

با استفاده از [مونو](#) امکان اجرای برنامه‌های دات نت تحت لینوکس وجود دارند. در ادامه سعی خواهیم کرد تا [نگارش 3 آن را](#) بر روی اوبونتو نصب کنیم. مونو 3 تا [دات نت 4 و نیم](#) را پشتیبانی می‌کند.

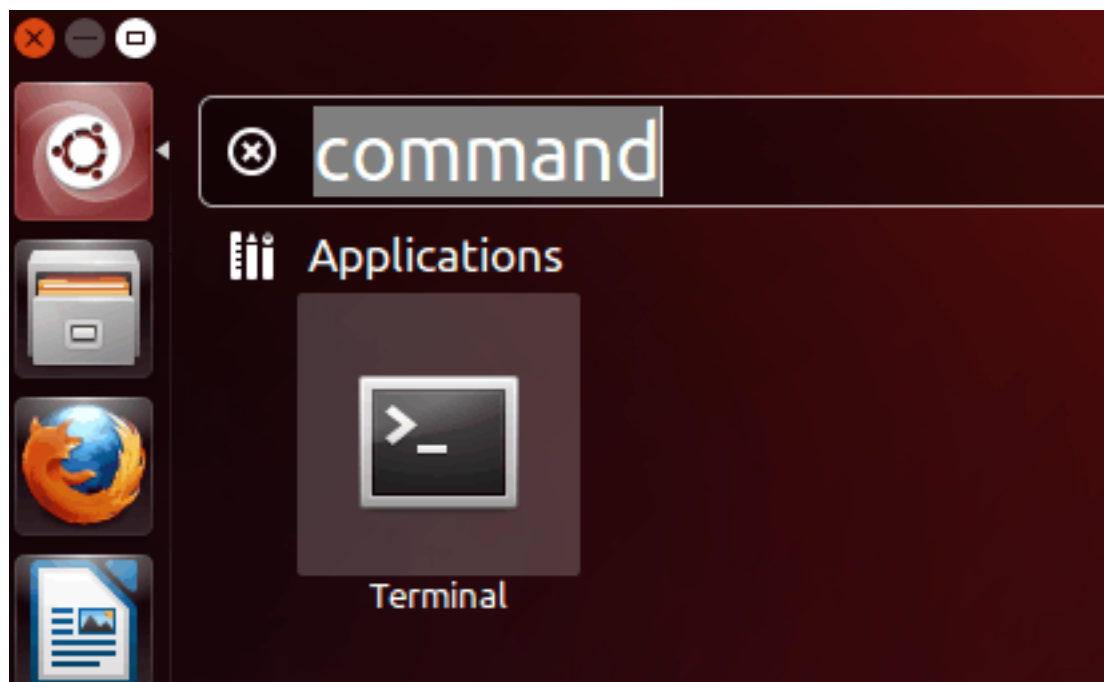
دریافت اوبونتو

برای دریافت اوبونتو به آدرس ذیل مراجعه نمائید.

<http://www.ubuntu.com/download>

نسخه سرور آن GUI ندارد (هرچند بعداً در طی یک بسته 450 مگابایتی قابل نصب است). نسخه دسکتاپ آن به همراه GUI نیز هست. البته برای نصب دات نت بر روی آن این مساله تفاوتی نمی‌کند. برای نصب آزمایشی و مجازی آن هم می‌توانید برای مثال از [VMWare workstation](#) استفاده کنید؛ بدون اینکه نیاز داشته باشید این توزیع خاص لینوکس را واقعاً بر روی کامپیوتر خود نصب کنید.

در تمام قسمت‌های ذیل فرض بر این است که ترمینال خط فرمان لینوکس را گشوده‌اید و همچنین سیستم به اینترنت وصل است.



دریافت Git و Curl

ابتدا دستور زیر را در خط فرمان لینوکس اجرا کنید تا سیستم بسته‌های لینوکس به روز شده و همچنین یک سری پیشنیاز مانند git, curl و امثال آن نصب شوند (کتابخانه curl جهت استفاده در محیط‌های برنامه نویسی کاربرد دارد و همچنین برنامه پیشرفته‌ای است برای کار با وب و دریافت فایل‌ها):

```

vahid@ubuntu: ~
vahid@ubuntu:~$ sudo apt-get update && sudo apt-get -y install git-core curl python-software-properties

```

```

sudo apt-get update && sudo apt-get -y install git-core curl python-software-properties
sudo apt-get install build-essential automake checkinstall intltool git

```

نصب آخرین نگارش Mono و وابستگی‌های آن

در ادامه نوبت به نصب آخرین نگارش مونو است. از روش متداول ذیل برای نصب مونو استفاده نکنید :

```

sudo apt-get install mono-complete

```

این دستور نگارش 2.10.8.1 را تا این تاریخ بر روی سیستم شما نصب خواهد کرد و اگر پیشتر مونو را به این روش نصب کرده‌اید، با استفاده از دستور ذیل آن را حذف کنید:

```

sudo apt-get purge mono-runtime

```

برای دسترسی به آخرین نگارش نگارش مونو، نیاز است آن را از روی سورس آن کامپایل کنیم. اسکریپت کامل نصب آن را در آدرس ذیل می‌توانید پیدا کنید:

<https://github.com/nathanb/iws-snippets/tree/master/mono-install-scripts/ubuntu>

و یا اگر آدرس فوق برقرار نبود از اینجا: [install_mono-3.0-sh](#)

برای نمونه جهت نصب mono نگارش 3 از اسکریپت install_mono-3.0.sh به نحو ذیل استفاده خواهیم کرد (این دستورات را به ترتیب در ترمینال لینوکس اجرا کنید):

```

mkdir mono-3.0
cd mono-3.0
wget --no-check-certificate https://github.com/nathanb/iws-snippets/raw/master/mono-install-scripts/ubuntu/install_mono-3.0.sh
chmod 755 install_mono-3.0.sh
./install_mono-3.0.sh

```

این پروسه مدتی طول خواهد کشید (تا تمام بسته‌های لازم از اینترنت دریافت شوند). استفاده از اسکریپت فوق کار را بسیار ساده کرده و بسیاری از مراحل لازم نصب مونو را یکجا در خود به همراه دارد. مونو 3 تا دات نت 4 و نیم را پشتیبانی می‌کند.



بعد از اجرای فرمان فوق به خطای ذیل خواهید رسید:

```
config.status: error: cannot find input file: `po/mcs/Makefile.in.in'
```

این مورد مشکلی است که در نگارش 3.0.10 رخ داده و [فراموش کرده‌اند](#) که یک پوشه را کپی کنند (در نگارش‌های قبلی آن این مشکل وجود نداشته و با توجه به آگاه شدن از آن، در نگارش‌های بعدی نیز نباید مشکلی باشد). برای رفع آن ابتدا به مسیر ذیل وارد شوید (پوشه build/mono-3.0.10/po)، فایل mcs را حذف (این مورد در اصل یک پوشه است و نه یک فایل) و سپس بسته اصلی mono را از github دریافت کنید. آنرا unzip کرده و کل پوشه mcs داخل آنرا به درون پوشه po جاری کپی کنید. سپس فایل zip دریافت شده را حذف کنید:

```
cd mono-3.0/build/mono-3.0.10/po
rm mcs
wget https://github.com/mono/mono/archive/master.zip
unzip master.zip
mv mcs/ mono-3.0/build/mono-3.0.10/po
```

```
rm -rf mono-master master.zip
```

البته برای اینکه وقت شما زیاد تلف نشود، پوشه mcs نگارش 3.0.10 را از آدرس ذیل دریافت و پس از unzip درون پوشه mono-3.0/build/mono-3.0.10/po کپی کنید. (6 سطر فوق هم نیازی به اجرا ندارند)

[mcs.zip](#)

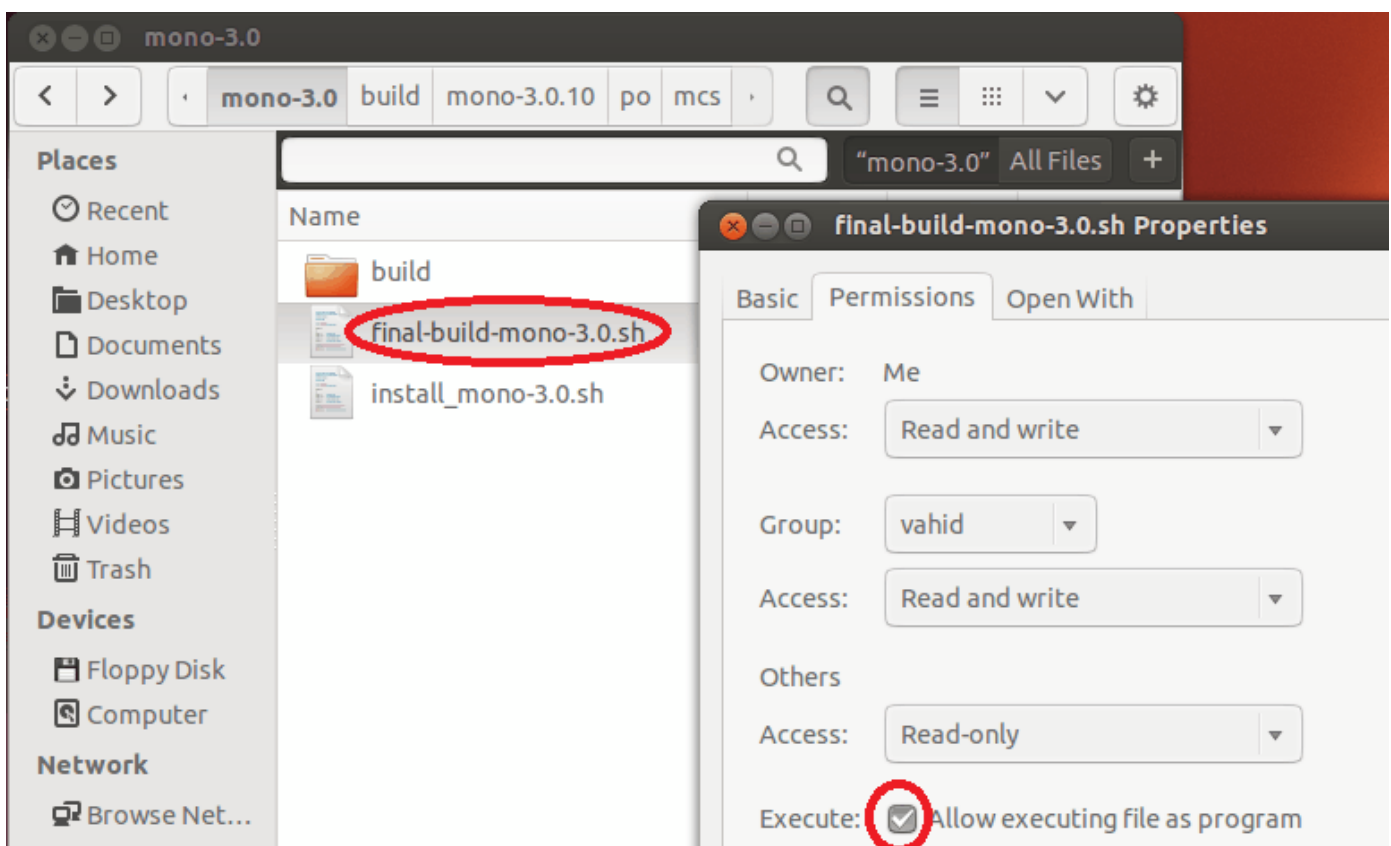
پس از باز سازی پوشه مفقود mcs، باید مرحله «building mono packages» موجود در فایل install_mono-3.0.sh اجرا شود. برای این منظور، فایل final-build-mono-3.0.sh را از آدرس ذیل دریافت و در کنار فایل install_mono-3.0.sh موجود کپی کنید.

[final-build-mono-3.0.sh](#)

سپس در خواص این فایل، مجوز execute را نیز فعال نمائید. اکنون آنرا اجرا کنید:

```
./final-build-mono-3.0.sh
```

فایل final-build-mono-3.0.sh در حقیقت همان فایل install_mono-3.0.sh اصلی است. با این تفاوت که قسمت ابتدای فایل که در آن وابستگی‌های لازم از اینترنت دریافت می‌شدند، حذف شده است. چون پیشتر اینکار را انجام داده بودیم (با اجرای اولیه آن).



اکنون مدتی صبر کنید تا کار کامپایل نهایی تمام بسته‌های دریافت شده پس از اجرای اسکریپت final-build-mono-3.0.sh انجام شود.

آزمایش Mono نصب شده

برای اینکه مطمئن شویم، Mono درست نصب شده است، دستور زیر را در خط فرمان صادر کنید:

```
/opt/mono-3.0/bin/mono -V
```

```

vahid@ubuntu: ~
vahid@ubuntu:~$ /opt/mono-3.0/bin/mono -V
Mono JIT compiler version 3.0.10 (tarball Wed Sep 18 03:12:32 PDT 2013)
Copyright (C) 2002-2012 Novell, Inc, Xamarin Inc and Contributors. www.mono-project.com
  TLS:             __thread
  SIGSEGV:         altstack
  Notifications:   epoll
  Architecture:    x86
  Disabled:        none
  Misc:            softdebug
  LLVM:            supported, not enabled.
  GC:              Included Boehm (with typed GC and Parallel Mark)
vahid@ubuntu:~$
```

برای اینکه این مسیر را به Path لینوکس اضافه کنیم تا قادر شویم فرمان mono را در هر مسیری اجرا کنیم، ابتدا دستور ذیل را اجرا کرده

```
sudo nano /etc/environment
```

و سپس در ادیتور باز شده، مسیر و عبارات ذیل را به انتهای مقدار جاری اضافه کنید:

```
:/opt/mono-3.0/bin
```

```

GNU nano 2.2.6      File: /etc/environment      Modified
social/games:/opt/mono-3.0/bin"
```

بعد از `ctrl+x` را زده، به پیام ذخیره سازی تغییرات پاسخ مثبت دهید. سپس نیاز است یکبار `login` و `logout` کنید تا این تغییرات اعمال شوند.

یک نکته تکمیلی:

اگر به [صفحه نگارش‌های رسمی Mono 3.x](http://download.mono-project.com/sources/mono) مراجعه کنید، نگارش‌های جدیدتری را نیز می‌توانید ملاحظه کنید. فایل‌های قابل نصب آن‌ها نیز در آدرس‌های ذیل قرار دارند:

<http://origin-download.mono-project.com/sources/mono>

<http://download.mono-project.com/sources> برای استفاده از اسکریپت `install_mono-3.0.sh` با این نگارش‌های جدیدتر فقط

کافی است تعاریف ذیل را بر اساس شماره نگارش بسته‌های جدید اصلاح کنید:

```
PACKAGES=("mono-3.0.10"  
"libgdiplus-2.10.9"  
"gtk-sharp-2.12.11"  
"xsp-2.10.2"  
"mod_mono-2.10")  
  
URLS=("http://download.mono-project.com/sources/mono/mono-3.0.10.tar.bz2"  
"http://download.mono-project.com/sources/libgdiplus/libgdiplus-2.10.9.tar.bz2"  
"http://download.mono-project.com/sources/gtk-sharp212/gtk-sharp-2.12.11.tar.bz2"  
"http://download.mono-project.com/sources/xsp/xsp-2.10.2.tar.bz2"  
"http://download.mono-project.com/sources/mod_mono/mod_mono-2.10.tar.bz2")
```

نظرات خوانندگان

نویسنده: سیروان عقیفی
تاریخ: ۲۱:۱۴ ۱۳۹۲/۰۶/۲۹

ممنون، ظاهراً باید دیگه به سمت لینوکس مهاجرت کنیم. یه سوال: آیا در مد GUI با رفتن به قسمت Software Center>Development>IDE و نصب MonoDevelop.NET پیش نیاز هایی که عنوان کردید رو نصب نمیکنه؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۰ ۱۳۹۲/۰۶/۲۹

اینکار رو انجام ندید. نصب MonoDevelop از طریق مرکز نرم افزار آن یا صدور فرمان `sudo apt-get install monodevelop` نگارش 3 این برنامه را نصب خواهد کرد (نگارش جاری آن تا این تاریخ 4 است) و همچنین مونوی قدیمی را هم به اجبار اضافه می کند (مونوی نگارش 2).

نویسنده: رضا خان میرزائی
تاریخ: ۲۲:۴ ۱۳۹۲/۰۶/۲۹

سلام
آیا با 4 monodevelop می توان برای 3 mono (به خصوص برنامه های 4 asp.net mvc) برنامه نویسی کرد ؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۱۳ ۱۳۹۲/۰۶/۲۹

[بله؛ پشتیبانی می شود.](#)

نویسنده: محمد نظری
تاریخ: ۲۳:۱۳ ۱۳۹۲/۰۶/۲۹

چند سوال:

- 1- آیا امکانات ویژوال استدیو در Linux به اندازه نسخه ویندوزی آن است؟
- 2- خبرهایی که مبنی بر مهاجرت به لینوکس میشه تا چه اندازه جدیه؟
- 3- آیا با محصولات دات نت در Linux با مشکلی مواجه نمیشویم؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۲۲ ۱۳۹۲/۰۶/۲۹

- مجبور نیستید 100 درصد در لینوکس برنامه نویسی کنید. در ویندوز با VS.NET کار کنید. نتیجه قرار است در لینوکس اجرا شود یا هاست شود.
- نمی دونم. من اصلاً مشتری دولتی ندارم. (ضمن اینکه به نظر خیلی ها [اصل خبر رو](#) درست نخوندند و بد تفسیر کردند)
- یک سری اما و اگر داره. اگر از API ویندوز استفاده نکرده باشید، اگر از WPF استفاده نکرده باشید برنامه شما به احتمال زیاد قابل اجرا و انتقال خواهد بود.

عنوان: نصب Mono Develop 4.x در Ubuntu

نویسنده: وحید نصیری

تاریخ: ۱۴۰۱/۰۶/۳۱

آدرس: www.dotnettips.info

برچسب‌ها: Mono, Linux

پیشنیازها

در قسمت قبل ، موفق به نصب Mono 3.0 در لینوکس شدیم. در ادامه قصد داریم یک IDE لینوکسی مخصوص کار با Mono را به نام Mono Develop بر روی Ubuntu نصب کنیم. اگر مونو را نصب کرده‌اید، نیاز است پیشنیازهای ذیل را بر روی سیستم خود نصب کنید:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install -y build-essential libc6-dev g++ gcc libglib2.0-dev pkg-config \
git-core apache2 apache2-threaded-dev bison gettext autoconf automake libtool \
libpango1.0-dev libatk1.0-dev libgtk2.0-dev libtiff5-dev libgif-dev libglade2-dev curl \
python-software-properties gawk libjpeg-dev libexif-dev flex checkinstall intltool git \
libcairo2-dev libgnomecanvas2-dev libgnome2-dev libgnomeui-dev libgnomeprint2.2-dev \
libgnomeprintui2.2-dev libgtkhtml3.14-dev libgtksourceview2.0-dev librsvg2-dev libvte-dev \
libnspr4-dev libnss3-dev libwebkit-dev apache2-threaded-dev libpng12-dev libfontconfig1-dev \
libfreetype6-dev zlib1g-dev libjpeg8-dev libjpeg-turbo8-dev libart-2.0-dev libgnomevfs2-dev \
libgnome-desktop-dev libnautilus-extension-dev libwnck-dev libvala-0.18-dev \
mono-addins-utils gtk-sharp2 gnome-sharp2
```

نصب این پیشنیازها ضروری بوده و در غیر اینصورت موفق به build کامل Mono Develop نخواهید شد. برای مثال پیغام خطای ذیل را در انتهای build دریافت می‌کنید؛ به این معنا که اسمبلی‌های ذیل کامپایل نشده‌اند:

```
* art-sharp.dll: no
* gnomevfs-sharp.dll: no
* gnome-sharp.dll: no
```

و یا عنوان می‌کند که gnome-sharp.dll برای کامپایل نیاز به یک سری کتابخانه کمکی دیگر نیز دارد:

gnome-sharp.dll requires libgnomecanvas, libgnome, and libgnomeui.

نصب متداول محیط برنامه نویسی Mono Develop

برای نصب یک IDE که بتوان تحت همان لینوکس نیز کار برنامه نویسی دات نت را انجام داد، می‌توان از [Mono develop](http://mono.develop) استفاده کرد. برای نصب آن فرمان ذیل را در خط فرمان لینوکس صادر نکنید !

```
sudo apt-get install monodevelop
```

این روش هر چند کار می‌کند، اما تا این تاریخ، نگارش 3 را نصب خواهد کرد؛ با توجه به اینکه آخرین نگارش موجود در سایت Mono Develop، نگارش 4 است. همچنین نصب آن نیز نگارش جاری Mono را به نگارش 2 آن تنظیم می‌کند که جالب نیست. اگر به اشتباه آن را نصب کرده‌اید، برای حذف مونو از دستور ذیل استفاده کنید:

```
sudo apt-get purge cli-common mono-runtime
```

همچنین الان کلیه مسیرهای سیستم به هم ریخته است. برای رفع آن مسیر نصب Mono-3.0 را باید به نحو ذیل مجدداً تنظیم کرد:

```
export PATH=/opt/mono-3.0/bin:$PATH
export PKG_CONFIG_PATH=/opt/mono-3.0/lib/pkgconfig:$PKG_CONFIG_PATH
```


نصب محیط برنامه نویسی Mono Develop از روی مخزن کد آن

دریافت و نصب وابستگی‌های Monodevelop جهت کامپایل سورس آن، شاید نصف روز شما را به خود اختصاص دهد؛ به علاوه حداقل مصرف حدود 500 مگابایت حجم اینترنت. راه ساده‌تری نیز برای دریافت آخرین نگارش سازگار با Ubuntu آن وجود دارد و آن هم استفاده از بسته‌های شخصی کامپایل شده است؛ که اصطلاحاً به آن‌ها PPA نیز گفته می‌شود. برای مثال: ([^](#) و [^](#)) چند نمونه بسته شخصی برای دریافت ساده آخرین نگارش Mono develop جهت نصب بر روی Ubuntu: ([^](#) و [^](#) و [^](#)) و به صورت خلاصه فرامین ذیل را در ترمینال لینوکس اجرا کنید تا از بسته شخصی [keks9n](#) استفاده کنیم:

```
sudo add-apt-repository ppa:keks9n/monodevelop-latest
sudo apt-get update
sudo apt-get install monodevelop-latest
```

این روش، از تمام روش‌های ذکر شده تا کنون، ساده‌تر است. از این لحاظ که mono 3.2.1 را نیز به صورت خودکار بر روی سیستم شما نصب می‌کند (این بسته شخصی، به صورت خودکار هر از چندگاهی آخرین نگارش مونو، وابستگی‌های آن و monodevelop جدید را بسته بندی و ارائه می‌دهد). بنابراین اگر مونو 3.2.1 یا جدیدتر را هنوز نصب نکرده‌اید، همین سه سطر فوق، کار نصب کلی آن را نیز انجام می‌دهد؛ علاوه بر نصب monodevelop در آخر کار به همراه تمام پیشنیازهای لازم مانند gtk-sharp و gnome-sharp. پس از نصب کامل، برای اجرای آن در همان خط فرمان، دستور monodevelop را صادر کنید.



نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۵۵ ۱۳۹۲/۰۶/۳۱

یک نکته تکمیلی در مورد نصب از روی PPA

اگر یک برنامه جدید را در monodevelop شروع کردید و به خطای کامپایل ذیل برخوردید:

```
Error: A dependency of a referenced assembly may be missing, or you may be referencing an assembly created with a newer CLR version. See the compilation output for more details. (Test1)
```

به این معنا است که mono تنظیم شده در مسیرهای لینوکس، به آخرین نگارش تنظیم نیست. برای رفع این مشکل دستورات ذیل را صادر و بعد monodevelop را اجرا کنید:

```
export PATH=/opt/mono-3/bin:$PATH
export PKG_CONFIG_PATH=/opt/mono-3/lib/pkgconfig:$PKG_CONFIG_PATH
mono -V
```

بسته PPA، نگارش 3.2.1 را در مسیر opt/mono-3 نصب می‌کند.

نویسنده: رضا خان میرزائی
تاریخ: ۱۶:۵۷ ۱۳۹۲/۰۶/۳۱

واقعا عالی بود . اگر لطف کنید و یک راهنمایی برای توسعه یک برنامه نمونه (مثلا برنامه تحت وب با mvc) و نکاتی که ممکن است پیش بیاید بنویسید خیلی ممنون می‌شوم .

عنوان: اجرای برنامه‌های ASP.NET توسط Mono در Ubuntu

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۰۷/۰۱

آدرس: www.dotnettips.info

برچسب‌ها: ASP.Net, MVC, Mono, Linux

در ادامه مباحث بررسی اجرای برنامه‌های دات نت بر روی لینوکس، قصد داریم برنامه‌های ASP.NET را به کمک Mono 3.0 و یک وب سرور لینوکسی، بر روی Ubuntu [اجرا کنیم](#).

پیشنیازها

دو پروژه خالی ASP.NET Web forms و ASP.NET MVC را در VS.NET تحت ویندوز ایجاد نمائید. آن‌ها را یکبار کامپایل کرده و اجرا کنید. سپس فایل‌های آن‌ها را به ubuntu منتقل کنید (پوشه‌های bin پروژه‌ها فراموش نشوند؛ خصوصا نگارش MVC که به همراه یک سری کتابخانه جانبی است).

برای انتقال فایل‌ها به لینوکس، اگر از VMWare workstation برای اجرا و آزمایش Ubuntu استفاده می‌کنید، کپی و paste مستقیم فایل‌ها از ویندوز به درون ماشین مجازی لینوکس پشتیبانی می‌شود.

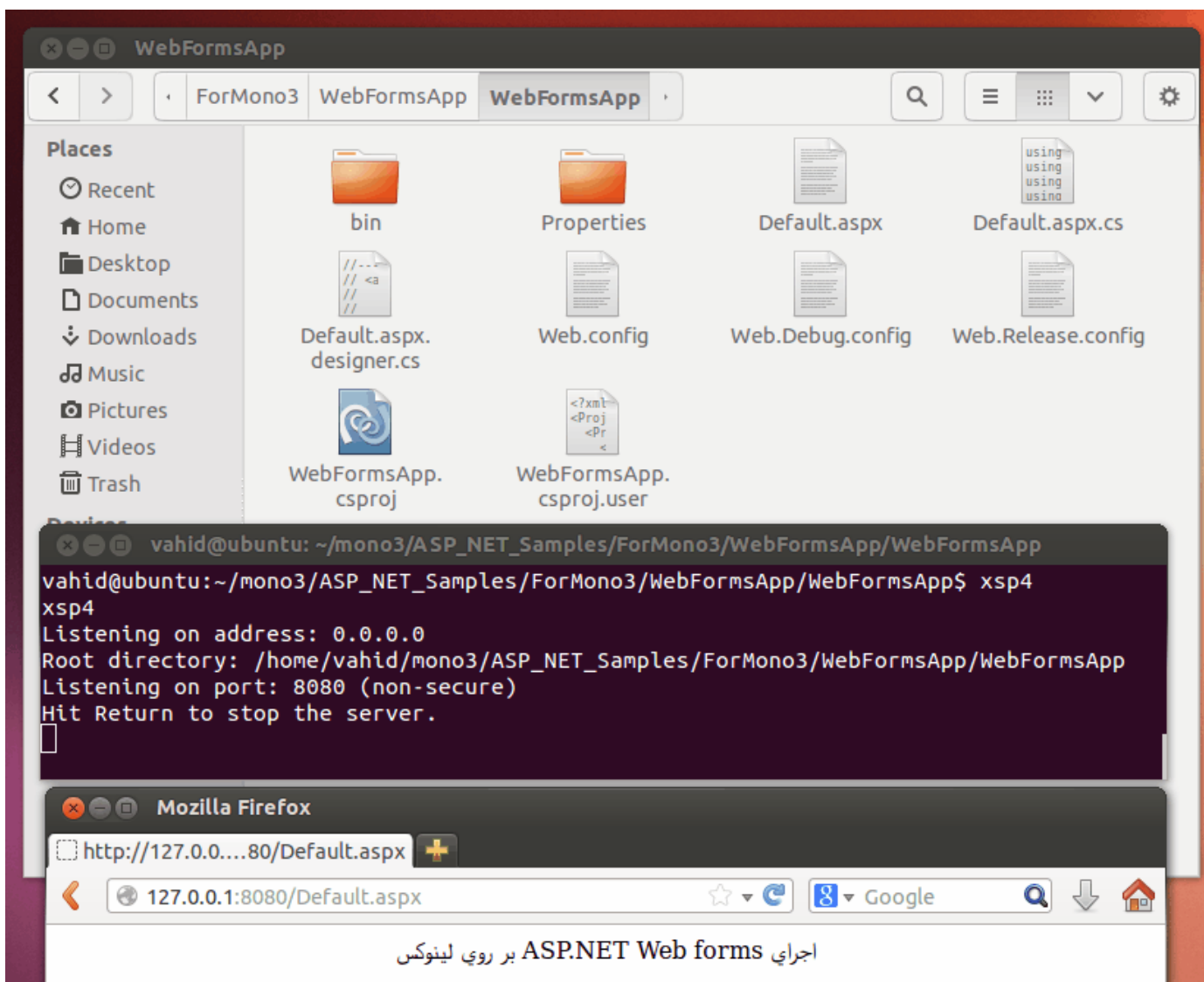
نصب وب سرور آزمایشی مونو یا XSP

اگر نیاز به یک وب سرور آزمایشی، چیزی شبیه به وب سرور توکار VS.NET داشتید، پروژه [XSP](#) جهت این نوع آزمایشات ایجاد شده است.

پس از نصب آن (که به همراه همان بسته PPA [قسمت قبل](#)، هم اکنون بر روی سیستم شما نصب است)، در ترمینال لینوکس، با استفاده از دستور cd به ریشه وب سایت خود وارد شوید، سپس دستور xsp4 را اجرا کنید تا وب سرور xsp4 مشغول هاست سایت شما شود (برای اجرا در مسیر opt/bin/xsp4/ نصب شده است).

اجرای برنامه 4 ASP.NET Web forms توسط XSP

بدون هیچ مشکل خاصی در همان ابتدای کار اجرا شد (البته باید دقت داشت که لینوکس به کوچکی و بزرگی حروف حساس است. یعنی حتما باید Default.aspx وارد شود و نه default.aspx):



اجرای برنامه ASP.NET MVC 4 توسط XSP

اجرا نشد! پیام می‌دهد که

"Missing method System.Web.Security.FormsAuthentication::get_IsEnabled() in assembly System.Web.dll

و یا

Compiler Error Message: CS1703: An assembly with the same identity 'mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089' has already been imported. Consider removing one of the references

علت اینجا است که XSP4 همراه با نسخه PPA، قدیمی است. بنابراین باید نسخه اصلی را از مخزن کد آن دریافت و کامپایل کنیم. پیشنهادها اینکار مانند Git و Mono، در قسمت قبل دریافت شدند. سپس فرامین ذیل را در خط فرمان لینوکس اجرا کنید:

```
git clone git://github.com/mono/xsp.git
cd xsp
./autogen.sh --prefix=/opt
make
sudo make install
```

پس از کامپایل، اگر این نگارش جدید را اجرا کنید، به خطای ذیل خواهید رسید:

```
System.IO.FileNotFoundException: Could not load file or assembly XSP, Version=3.0.0.0
```

برای رفع این مشکل باید اینبار وب سرور جدید را با فرمان `sudo` یا دسترسی مدیریتی اجرا کنید تا مشکل برطرف شود. البته من [سورس دریافت شده را](#) در خود `monodevelop` کامپایل کردم (فایل `sln` آن را در `monodevelop` باز کرده و پروژه را `build` کنید). در این حالت دو فایل `Mono.WebServer.dll` و `Mono.WebServer.XSP.exe` در پوشه `xsp/src/Mono.WebServer.XSP/bin/Debug` ظاهر می‌شوند.

یکی دیگر از دلایل ظاهر شدن خطای فوق، نیاز به نصب این دو فایل در GAC است که به نحو زیر قابل انجام می‌باشد:

```
cd xsp/src/Mono.WebServer.XSP/bin/Debug
sudo gacutil -i Mono.WebServer.XSP.exe
sudo gacutil -i Mono.WebServer.dll
```

بعد این دو فایل `dll` و `exe` را در پوشه برنامه MVC خود کپی کنید و سپس دستور ذیل را اجرا نمایید:

```
cd myMvcAppPath
sudo mono Mono.WebServer.XSP.exe
```

اینبار وب سرور جدید، روی پورت 9000 شروع به کار می‌کند. اکنون اگر در فایرفاکس آدرس `http://localhost:9000` را باز کنید، برنامه اجرا شده اما به خطای ذیل خواهید رسید:

```
CS0234: The type or namespace name 'Helpers' does not exist in the namespace 'System.Web'.
Are you missing an assembly reference?
```

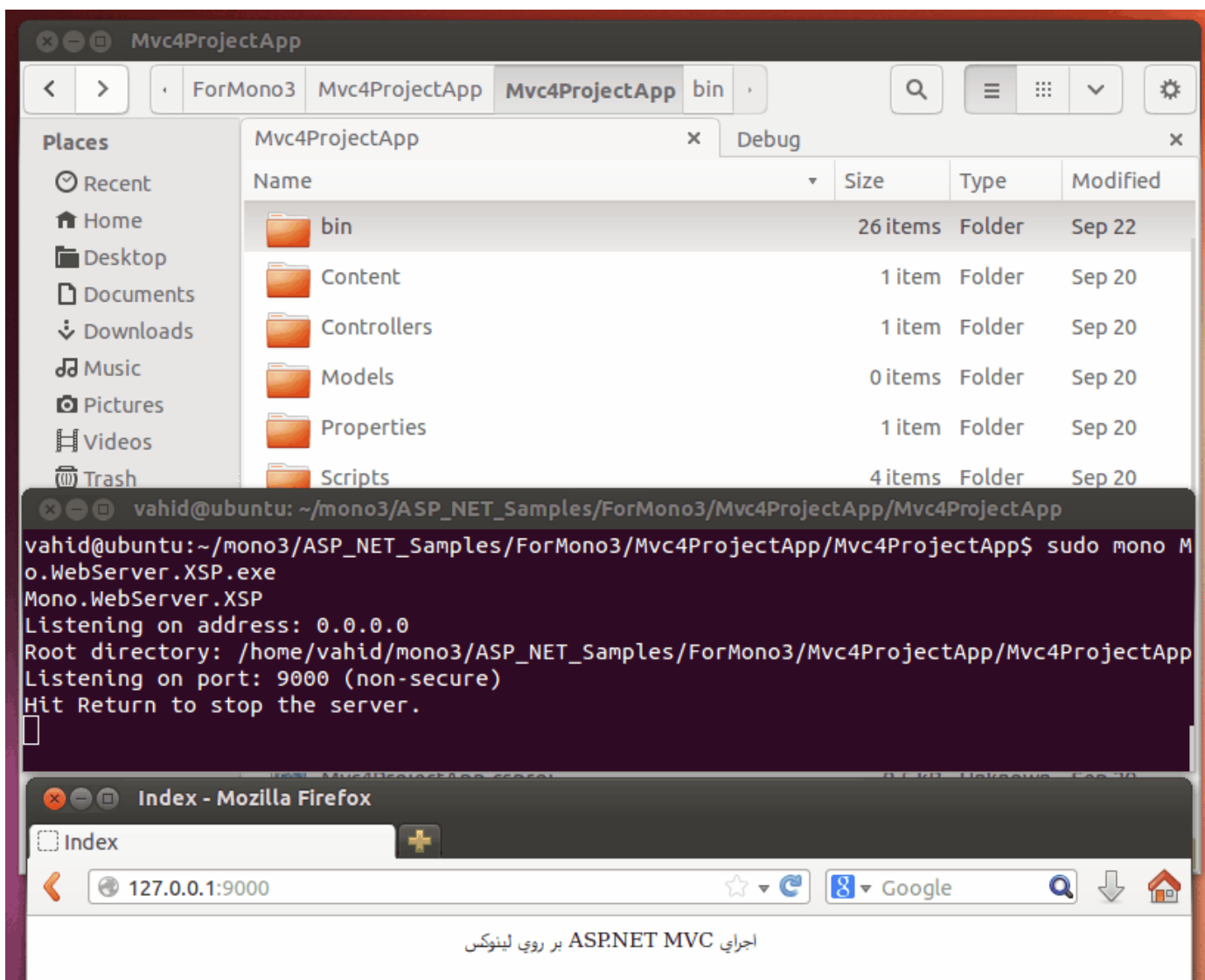
برای رفع این مشکل باید اندکی فایل `web.config` برنامه را ویرایش کرد:

```
<system.web>
  <compilation debug="true" targetFramework="4.0">
    <assemblies>
      <add assembly="System.Web.Helpers, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
      <add assembly="System.Web.Mvc, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
    </assemblies>
  </compilation>
```

سعی بعدی ... اجرا نشد! با هر بار `refresh` صفحه یک خطای جدید نمایش می‌داد که ... `Type` خاصی را نمی‌تواند بارگذاری کند (به همراه نام اسمبلی مربوطه). برای رفع این مشکل `dll`های ذیل را از پوشه `bin` پروژه MVC خود که از ویندوز به لینوکس کپی کرده‌اید، حذف کنید:

```
Microsoft.Web.Infrastructure.dll
System.Net.Http.dll
System.Net.Http.Formatting.dll
System.Web.Http.dll
System.Web.Http.WebHost.dll
```

این فایل‌ها توسط تیم Mono به صورت مستقل پیاده سازی شده‌اند و نیازی نیست تا از ویندوز به لینوکس کپی شوند. بعد از حذف این فایل‌های اضافی، برنامه ASP.NET MVC نیز اجرا شد:



چند نکته تکمیلی

- نحوه تشخیص موجود بودن یک DLL خاص، در نگارش جاری Mono نصب شده:

```
$ gacutil -l Microsoft.Web.Infrastructure
The following assemblies are installed into the GAC:
Microsoft.Web.Infrastructure, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
Number of items = 1
```

- اگر می‌خواهید مطمئن شوید که تمام اسمبلی‌های موجود در GAC درست نصب شده‌اند یا خیر، فرمان ذیل را اجرا کنید:

```
cd /opt/lib/mono/gac # assuming this is your main gac
sudo find . */*/*.dll -exec gacutil -i '{}' \;
```

- در نسخه لینوکسی System.Web ممکن است یک سری از فضاها نام هنوز موجود نباشند. لیست آن‌ها را در این آدرس می‌توانید مشاهده کنید:

<http://go-mono.com/status/status.aspx?reference=4.0&profile=4.0&assembly=System.Web>

نظرات خوانندگان

نویسنده: علی صداقت
تاریخ: ۱۷:۲۲ ۱۳۹۲/۰۷/۰۱

جناب نصیری، دیتابیس رو چه کنیم؟ واقعا جانشین منبع بازی، همسان با sql server در این طرف داریم؟ MySQL کفایت می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۵۶ ۱۳۹۲/۰۷/۰۱

- MySQL هم خوبه (البته مجوز استفاده‌اش رو هم باید [مدنظر داشته باشید](#)). هرچند از زمانیکه به تملک اوراکل درآمده، خیلی از شرکت‌ها دارند به سمت MariaDB کوچ می‌کنند: [^](#) و [^](#)
+ Oracle [نسخه لینوکسی](#) هم دارد.
- و برای کارهای سبک SQLite [نسخه لینوکسی](#) دارد.

نویسنده: محمد جواد تواضعی
تاریخ: ۱:۱۲ ۱۳۹۲/۰۷/۰۲

شما می‌توانید از postgresql استفاده کنید . کاملاً متن باز بوده و چیزی از oracle و Ms SqlServer کم ندارد .

در قسمت قبل نحوه‌ی اجرای برنامه‌های ASP.NET را به کمک وب سرور آزمایشی Mono بررسی کردیم. بدیهی است این وب سرور برای هاست کردن سایت‌ها و خدمات دهی عمومی، مناسب نیست و صرفاً جهت کارهای برنامه نویسی طراحی شده است. در ادامه، ابتدا وب سرور معروف آپاچی را نصب کرده و سپس سعی خواهیم کرد تا همان دو برنامه ASP.NET و ASP.NET Web forms MVC قسمت قبل را بر روی آن اجرا کنیم (دو برنامه خالی ASP.NET کامپایل شده که از ویندوز به لینوکس کپی شده بودند).

نصب وب سرور آپاچی

به اینترنت متصل شده، ترمینال Ubuntu را گشوده (با میانبر `ctrl+alt+t`) و سپس فرمان‌های ذیل را صادر کنید:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install apache2
```

به این ترتیب Apache web server بر روی سیستم نصب خواهد شد.

نصب ماژول‌های ASP.NET مخصوص آپاچی

سپس نیاز است ASP.NET runtime for Apache را نصب کنیم:

```
sudo apt-get install libapache2-mod-mono mono-apache-server4
```

mod_mono درون پروسه آپاچی اجرا شده و تمام درخواست‌های رسیده به یک وب سایت ASP.NET را به mod-mono-server که کار هاست سایت را انجام می‌دهد، انتقال خواهد داد.

اگر علاقمند به مشاهده تنظیمات آن بودید باید به مسیر `etc/apache2/mods-enabled` مراجعه کرده و فایل `mod_mono.conf` را بررسی کنید (اختیاری). برای مثال، در آن حالت اجرا، بر روی 4 ASP.NET تنظیم شده‌است.

تنظیمات آپاچی برای کار با ASP.NET

نیاز است فایل تنظیمات پیش فرض وب سرور آپاچی را جهت معرفی ASP.NET به آن، اندکی ویرایش کنیم:

```
sudo gedit /etc/apache2/sites-available/default
```

در ادیتور باز شده، ذیل قسمت `ServerAdmin`، موارد زیر را اضافه کنید:

```
MonoAutoApplication disabled
AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
MonoApplications "/:/var/www"
MonoServerPath default "/usr/bin/mod-mono-server4"
```

در ادامه به المان `Directory /var/www` سطرهای زیر را اضافه کنید:

```
SetHandler mono
DirectoryIndex index.aspx index.html default.aspx Default.aspx
```

المان جدید ذیل را در انتهای فایل اضافه کنید تا یک سری از پوشه‌های خاص ASP.NET همانند محیط‌های ویندوزی در آپاچی نیز محافظت شوند:

```
<DirectoryMatch
"/([bB]in|[Aa]pp_[Cc]ode|[Aa]pp_[Dd]ata|[Aa]pp_[Gg]lobal[Rr]esources|[Aa]pp_[Ll]ocal[Rr]esources)/">
```

```
Order deny,allow
Deny from all
</DirectoryMatch>
```

فایل را ذخیره کرده و ادیتور را ببندید. اکنون وب سرور آپاچی را اجرا و ری استارت کنید:

```
sudo service apache2 restart
```

همچنین سطح دسترسی مسیر `var/www` که حاوی برنامه‌های ما خواهد بود نیز باید تغییر کند:

```
# r,w,exec for user + group
sudo chgrp -R www-data /var/www
sudo chown -R www-data /var/www
sudo chown -R www-data:www-data /var/www
sudo chmod -R 755 /var/www
```

اکنون اگر آدرس `http://127.0.0.1` را در مرورگر وارد کنید، باید پیام «It Works» را مشاهده کنید و اگر خواستید محتوای آن را ویرایش کنید، دستور ذیل را صادر نمائید:

```
gedit /var/www/index.html
```

اجرای مثال‌ها

با تنظیمات فوق، برنامه‌های کپی شده در مسیر `var/www` به کمک مونو و آپاچی اجرا خواهند شد. دسترسی‌های فعلی کاربر وارد شده به لینوکس اجازه کپی فایل‌ها را به مسیر `var/www` نمی‌دهد. همچنین می‌خواهیم این کارها را توسط File browser آن انجام دهیم و نه خط فرمان. برای این منظور دستور ذیل را اجرا کنید تا File browser آن با دسترسی مدیریتی اجرا شود:

```
sudo nautilus
```

الان می‌توان بدون مشکل در همان File browser گرافیکی آن، پوشه مثال وب فرم‌های خود را به درون پوشه `var/www` کپی کرد. پس از کپی دو برنامه Web forms و MVC ابتدای بحث، نیاز است مجدداً فایل تنظیمات آپاچی را ویرایش کنیم:

```
sudo gedit /etc/apache2/sites-available/default
```

اکنون تنظیم برنامه وب فرم را به نحو ذیل به این فایل اضافه کنید:

```
AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/"
<Location /webforms_test>
SetHandler mono
</Location>
```

در تنظیم `/webforms_test:/var/www/WebFormsApp` قسمت `webforms_test` نامی است که در مرورگر وارد خواهد شد و قسمت بعد از : مسیر فایل‌های برنامه و ریشه سایت است. و تنظیم برنامه MVC به صورت زیر می‌باشد:

```
AddMonoApplications default "/mvc_test:/var/www/Mvc4ProjectApp/"
<Location /mvc_test>
SetHandler mono
</Location>
```

بعد از هر بار تغییر فایل تنظیمات آپاچی باید یکبار وب سرور را توسط دستور `sudo service apache2 restart` ری استارت کنیم.

اکنون برنامه وب فرم‌ها در مسیر `http://127.0.0.1/webforms_test` و برنامه MVC در مسیر `http://127.0.0.1/mvc_test` قابل دسترسی است.

خلاصه بحث

پس از نصب وب سرور آپاچی و ماژول مونوی مخصوص آن، فایل `etc/apache2/sites-available/default` را به نحو ذیل ویرایش کنید و مسیرهای برنامه‌های خود را در آن تعریف نمایید:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
    MonoAutoApplication disabled
    AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
    MonoApplications "/:/var/www"
    MonoServerPath default "/usr/bin/mod-mono-server4"

DocumentRoot "/var/www"
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
<Directory /var/www/>
    SetHandler mono
    DirectoryIndex index.aspx index.html default.aspx Default.aspx
Options Indexes Includes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>

    <DirectoryMatch
"/([bB]in|[Aa]pp_[Cc]ode|[Aa]pp_[Dd]ata|[Aa]pp_[Gg]lobal[Rr]esources|[Aa]pp_[Ll]ocal[Rr]esources)/">
        Order deny,allow
        Deny from all
    </DirectoryMatch>

    AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/"
    <Location /webforms_test>
        SetHandler mono
    </Location>

    AddMonoApplications default "/mvc_test:/var/www/Mvc4ProjectApp/"
    <Location /mvc_test>
        SetHandler mono
    </Location>

ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۹:۴۷ ۱۳۹۲/۰۷/۰۳

در مورد هاست کردن سرویس‌های WCF هم آگه ممکنه توضیح بدین.

نویسنده: وحید نصیری
تاریخ: ۱۴:۳ ۱۳۹۲/۰۷/۰۴

الف) تنظیمات سرور

برای اضافه کردن تنظیمات WCF، ابتدا فایل mod_mono.conf را باز کنید:

```
sudo gedit /etc/apache2/mods-available/mod_mono.conf
```

بعد در سطر اول آن پسوند svc را هم اضافه کنید:

```
AddType application/x-asp-net .svc .aspx .ashx .asmx .ascx .asax .config .ascx
```

و همچنین اینکار را برای فایل default انجام دهید:

```
sudo gedit /etc/apache2/sites-available/default
```

و پسوند svc را در اینجا نیز اضافه نمائید:

```
AddHandler mono .svc .aspx .ascx .asax .ashx .config .cs .asmx .axd
```

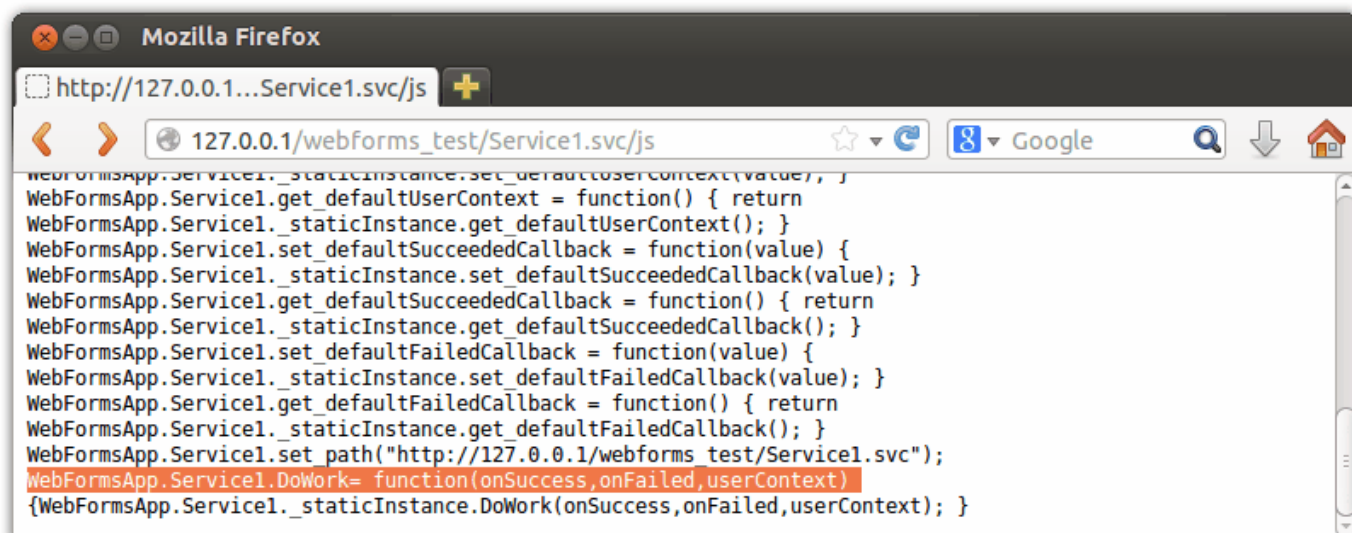
بعد از آن یکبار سرور را با دستور `sudo service apache2 restart` ری استارت کنید.

ب) تنظیمات وب کانفیگ

در اینجا serviceMetadata را طوری تنظیم کنید تا با HTTP GET قابل دریافت باشد

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior>
        <serviceMetadata httpGetEnabled="true" httpGetUrl="wsdl"/>
      </behavior>
    </serviceBehaviors>
  </behaviors>
</system.serviceModel>
```

بعد از آن سرویس شما در آدرس `http://127.0.0.1/webforms_test/Service1.svc/js` یا در آدرس `http://127.0.0.1/webforms_test/Service1.svc/jsdebug` برای کارهای Ajax ایی قابل استفاده خواهد بود (فرض بر این بود که یک Ajax Enabled WCF Service را در VS.NET ایجاد و به لینوکس منتقل کردید).



```
WebFormsApp.Service1._staticInstance.set_defaultUserContext(value); }
WebFormsApp.Service1.get_defaultUserContext = function() { return
WebFormsApp.Service1._staticInstance.get_defaultUserContext(); }
WebFormsApp.Service1.set_defaultSucceededCallback = function(value) {
WebFormsApp.Service1._staticInstance.set_defaultSucceededCallback(value); }
WebFormsApp.Service1.get_defaultSucceededCallback = function() { return
WebFormsApp.Service1._staticInstance.get_defaultSucceededCallback(); }
WebFormsApp.Service1.set_defaultFailedCallback = function(value) {
WebFormsApp.Service1._staticInstance.set_defaultFailedCallback(value); }
WebFormsApp.Service1.get_defaultFailedCallback = function() { return
WebFormsApp.Service1._staticInstance.get_defaultFailedCallback(); }
WebFormsApp.Service1.set_path("http://127.0.0.1/webforms_test/Service1.svc");
WebFormsApp.Service1.DoWork= function(onSuccess,onFailed,userContext)
{WebFormsApp.Service1._staticInstance.DoWork(onSuccess,onFailed,userContext); }
```

نویسنده: ایمان اسلامی
تاریخ: ۸:۲۷ ۱۳۹۲/۱۲/۳۰

میخواستم بدونم در شرایط مساوی نصب یک برنامه asp.net روی iis از نظر performance بهتره یا روی apache؟

یکی از روش‌های ارسال و رمزگذاری اطلاعات، استفاده از کلیدهای امنیتی مورد استفاده‌ی در سیستم یونیکس یا [GnuPG](#) است. استفاده از نرم افزار Gnu Privacy Guard یا گارد حفاظتی گنو، به ما این اجازه را می‌دهد که بتوانیم اطلاعاتمان را در بسترهای ارتباطی، با خیالی راحت‌تر ارسال کنیم و تا حد زیادی مطمئن باشیم که تنها فرد هدف توانایی دسترسی به اطلاعات را خواهد داشت. گارد امنیتی گنو زیر مجموعه‌ای از پروژه‌ی گنو است که دولت آلمان پایه ریز اصلی آن بوده است. این نرم افزار از یک روش رمزگذاری ترکیبی استفاده می‌کند که الگوریتم‌های [کلیدهای برابر\(مقارن\)](#) و [کلیدهای عمومی \(نامقارن\)](#) جهت تبادل آسان کلید را شامل می‌شود. در حال حاضر که نسخه‌ی دو این برنامه ارائه شده است، برای رمزگذاری‌ها از کتابخانه‌ای به اسم [libgcrypt](#) استفاده می‌کند. یکی از مشکلات فعلی این پروژه، عدم وجود api مناسبی جهت دسترسی راحت‌تر است و برای حل این مشکل، GPGME که مخفف GnuPG Made Easy ایجاد شد. بسیاری از برنامه‌ها و پلاگین‌های ارسال اطلاعات، امروزه همچون ارسال ایمیل، از این کلیدها بهره می‌برند.

پروژه‌های مرتبط با این قضیه اسم‌های مشابهی دارند که گاهی بعضی افراد، هر کدام از اسم‌ها را که دوست دارند، به همه اطلاق می‌کنند؛ ولی تفاوت‌هایی در این بین وجود دارد:

OpenPGP : یک برنامه نیست و یک قانون و استاندارد برای تهیه‌ی آن است؛ که رعایت اصول آن الزامی است و برنامه‌ی بالا، یک پیاده سازی از این استاندارد است.

PGP : یک برنامه، برای رمزگذاری اطلاعات است که مخفف [Pretty Good Privacy](#) است.

و **GnuPG** یا **GPG** که در بالا به آن اشاره شد.

برای ساخت کلید، ما از دستور یا برنامه‌ی GPG که که عمدتاً در همه‌ی لینوکس‌ها مثل دبیان و مشتقات آن نصب است، استفاده می‌کنیم و اگر نصب نیست از طریق توزیع آن اقدام نمایید.

در صورتیکه از ویندوز استفاده می‌کنید، نیاز است ابتدا خط فرمان یونیکس را روی آن نصب کنید. برنامه‌ی [Cygwin](#) این امکان را به شما می‌دهد تا خط فرمان یونیکس و دستورات پیش فرض آن را داشته باشید. این برنامه در دو حالت ۳۲ بیتی و ۶۴ بیتی ایجاد شده است. از آنجا که گفتیم این برنامه شامل دستورات پیش فرض آن است، برای همین GPG باید به صورت یک بسته‌ی جداگانه نصب شود که در [سایت آن](#) می‌توانید بسته‌های مختلف آن را برای پلتفرم‌های مختلف را مشاهده کنید.

ساخت کلید

برای ساخت کلید دستور زیر را صادر کنید:

```
gpg --gen-key
```

اگر از نسخه‌های جدیدتر GPG استفاده می‌کنید، گزینه‌هایی به شکل زیر ایجاد می‌شوند؛ ولی اگر خیر، ممکن است تعداد و شمارهی گزینه‌ها متفاوت باشند که در این مورد دقت کنید. من در اینجا همان حالت پیش فرض، یعنی ۱ را انتخاب می‌کنم. این گزینه نحوه‌ی امضاء و یا رمزگذاری شما با استفاده از الگوریتم‌های [RSA](#) و [DSA](#) را مشخص می‌کند.

Please select what kind of key you want:

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)

در کل در هر حالتی، استفاده‌ی از RSA پیشنهاد می‌شود. بعد از آن، از شما اندازه‌ی کلید را می‌پرسد که همان مقدار پیش فرض خودش را وارد می‌کنیم:

What keysize do you want? (2048)

البته بسیاری ۲۰۴۸ بیت را نیز کافی می‌دانند.
بعد از آن مدت زمان اعتبار این کلید را از شما جویا می‌شود:

Key is valid for? (0)

هنگام این پرسش نحوه‌ی ورود زمان را به شما خواهد گفت که می‌تواند به شکل‌های زیر باشد:

دو هفته
2w
دو سال
2y

پس از آن هم یک تاییدیه از شما می‌گیرد و تاریخ انقضاء را به طور کامل برای شما می‌نویسد و سپس نیاز است که اطلاعاتی از قبیل نام و ایمیل و توضیح را وارد کنید:

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:
"Heinrich Heine (Der Dichter) <heinrich@duesseldorf.de>"

Real name: ali yeganeh.m
Email address: yeganehaym@gmail.com
Comment: androidbreadcrumb
You selected this USER-ID:
"ali yeganeh.m (androidbreadcrumb) <yeganehaym@gmail.com>"

بعد از آن از شما می‌خواهد که کل عملیات را تایید و یا کنسل کنید؛ یا اگر اطلاعات بالا را اشتباه وارد کرده‌اید، اصلاح کنید. با زدن کلید 0 عملیات را تایید کنید. در این حین از شما یک کلید برای رمزگذاری می‌پرسد که باید آن را دو بار بدهید و کارتان در اینجا به پایان می‌رسد و کلید ایجاد می‌شود.
اگر مشکلی در ساخت کلید نباشد با ارسال دستور زیر باید آن را در لیست کلیدها ببینید:

```
ali@alipc:~$ gpg --list-keys
/home/ali/.gnupg/pubring.gpg
-----
pub 2048R/8708016A 2015-10-23 [expires: 2065-10-10]
uid ali yeganeh.m (androidbreadcrumb) <yeganehaym@gmail.com>
sub 2048R/533B7E96 2015-10-23 [expires: 2065-10-10]
```

در اینجا کلید عمومی در خط pub بعد از / قرار دارد؛ یعنی عبارت ۸۷۰۸۰۱۶A کلید عمومی ماست که بر روی هر سیستم و هر کلیدی متفاوت است.

تبدیل کد متنی به کد دودویی

یکی از روش‌های ارسال کدهای دودویی تبدیل آنان به یک قالب متنی ASCII است که به آن قالب [ASCII Armor](#) هم می‌گویند. سایت‌های زیادی وجود دارند که این عبارت متنی را از شما می‌خواهند. چرا که مثلاً این امکان وجود دارد که کلیدی که کاربر به سمت آنان می‌فرستد، آسیب دیده باشد یا اینکه KeyServerها در دسترس نباشند. در مورد این سرورها در ادامه صحبت خواهیم کرد. مثلاً یکی از سایت‌هایی که به این عبارت‌ها نیاز دارد [Bintray](#) است.

برای دریافت این کلید متنی باید دستور زیر را صادر کنید:

```
gpg --output mykey.asc --export -a $GPGKEY
```

که برای مثال ما می‌شود:


```
gpg --output mykey.asc --export -a 8708016A
```

و اگر کلید را با یک ویرایشگر متنی باز کنید، محتوایی شبیه محتوای زیر را خواهید دید:

```
ali@alipc:~$ cat mykey.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1

mQENBFYqAJABCAdcw5xPonh5Vj7nDk1CxDskq/Vs008X0a/i20L0zatB4oK5x+0x
jxORxMnIAR83PCK5/Wk0Ba64jnu3eiP3jKEwAyKGGZ/Z1bezC9TIP8y+PnsiDhT
aFarluUJx+RT5q7s27aKjqoc3fR/xuwlWopZt9uYzE/DQAPDsHdUoUg+fh4Hevm+
a8/3ncR7q6nM8gc9wk621Urb1HaRrILdmeh7ZpJc18ZUbc+N0bw357fGsJnpfHX0
rdCr7C1vNUq6I+IeGMQG/6040LeeaqhaRxPrUhbFjLA155gkSqzecx17wQaYc71M
Zdlv+6Pt1B8nPAA3WxQ0ypjU8A5bvmAQRD5LABEBAAG0GfSaSB5ZWdhbmVoLm0g
KGFuZHZJvaWRicmVhZGNYdW1kSA8eWVnYw5laGF5bUBnbWfPpbC5jb20+iQE+BBMB
AgAoBQJWkgCQAhsDBQ1d/A8ABgsJCAcDAgYVCAIJCgsEFgIDAQIeAQIXgAAKCRDS
Lhq8hwgBanaHB/4reGxUjR6dB08ykfwQOx+raYHGqJ1lgawisE4qUHTkGaspyQaNy
yxh0vwKkGvg6nNy2VN1XFbc7j1H1rYqPPuPdG2B+1LvEghb30ESDbHUvk8NrJgDJ
C0257gxqWvUQTWvMC3FkSLdw3tyQ8dF7FxmSU79XcxVqGeseaDzMQrEasP0yJHsm
NJf8pvuD6qiWu3KSSoQmI/17Sj8s7eGJMh6o5YRFghc1Bt9tCD+52bvt579Ju4vZ
tmQvXR4fNQo9sAeMqAJhIpF7IYcuyCEy+CQ847UkzE4f/OCCPxvV3samV/nnBJJ9
Ouu+681k6Fpx4A0a3nEwqoAmMwXrbSSUFW97uQENBFYqAJABCAC4CzrUOKskE4hK
GVCja0JKxhbuUdOrep6n3vof0fscs5Dy7h2oVh2vb12WH9X6pijJVPiUpGR4Mpu0
102Bu9Rwt38AQ6mRmL/hfzjEXSvKkdX7osk+1CVnnUaSdM9Ek2hWUH8JcN28z/WT
X9Bw8MCDZF7j1HvX/5ojghzMZYm4e1WJLBr1gON6xXAI6HR7D1nRkaVr8L9SYGm
FyAXZ0LzWYwG1Z1AntYxf6v/Mn3p1/1E3aBA+LkQqBzHg2nBm4jCaFWfCdINBf
CHkY9r/Evo9hUPD+CtBNFwsUm1D4maz0FFtIQ701QhVmupnub+rKo0bC0AFj3abK
MCw9uo8TABEBAAGJASUEGAECFAA8FAlYqAJACGwwFCV38DwAACgkQ0i4avIcIAWrz
rAf+K1IIMtBq3WlabfZQrgzFHQ62ugVJO/yI1ITkm4l08XHDf+ShqDg4urNumDEe
oQD35MvB2BhER1jL6VR3qjLkZyZYJ+EQiSxEDWxooav3KvpWjhcqjQy79GFs8waH
E7ssGmWwaugVS/PJAmGQ+s8YWDNa6aCC1mp2dJRiwbTyFdeWNBLa2V32xzWCYxhI
YtEp+K15XuCDTRatOPWFGSPe/paytmpGZc0XzU/W9sBpabhxVmcL4H6L07uCeF
IOOn/S5QXo3P9X/3ckmJ9GUb7rjdq1ivYgX53xI75jlePsmN/2f+3fNffUaZgFTTd
Uls+XCun70VYSBBfjgRfQbTvoA==
=6j7i
-----END PGP PUBLIC KEY BLOCK-----
```

در صورتی که قصد دارید متن کلید خصوصی را به دست بیاورید، لازم است بعد از export- عبارت secret-key- را نیز اضافه کنی
د؛ یعنی:

```
gpg --output mykey.asc --export-secret-key -a 8708016A
```

آپلود کلید به سرورهای کلید (Key Servers)

یکی از روش‌های به اشتراک گذاری کلید برای کاربران این است که از [سرورهای کلید](#) استفاده کنیم. یکبار آپلود روی یکی از این سرورها باعث می‌شود که به بقیه‌ی سرورها هم اضافه شود. یکی از این سرورهای کلید که خودم از آن استفاده می‌کنم، سرور ابونتو است و با استفاده از دستور زیر، همان کلید بالا را برای آن سرور ارسال می‌کنم:

```
gpg --send-keys --keyserver keyserver.ubuntu.com $GPGKEY
==>
gpg --send-keys --keyserver keyserver.ubuntu.com 8708016A
```

سپس از طریق کلید متنی، کلید آپلود شده را تایید می‌کنیم. به [این آدرس](#) رفته و محتوای کلید متنی خود را به طور کامل به همراه تگ‌های شروع و پایان کپی کنید و حتی می‌توانید کلید خود را از طریق کادر جست و جو پیدا کنید.

رمزگذاری

ابتدا در محیط یونیکس، یک فایل متنی ساده با متن hello ubuntu را ایجاد می‌کنم. در ادامه قصد دارم این فایل را رمزنگاری کنم:

```
ali@alipc:~$ cat >ali.txt
hello ubuntu
```

سپس همین فایل را رمزنگاری می‌کنم:

```
ali@alipc:~$ gpg --output myali.gpg --encrypt --recipient yeganehaym@gmail.com ali.txt
```

در این دستور ابتدا گفتیم که نام فایل خروجی ما myali.gpg است و می‌خواهیم آن را رمزگذاری کنیم که توسط کلیدی با ایمیل yeganehaym@gmail.com می‌باشد فایل ali.txt را رمزگذاری می‌کنیم.

رمزگشایی

برای رمزگشایی می‌توانید از طریق دستور زیر اقدام کنید:

```
gpg --output output.txt --decrypt myali.gpg
```

```
You need a passphrase to unlock the secret key for
user: "ali yeganeh.m (androidbreadcrumb) <yeganehaym@gmail.com>"
2048-bit RSA key, ID 533B7E96, created 2015-10-23 (main key ID 8708016A)
```

در اینجا دستور دادیم محتوای فایل رمزشده‌ی myali.gpg را رمزگشایی کن و محتوای آن را داخل فایل با نام output.txt قرار بده. بعد از اجرای این دستور از شما عبارت رمزی را که در مرحله‌ی ساخت کلید دوبار از شما پرسید، درخواست می‌کند. در بعضی سیستم‌ها در همان ترمینال می‌پرسد، ولی بعضی سیستم‌ها مثل ابونتو که من از آن استفاده می‌کنم، به صورت گرافیکی یک کادر باز کرده و از شما خواهش می‌کند عبارت رمز را وارد کنید. عبارت رمز را وارد کنید و حالا فایل output.txt را باز کنید:

```
ali@alipc:~$ cat output.txt
hello ubuntu
```