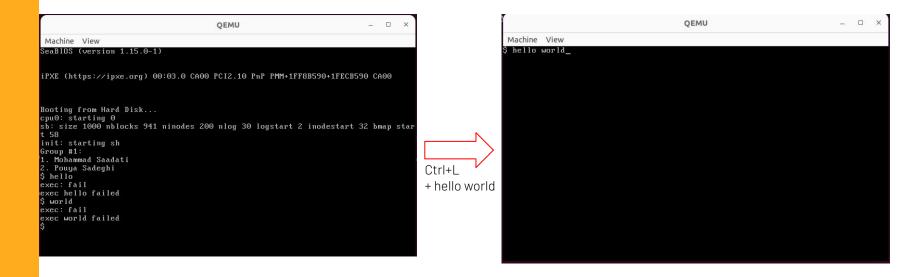


CTRL + F



hel I o

CTRL + L



Arrow UP

trdiff

console cat README 2 18 16008

```
pouya@pouya-LP5: ~/Desktop/OS-Lab1-Fall2023 Q = _ _ _ _
                                            SeaBIOS (version 1.15.0-1)
                                           iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B4A0+1FECB4A0 CA00
                                           Booting from Hard Disk..xv6...
1. cat README specific from Hard DISK...XVo...

COLUMN STATTING 0

Sch: Stze: 1000 nblocks 941 nlnodes 200 nlog 30 logstart 2 inodestart 32 bmap stab
                                             nit: starting sh
                                            Group #1:
                                              Mohammad Saadati
                                             . Pouya Sadeghi
                                             cat README
```

2. ls

```
pouya@pouya-LP5: ~/Desktop/OS-Lab1-Fall2023 Q = _ 0
He are also grateful for the bug reports and patches contributed by Silas
Boyd-Wicklzer, Anton Burtsev, Cody Cutler, Mike CAT, Tej Chajed, eyal2800,
Nelson Elhage, Saar Ettinger, Alice Ferrazzi, Nathaniel Filardo, Peter
FroeNtLich, Yakir Gosron, Shiven Handa, Bryan Henny, Jis Huang, Alexander Kapshuk, Anders Kaseora, kehangis, Nolfang Keiler, Eddle Khiler, Austin Litew, Imbar Marinescu, Yandong Mao, Matan Shabtay, Hitoshi Mitake, Carni Herimovich, Mark Morrissey, masan, Joak Nider, Greg Price, Ayan Shafqat, Eldar Sehayek, Yongning Shen, Can Tenny, Tyfkda, Rafael Ubal, Narren Tooney, Stephen Tu, Pablo Ventura, Xi Wang, Ketichi Watanabe, Nicolas
   plovick, wxdao, Grant Wu. Jindong Zhang, Icenowy Zheng, and Zou Chang Wei.
 The code in the files that constitute xv6 is
   opyright 2006-2018 Frans Kaashoek, Robert Morris, and Russ Cox.
   e don't process error reports (see note on top of this file).
 BUILDING AND RUNNING XV6
 To build xv6 on an x86 ELF machine (like Linux or FreeBSD), run
  make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries (see hitps://pdos.csail.mit.edu/6.828/).
Then run "make TOOLPEFIX=1386-jos-elf-". Now install the QEMU PC
   lmulator and run "make gemu".$ ls
```

pouya@pouya-LP5: ~/Desktop/OS-Lab1-Fall2023 Q = 'make". On non-x86 or non-ELF machines (like OS X, even on x86), you will need to install a cross-compiler gcc suite capable of producing x86 ELF binaries (see https://pdos.csail.mit.edu/6.828/). Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC simulator and run "make gemu".\$ ls README 2 2 2286 2 3 15476 forktest 2 5 8804 2 6 18320 grep init kill 2 7 15060 2 8 14444 2 9 14340 2 10 16908 mkdir 2 12 14448 2 13 28504 2 14 15376 stressfs usertests 2 15 62876 2 16 15904 2 17 14024 strdiff 2 18 16008 3 19 0

pouya@pouya-LP5: ~/Desktop/OS-Lab1-Fall2023 Q = 'make". On non-x86 or non-ELF machines (like OS X, even on x86), you will need to install a cross-compiler gcc suite capable of producing x86 ELF binaries (see https://pdos.csail.mit.edu/6.828/). Then run "make TOOLPREFIX=i386-ios-elf-". Now install the OEMU PC simulator and run "make gemu".\$ ls FADME 2 3 15476 2 4 14360 forktest 2 5 8804 2 6 18320 grep init kill 2 7 15060 2 10 16908 mkdir 2 11 14468 2 12 14448 2 13 28504 stressfs 2 14 15376 sertests 2 15 62876 2 16 15904 zombie

3. Arrow UP

4. Arrow UP

Arrow DOWN

```
pouya@pouya-LP5: ~/Desktop/OS-Lab1-Fall2023 Q =
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries (see https://pdos.csail.mit.edu/6.828/).
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
simulator and run "make qemu".$ ls
               1 1 512
               1 1 512
README
               2 2 2286
cat
               2 3 15476
echo
               2 4 14360
forktest
               2 5 8804
дгер
               2 6 18320
init
               2 7 15060
kill
               2 8 14444
ln
               2 9 14340
               2 10 16908
mkdir
               2 11 14468
               2 12 14448
               2 13 28504
stressfs
               2 14 15376
usertests
               2 15 62876
               2 16 15904
zombie
               2 17 14024
strdiff
               2 18 16008
console
               3 19 0
S cat README
```

```
pouya@pouya-LP5: ~/Desktop/OS-Lab1-Fall2023
"make". On non-x86 or non-ELF machines (like OS X, even on x86), you
will need to install a cross-compiler gcc suite capable of producing
x86 ELF binaries (see https://pdos.csail.mit.edu/6.828/).
Then run "make TOOLPREFIX=i386-jos-elf-". Now install the QEMU PC
simulator and run "make gemu".$ ls
               1 1 512
               1 1 512
README
              2 2 2286
cat
               2 3 15476
echo
               2 4 14360
forktest
               2 5 8804
дгер
               2 6 18320
init
               2 7 15060
kill
               2 8 14444
               2 9 14340
               2 10 16908
mkdir
               2 11 14468
               2 12 14448
               2 13 28504
stressfs
               2 14 15376
usertests
               2 15 62876
               2 16 15904
zombie
               2 17 14024
strdiff
               2 18 16008
console
              3 19 0
S ls
```

previous state



اجرای برنامه سطح کاربر

```
QEMU
 Machine View
SeaBIOS (version 1.15.0-1)
iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1FF8B590+1FECB590 CA00
Booting from Hard Disk...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
 58
init: starting sh
Group #1:
1. Mohammad Saadati
2. Pouya Sadeghi
 strdiff apple banana
 cat strdiff result.txt
100011
```

GDB

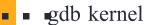
ساده ترین راه برای اشکال زدایی xv6 تحت Qemu استفاده از ویژگی اشکال زدایی از راه دور GDB است.

اشکال زدایی از راه دور یک تکنیک بسیار مهم برای توسعه هسته به طور کلی است: ایده اصلی این است که اشکال زدای اصلی (در این مورد GDB) جدا از برنامه در حال اشکال زدایی اجرا می شود.

برای اجرای xv6 تحت Qemu و فعال کردن اشکال زدایی از راه دور، عبارت زیر را تایپ کنید:

make qemu-gdb

برای راه اندازی اشکال زدا و اتصال آن به سیستم رفع اشکال از راه دور Qemu، یک پنجره ترمینال جداگانه باز کنید، به دایرکتوری xv6 رفته و تایپ کنید:



پس از اتصال موفقیت آمیز GDB به بخش اشکال زدایی از راه دور Qemu ، اطلاعات مربوط به محل توقف برنامه از راه دور را بازیابی و نمایش می دهد:

The target architecture is assumed to be i8086

[f000:fff0] 0xffff0: ljmp \$0xf000,\$0xe05b

0x0000fff0 in ?? ()

پیکر بندی و ساختن هسته لینوکس (امتیازی)

هسته لینوکس پایه و اساس سیستم عامل های یونیکس مانند است. هسته، مسئول ارتباط بین سخت افزار و نرم افزار و تخصیص منابع موجود است. تمام توزیع های لینوکس بر اساس یک هسته از پیش تعریف شده است اما اگر می خواهید چندین گزینه و درایور را غیرفعال کنید یا patch های آزمایشی را امتحان کنید، باید یک هسته لینوکس بسازید.

کرنل لینوکس را در محیط سیستم عامل build کنید و سپس روی یک ماشین مجازی اجرا کنید.

پس از اجرای دستورات نوشته شده در فایل Linux Kernel.pdf دستور printk را در کد قرار دهید و نام اعضای گروه را چاپ کنید. این متن باید در خروجی دستور dmesg قابل مشاهده باشد. خروجی این دستور را در یک فایل قرار داده و در کنار فایل های دیگر پروژه آپلود کنید.

گزارش کار

گزارش کار شما باید شامل موارد زیر باشد:

- 1. پاسخ به تمامی سوالات مطرح شده
 - 2. تصاویر مرتبط از اجرای کدها

تحويل پروژه

- 1. همه اعضای گروه باید در تحویل پروژه حضور داشته و به تمامی قسمت ها مسلط باشند.
 - 2. نمره افراد یک گروه لزوما با هم یکسان نخواهد بود

انكات پاياني

- 1. سعی کنید هرچه زودتر پروژه را شروع کرده و انجام آن را به روزهای آخر موکول نکنید تا اگر به مشکلی برخوردید بتوانید به موقع آن را برطرف کنید.
 - 2. هرگونه سوال یا اشکال خود را می توانید از ما بپرسید و از راه های زیر با ما در ارتباط باشید:

mohammad.saadati80@gmail.com

pouyasadeghi2012@gmail.com

موفق باشيد!