

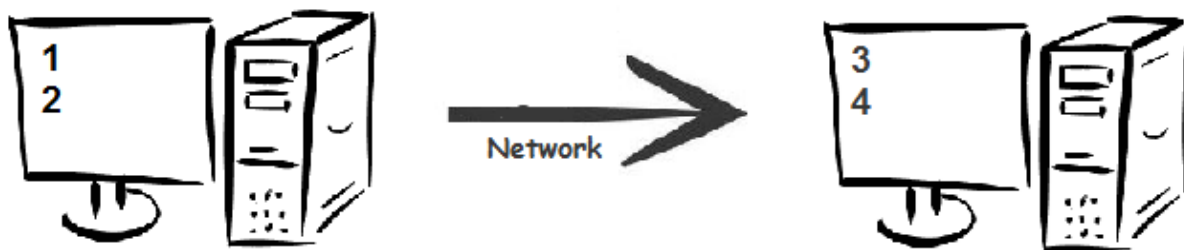
پروژه درس سیستم عامل

موعد تحویل: ۱ بهمن ماه

- این پروژه را به صورت گروه‌های دو نفره انجام دهید.
- برای آپلود پروژه، ابتدا در پوشه مورد نظر دستور `make clear` را اجرا کنید و سپس کل فایل‌ها را در یک فایل فشرده کرده و با قالب `name_studentNumber_project.zip` آپلود کنید.
- دقت کنید که اسم تمام فایل‌هایی که تغییر دادی را در فایلی به اسم `project_changes.log` ذخیره کنید و همراه پروژه آپلود کنید. فقط اسم فایل‌ها (هر اسم در یک خط) باید در این فایل باشد.
- **توجه:** تمامی اعضای گروه باید در کد زدن و انجام پروژه مشارکت داشته باشند.

مهاجرت فرآیندها (Process Migration)

هدف از انجام این پروژه اضافه کردن امکانی به سیستم عامل xv6 است که توانایی انتقال process ها را از طریق شبکه به یک سیستم دیگر (پیاده سازی شبکه امتیازی است) امکان پذیر کند. در واقع، کد شما باید در دو بخش جداگانه (۱) امکان ذخیره وضعیت یک process در حال اجرا در یک فایل و (۲) بارگزاری و اجرا وضعیت process ذخیره شده در فایل از لحظه‌ای که توسط دستور قبلی متوقف شده بود، را برای سیستم عامل فراهم کند. با استفاده از این کد می‌توانید process ها را توسط شبکه برای کامپیوترهای دیگر ارسال کنید.



شکل ۱

توجه: برای انجام این پروژه حتما باید سیستم عامل لینوکس را بر روی سیستم خود نصب کنید (روند پیشنهادی بر روی سیستم عامل Ubuntu 15.04 64-bit تست شده است).

سیستم عامل xv6 چیست؟

سیستم عامل xv6 یک سیستم عامل آموزشی است که در تابستان سال ۲۰۰۶ برای درس سیستم عامل دانشگاه MIT ساخته شد. اگر چه نسخه‌های مختلفی از لینوکس هستند که کد kernel آنها open source است، برای پیاده سازی این پروژه به یک kernel که در کنار accessible بودن، ساده باشد و دارای کامنت باشد نیاز دارید. با توجه به این ویژگیها سیستم عامل xv6 بهترین گزینه برای پروژه شما است. نمونه مدل شده Unix V6 هست.



شکل ۲

اطلاعات بیشتر در مورد xv6 : <https://pdos.csail.mit.edu/6.828/2014/xv6.html>

سیستم عامل xv6 را از این لینک دانلود کنید : <https://github.com/mit-public-pdos/xv6>
بعد از اینکه که کد سیستم عامل را دانلود کردید، باید آنرا اجرا کنید. بنابراین ابتدا کد را در یک مکان مناسب extract کنید و سپس با استفاده از یک terminal و دستورات زیر می توانید آنرا اجرا کنید.
بدین منظور ابتدا باید کد را کامپایل کنید:

make

اگرچه اجرا کردن این سیستم عامل بر روی لپتاپ و کامپیوتر امکان پذیر است، ولی به طور معمول برای اجرا کردن آن از شبیه ساز QEMU استفاده می کنند. برای این کار دستور

make qemu

را اجرا کنید. توجه کنید که قبل از آن باید QEMU را نصب کرده باشید.

QEMU چیست؟

به زبان ساده، QEMU یک emulator سریع و مدرن است! برای استفاده از این emulator در این پروژه، از patch فراهم شده توسط دانشگاه MIT استفاده می کنیم. برای این کار، ابتدا بایستی دستورات زیر را اجرا کنید که کد QEMU بر روی سیستم شما دانلود شود (حجم دانلود: حدود ۱۰۰ مگابایت)

```
git clone https://github.com/geofft/qemu.git -b 6.828-1.7.0
```

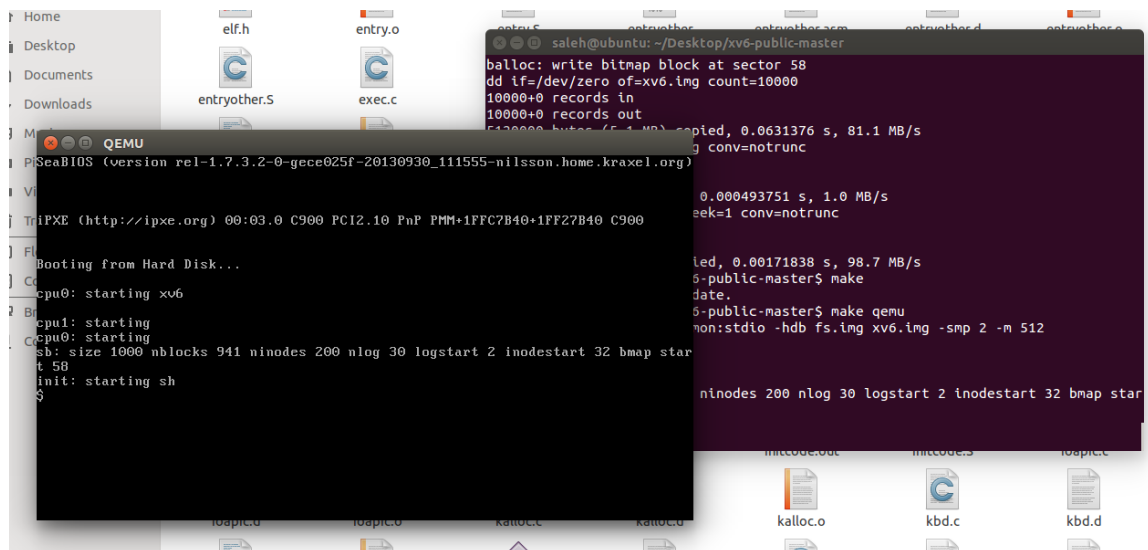
برای مشاهده صفحه گرافیکی، باید بسته libSDL1.2-dev نصب شده باشد (اسم این package در نسخه های متفاوت سیستم عامل لینوکس متفاوت است). برای نصب این بسته دستور زیر را اجرا کنید:

```
sudo apt-get install libSDL1.2-dev
```

برای کامپایل کردن کد QEMU دستورات زیر را وارد کنید:

```
cd qemu
./configure --disable-kvm --target-list="i386-softmmu x86_64-softmmu"
make -j4
sudo make install
```

در این نقطه QEMU بر روی سیستم شما نصب شده است و با مراجعه به پوشه ای که کد سیستم عامل در آن قرار دارد و اجرا کردن دستور `make qemu` سیستم عامل اجرا می شود.



نحوه نوشتن کد

برای پیاده سازی این کد می بایست ابتدا اطلاعاتی از process مورد نظر را ذخیره کنید (برای مثال مقدار (Program Counter) PC و ..) و در اجرا کردن مجدد، این مقادیر را دوباره بارگزاری کنید که برنامه از موقعیت قبلی به کار خود ادامه دهد. (برای پیاده سازی می توانید از کد تابع fork کمک بگیرید.)

اطلاعات process ها در فایل proc.h قرار دارد. بعضی از این اطلاعات باید برای نگه داشته شدن state برنامه و فضایی از RAM که در اختیار این برنامه قرار داده شده ذخیره شود و سپس هنگامی که لازم است با یک دستور بر روی سیستم عامل بارگزاری شود.

دقت کنید که با توجه به اینکه سیستم عامل xv6 پیاده سازی ای برای شبکه ندارد، پیاده سازی کردن شبکه برای یک سیستم عامل از شما خواسته نشده است، شما فقط باید دو برنامه بنویسید که با یکی وضعیت process را در یک فایل ذخیره شود و process بسته شود و با دیگری مجددا وضعیت آن از فایل بارگزاری گردد و process فعالیتش را ادامه دهد.

Process مورد نظر برای تست این کد برنامه های است که از ۱ شروع کند و با فاصله یک ثانیه مقدار متغیر را افزایش داده و در خروجی چاپ کند.