## Instalando o novo Opera no Linux

O navegador Opera é um dos navegadores mais antigos ainda em funcionamento - http://upload.wikimedia.org/wikipedia/commons/7/74/Timeline_of_web_browsers.svg

Mas o que tem de bom nesse navegador? Saiba que muitas funções que existem hoje em dia no seu navegador de internet provavelmente foi inventada pelo pessoal do Opera! Procure na internet e descobrirá quem inventou as abas, zoom na página, salvar sessão, palavra chave para busca, pesquisa integrada, bloqueador de pop-up...

O Opera até a versão 12 era baseado em tecnologia própria. Com a aparição do projeto open-source Chromium (base do navegador Google Chrome), o Opera passou a utilizar esse mesmo projeto como base para seu navegador, customizando à sua maneira. Ganhou mais velocidade para continuar entre os navegadores mais rápidos (teste seu navegador http://peacekeeper.futuremark.com )

Bom, agora vamos ao que interessa... Como instalar no Linux? Visitando http://www.opera.com , as versões de Windows e OSX são triviais. Para Linux baseado em Debian (Ubuntu, Mint, etc...) existe um pacote .deb na página oficial que também torna fácil o trabalho.

Mas para aqueles que utilizam outras versões que não utilizam pacotes .deb - OpenSUSE, Fedora, CentOS, Arch, Slackware, etc... - é necessário fazer a instalação manual ou utilizar um script para fazer isso para você. No site http://gist.github.com/ruario/99522c94838d0680633c temos toda a explicação do processo manual ~~e também temos um script que faz todo o trabalho "sujo": Baixa a versão mais atual, descompacta e joga na pasta /usr/local.~~

**NOVO!**

O script acima que baixa e instala o Opera em /usr/local/ tem apresentado problemas no OpenSUSE 13.2. O problema é que as abas que são fechadas não "devolvem" a memória que estavam utilizando. Após abrir e fechar várias abas, o seu sistema pode ficar com pouca memória disponível e pode ser necessário ou matar os subprocessos manualmente ou fechar e reabrir o Opera.

Contudo, se você mesmo compilar a sua versão do Opera para OpenSUSE, Fedora e outros sistemas que utilizam o formato .RPM, esse problema não acontece.

Para fazer isso é simples. Primeiro instale o pacote rpm-build no seu sistema. No OpenSUSE por exemplo podemos usar a ferramenta zypper para isso (equivalente do apt-get no OpenSUSE). Abra um terminal e utilize o comando

```
sudo zypper install rpm-build
```

Você precisará agora pegar a versão mais nova do .DEB do Opera-stable, que está localizada em

```
http://get.geo.opera.com/pub/opera/desktop/NUMERODAVERSAO/linux/opera-stable_NUMERODAVERSAO_amd64.deb
```

Exemplo: Versão de 13/7/2015

```
http://get.geo.opera.com/pub/opera/desktop/30.0.1835.125/linux/opera-stable_30.0.1835.125_amd64.deb
```

Salve o arquivo .DEB que baixar na pasta ~/rpmbuild/SOURCES

Crie um arquivo chamado opera-stable.spec (pode ser na sua pasta pessoal mesmo), com o conteúdo abaixo. NOTA: Você deverá trocar NUMERODAVERSAO pelo número da versão do arquivo .DEB que você baixou!

```
%define appname opera
%define debug_package %{nil}

Summary:    Fast and secure web browser
Name:       opera-stable
Version:    NUMERODAVERSAO
Release:    0
Group:      Applications/Internet
License:    Proprietary
URL:        http://www.opera.com/browser
Source0:
http://get.geo.opera.com/pub/%{appname}/desktop/%{version}/linux/%{name}_%{v
ersion}_amd64.deb
Vendor:     Opera Software ASA
Packager:   ruario

%description
Opera is a fast, secure and user-friendly web browser. It
includes web developer tools, news aggregation, and the ability
to compress data via Opera Turbo on congested networks.

%prep

%setup -T -c

%build

%install

# Not needed on Fedora but it is on some other distros
mkdir -p "%{buildroot}"

# Unpack the deb, correcting the lib directory and removing debian
directories
ar p %{SOURCE0} data.tar.xz | tar -xJf- -C %{buildroot} \
  --transform="s,/usr/lib/.*-linux-gnu,%{_libdir}," \
  --exclude="./usr/share/lintian" \
  --exclude="./usr/share/menu"

# Fix the location of the doc directory on OpenSUSE
%if 0%{?suse_version}
  mkdir -p "%{buildroot}/%{_defaultdocdir}"
  mv "%{buildroot}/usr/share/doc/%{name}"
"%{buildroot}/%{_defaultdocdir}/%{name}" 2>/dev/null ||:
```

```
%endif

# Set the correct permissions on the sandbox
chmod 4755 %{buildroot}%{_libdir}/%{appname}/opera_sandbox

# Correct the symlink due to changed lib directory
ln -fs %{_libdir}/%{appname}/%{appname} %{buildroot}%{_bindir}/%{appname}

%post

# Setup icons
touch -c /usr/share/icons/hicolor
if command -v gtk-update-icon-cache >/dev/null 2>&1; then
  gtk-update-icon-cache -tq /usr/share/icons/hicolor 2>/dev/null ||:
fi

# Setup desktop file
if command -v update-desktop-database >/dev/null 2>&1; then
  update-desktop-database -q /usr/share/applications 2>/dev/null ||:
fi

%postun

# Remove compatibility symlinks
if [ -e "%{_libdir}/%{appname}/lib/libudev.so.0" ]; then
  rm -f %{_libdir}/%{appname}/lib/libudev.so.0
fi

if [ -e "%{_libdir}/%{appname}/lib/libcrypto.so.1.0.0" ]; then
  rm -f %{_libdir}/%{appname}/lib/libcrypto.so.1.0.0
fi

# Remove directories left behind due to compatibility symlinks
if [ -d "%{_libdir}/%{appname}/lib" ]; then
 rmdir --ignore-fail-on-non-empty %{_libdir}/%{appname}/lib
fi

if [ -d "%{_libdir}/%{appname}" ]; then
  rmdir --ignore-fail-on-non-empty %{_libdir}/%{appname}
fi

%clean
rm -rf %{buildroot}

%files
%{_defaultdocdir}/%{name}
%{_bindir}/%{appname}
%{_libdir}/%{appname}
%{_datadir}/applications/*.desktop
%{_datadir}/icons/*
%{_datadir}/pixmaps/*
```

No caso do nosso exemplo do dia 13/7/2015, número da versão é seria substituído por 30.0.1835.125

Baixado o .DEB mais novo na pasta ~/rpmbuild/SOURCES/ e criado o arquivo opera-stable.spec com o número da versão correspondente na sua pasta pessoal, abra um terminal e execute o seguinte comando:

```
rpmbuild -bb opera-stable.spec
```

Esse processo irá demorar um pouco e irá gerar um arquivo .RPM que estará na pasta

```
~/rpmbuild/RPMS/x86_64/
```

.

Assim, basta instalar esse arquivo RPM. No OpenSUSE por exemplo, basta um clique duplo e fornecer a senha de administrador para fazer a instalação.

Toda vez que sair uma nova versão do Opera será necessário repetir toda operação acima. Estou estudando um método de automatizar o processo de baixar a versão mais nova, compilar um pacote RPM e instalar.

**VELHO**

A continuação do passo-a-passo antigo está abaixo.

O script é esse: install-opera.sh.tar.gz Você deve descompactar o arquivo, dar permissão de execução e chamá-lo com a seguinte linha de comando para instalar a versão estável mais atual:

```
sudo ./install.sh --stable
```

e, se quiser ver o código:

```
#!/bin/sh

available () {
  command -v "$1" >/dev/null 2>&1
}

updatedbs () {
  # Setup menu entries
  if available update-desktop-database; then
    update-desktop-database -q /usr/local/share/applications
  fi

  # Setup icons
  touch -c /usr/local/share/icons/hicolor
  if available gtk-update-icon-cache; then
    gtk-update-icon-cache -tq /usr/local/share/icons/hicolor
  fi
}
```

```
# Uninstall function to be used by the removal script
removefiles () {
  while read f; do
    # '-e' alone would not find broken symlinks
    if [ -e "$f" -o -h "$f" ]; then
      if [ -d "$f" ]; then
        if ! ls -A "$f" | grep -q ^; then
          # Don't remove a symlink pointing to a directory, as it could have
          # been created by the user or the distribution
          if [ ! -h "$f" ]; then
            rmdir -v "$f"
          fi
        fi
      else
        rm -v "$f"
      fi
    fi
  done
} #

# Ar (Binutils) or BSD tar are needed to extract from a Debian package
if available bsdtar; then
  AR_EXTRACT="bsdtar xOf"
elif available ar; then
  AR_EXTRACT="ar p"
else
  echo "You must install BSD Tar or GNU Binutils to use this script" >&2
  exit 1
fi

# Quit if you can't write to /usr/local
if [ ! -w "/usr/local" ]; then
  echo "You do not have write permission to /usr/local" >&2
  echo "Re-run this script as root or prefaced with sudo, e.g." >&2
  echo "  \$ sudo $0 $@" >&2
  exit 1
fi

# Check if automatic download has been selected
while [ 0 ]; do
  if [ "$1" = "-d" -o "$1" = "--developer" ]; then
    OPERA_STREAM=opera-developer
    shift 1
  elif [ "$1" = "-b" -o "$1" = "--beta" ]; then
    OPERA_STREAM=opera-beta
    shift 1
  elif [ "$1" = "-s" -o "$1" = "--stable" ]; then
    OPERA_STREAM=opera-stable
    shift 1
  else
    break
```

```
  fi
done

DESTROY_DEB=N
OPERA_AUTO_FETCH=N

if [ -n "$OPERA_STREAM" ]; then
  OPERA_AUTO_FETCH=Y
  DESTROY_DEB=Y
  # Set architecture information
  ARCH=$(uname -m | sed 's/i.86/i386/')
  case "$ARCH" in
    x86_64) DEBARCH=amd64; LIBDIRSUFFIX="64" ;;
      i386) DEBARCH=$ARCH; LIBDIRSUFFIX="" ;;
         *) echo "The architecture $ARCH is not supported." >&2 ; exit 1 ;;
  esac

  # Make sure we have wget or curl
  if available wget; then
    SILENT_DL="wget -qO-"
    LOUD_DL="wget"
    DL_OUTPUT="-O"
  elif available curl; then
    SILENT_DL="curl -s"
    LOUD_DL="curl -O"
    DL_OUTPUT="-o"
  else
    echo "Install wget or curl" >&2
    exit 1
  fi

  # Work out the latest stable Google Chrome if VERSION is unset
  OPERA_VERSION=$($SILENT_DL
http://deb.opera.com/opera/dists/stable/non-free/binary-$DEBARCH/Packages.gz
| gzip -d | grep -A1 -x "Package: $OPERA_STREAM" | sed -n "/Version/s/.*
//p")

  # Error out if $OPERA_VERISON is unset, e.g. because previous command
failed
  if [ -z "$OPERA_VERSION" ]; then
    echo "Could not work out the latest version of $OPERA_STREAM for $ARCH;
exiting" >&2
    exit 1
  fi

  if [ -e "/usr/local/lib$LIBDIRSUFFIX/$(echo $OPERA_STREAM | sed 's/\-
stable//')/VERSION_$OPERA_VERSION" ]; then
    echo "The latest version of $OPERA_STREAM ($OPERA_VERSION) is already
installed; exiting"
    exit 0
```

```
  fi

  OPERA_DEB=$(mktemp -t opera-deb.XXXXXX)
  $LOUD_DL
http://deb.opera.com/opera/pool/non-free/o/$OPERA_STREAM/${OPERA_STREAM}_${O
PERA_VERSION}_${DEBARCH}.deb $DL_OUTPUT $OPERA_DEB
  if ! [ "$?" = 0 ]; then
    echo "Download failed!" >&2
    exit 1
  fi
fi

# Check if this is being run as a self extractor
if [ -z "$1" -a -z "$OPERA_DEB" ]; then
  if [ $(sed '1,/^exit$/d' "$0" | head -n 1 | wc -l) = "1" ]; then
    OPERA_DEB=$(mktemp -t opera-deb.XXXXXX)
    sed '1,/^exit$/d' "$0" > "$OPERA_DEB"
    DESTROY_DEB=Y
  else
    echo "You must provide the name of an Opera package, e.g." >&2
    echo "  # $(basename $0) opera.deb" >&2
    exit 1
  fi
fi

# Run some checks to see if a proper package was provided
if [ -z "$OPERA_DEB" ]; then
  if ! echo "$1" | grep -q 'opera.*\.deb$'; then
    echo "$1 is not named like an Opera Debian package" >&2
    exit 1
  fi
  if [ -r "$1" ]; then
    OPERA_DEB="$1"
  else
    echo "$1 is either not present or cannot be read" >&2
    exit 1
  fi
fi

# Extract information from control file
if [ -z "$OPERA_STREAM" ]; then
  OPERA_STREAM_VERSION_DEBARCH=$($AR_EXTRACT "$OPERA_DEB" control.tar.gz |
tar -xzOf- ./control | sed -n 's/^Package: //p;s/^Version:
//p;s/^Architecture: //p')
  if [ -z "$OPERA_STREAM_VERSION_DEBARCH" ]; then
    echo "Could extract the package name and architecture from the control
file" >&2
    exit 1
  else
    OPERA_STREAM=$(echo $OPERA_STREAM_VERSION_DEBARCH | cut -d' ' -f1)
    OPERA_VERSION=$(echo $OPERA_STREAM_VERSION_DEBARCH | cut -d' ' -f2)
```

```
    DEBARCH=$(echo $OPERA_STREAM_VERSION_DEBARCH | cut -d' ' -f3)
  fi

  # Set the architecture
  case "$DEBARCH" in
    amd64) ARCH=x86_64; LIBDIRSUFFIX="64" ;;
        *) ARCH=$DEBARCH; LIBDIRSUFFIX="" ;;
  esac
fi

# If Opera is already installed, assume this is an upgrade and remove it
first
UNINSTALL_OPERA_SCRIPT=/usr/local/bin/remove-$OPERA_STREAM
if [ -x "$UNINSTALL_OPERA_SCRIPT" ]; then
  echo "Removing previously installed Opera first"
  sleep 1
  . "$UNINSTALL_OPERA_SCRIPT"
fi

# Stop the script as soon as there is a problem
set -eu

# Extract files from the Debian package to a temporary location
OPERA_FILES=$(mktemp -d -t opera-files.XXXXXX)
printf "\nUncompressing Opera ...\n"
mkdir -p "$OPERA_FILES"
$AR_EXTRACT "$OPERA_DEB" data.tar.xz | xz -d | tar -xf- -C "$OPERA_FILES" \
  --transform="s,^\./usr,./usr/local,;s,/lib/${ARCH}-linux-
gnu,/lib$LIBDIRSUFFIX," \
  --exclude="./usr/share/lintian" --exclude="./usr/share/menu"

# Create the first part of the uninstall script
mkdir -p "$(dirname $OPERA_FILES/$UNINSTALL_OPERA_SCRIPT)"
sed -n '1,/^} #$/p' "$0" > "$OPERA_FILES/$UNINSTALL_OPERA_SCRIPT"

# Remove the last part of stable stream name
OPERA_STREAM=$(echo $OPERA_STREAM | sed 's/\-stable//')

# Setup udev symlink if needed
cd "$OPERA_FILES"
if ! [ -e "/lib$LIBDIRSUFFIX/libudev.so.0" -o -e
"/usr/lib$LIBDIRSUFFIX/libudev.so.0" -o -e "/lib/${ARCH}-linux-
gnu/libudev.so.0" -o -e "/usr/lib/${ARCH}-linux-gnu/libudev.so.0" -o -e
"/lib/libudev.so.0" -o -e "/usr/lib/libudev.so.0" ]; then
  mkdir -p usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib
  if [ -e "/lib$LIBDIRSUFFIX/libudev.so.1" ]; then
    ln -s /lib$LIBDIRSUFFIX/libudev.so.1
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libudev.so.0
  elif [ -e "/usr/lib$LIBDIRSUFFIX/libudev.so.1" ]; then
    ln -s /usr/lib$LIBDIRSUFFIX/libudev.so.1
```

```
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libudev.so.0
  elif [ -e "/lib/${ARCH}-linux-gnu/libudev.so.1" ]; then
    ln -s /lib/${ARCH}-linux-gnu/libudev.so.1
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libudev.so.0
  elif [ -e "/usr/lib/${ARCH}-linux-gnu/libudev.so.1" ]; then
    ln -s /usr/lib/${ARCH}-linux-gnu/libudev.so.1
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libudev.so.0
  elif [ -e "/lib/libudev.so.1" ]; then
    ln -s /lib/libudev.so.1
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libudev.so.0
  elif [ -e "/usr/lib/libudev.so.1" ]; then
    ln -s /usr/lib/libudev.so.1
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libudev.so.0
  else
    echo "Neither libudev.so.0 nor libudev.so.1 was found." >&2
    exit 1
  fi
fi

# Setup libcrypto symlink on Fedora and derivatives
cd "$OPERA_FILES"
if [ -e "/usr/lib$LIBDIRSUFFIX/libcrypto.so.10" ]; then
  mkdir -p usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib
  ln -s /usr/lib$LIBDIRSUFFIX/libcrypto.so.10
usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/lib/libcrypto.so.1.0.0
fi

# Correct the Opera sandbox permissions
chmod 4755 usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/opera_sandbox

# Record the version number in the package
touch usr/local/lib$LIBDIRSUFFIX/$OPERA_STREAM/VERSION_$OPERA_VERSION

# Finish uninstall script
printf 'set -e\nremovefiles << FILE_LIST\n' >> ".$UNINSTALL_OPERA_SCRIPT"
find . ! -type d ! -print | sed 's,\.,,' | grep -vF
"$UNINSTALL_OPERA_SCRIPT" >> ".$UNINSTALL_OPERA_SCRIPT"
find . -depth -type d -print | sed 's,\.,,' | \
grep -vxE '(^$|/usr(/local(/bin|/lib(64)?|/share(/doc|/man(/man1)?)?)?)?)'
>> ".$UNINSTALL_OPERA_SCRIPT"
printf "FILE_LIST\nupdatedbs\nrm -v \"$UNINSTALL_OPERA_SCRIPT\"\n" >>
".$UNINSTALL_OPERA_SCRIPT"
chmod 755 ".$UNINSTALL_OPERA_SCRIPT"

# Install the files only, *not* directories.
# This avoids changing system directory permissions/ownership
printf "Installing Opera ...\n\n"
find . ! -type d | tar -cf- -T- | tar -xf- -C /

# Note: Originally I used a cpio instead of a tar pipe but some
# systems might not have cpio installed by default.
```

```
# find . ! -type d | cpio --quiet -pmd /

# Remove temporary files
cd - >/dev/null
rm -r "$OPERA_FILES"
if [ "$DESTROY_DEB" = "Y" ]; then
  rm "$OPERA_DEB"
fi

# Update the desktop and icons databases
updatedbs

# And ... we're done! ;)
echo 'Opera was successfully installed into "/usr/local/"'.
printf "\nTo uninstall, issue the following as root (or prefaced with
sudo):\n\n"
printf "  $UNINSTALL_OPERA_SCRIPT\n\n"
exit
```