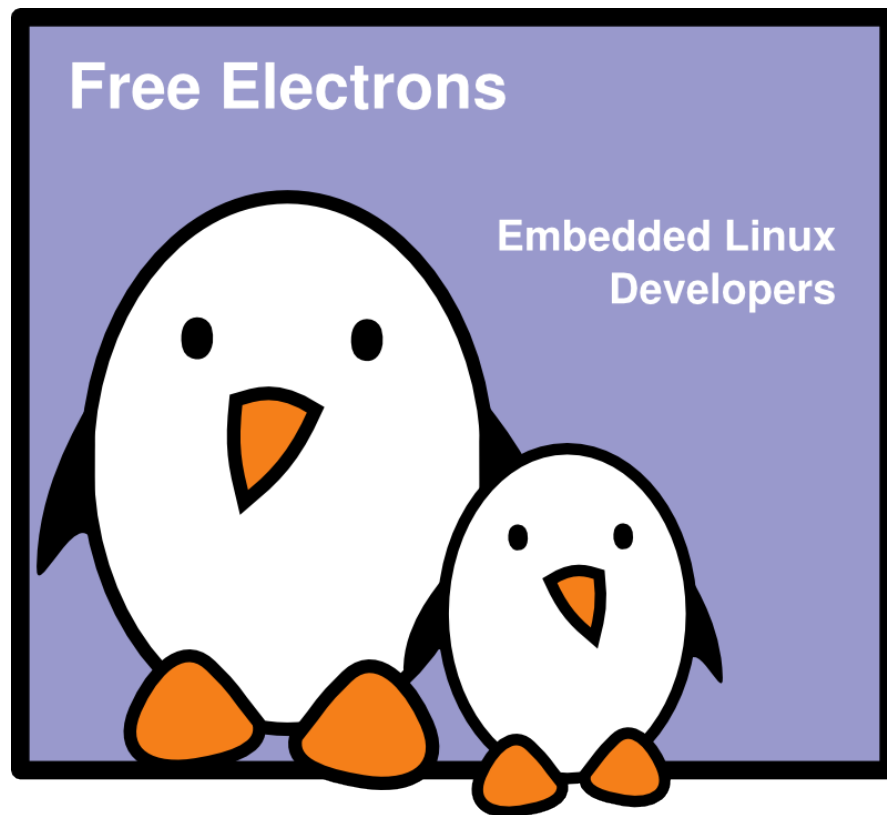




SSH

Thomas Petazzoni
Free Electrons





Rights to copy

© Copyright 2008-2009, Free Electrons
feedback@free-electrons.com

Document sources, updates and translations:
<http://free-electrons.com/docs/ssh>

Corrections, suggestions, contributions and translations are welcome!

Latest update: Sep 15, 2009



Attribution – ShareAlike 3.0

You are free

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions



Attribution. You must give the original author credit.



Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

License text: <http://creativecommons.org/licenses/by-sa/3.0/legalcode>



Introduction

- ▶ SSH stands for *Secure SHell*
- ▶ SSH is a secure communication protocol that allows remote login, file transfer and port tunneling, normalized by RFC 4251, 4252, 4253 and 4254.
- ▶ Replacement for telnet, rlogin, rsh, etc.
- ▶ On Linux, the main implementation is **OpenSSH**, with both the server and client programs
- ▶ A smaller implementation for embedded systems called **Dropbear** is also available
- ▶ On Windows, **Putty** is one of the free SSH client available.



Installation and basic usage

- ▶ OpenSSH is available as a package in all GNU/Linux distributions
- ▶ On Ubuntu, two packages are available
 - ▶ openssh-client, the client programs
 - ▶ openssh-server, the server program
- ▶ Connecting to an SSH server is as simple as
`ssh username@hostname`
- ▶ ssh will prompt for the user password and log in to the remote system.



File transfer and X forwarding

- ▶ Files can be transferred using the scp client program

```
scp myfile1 myfile2 \  
username@hostname:~/dest/directory/  
scp -r mydirectory user@host:~/dest/
```
- ▶ With ssh -x option, one can tell ssh to enable X11 forwarding
 - ▶ It allows graphical applications run on the remote host to be displayed on the local screen
 - ▶ On the server, X11Forwarding must be enabled in the configuration file /etc/ssh/sshd_config.



Remote execution

- ▶ ssh not only allows to connect to a remote host, but also allows remote execution of commands
 - ▶ `ssh user@host ls`
 - ▶ This is very useful in shell scripts, for example
- ▶ ssh is also used by other programs as a transport layer
 - ▶ rsync, the synchronisation tool, can work over ssh
`rsync -e ssh ~/work user@workhost:~/work`
 - ▶ CVS, Subversion and most of the version control tools can work over SSH



Skipping the password with keys

- ▶ An interesting feature of SSH is that you can bypass the password step by using cryptographic keys
- ▶ First, generate a private and public SSH key using `ssh-keygen -t dsa`
- ▶ It will prompt you for a passphrase, which will be required to «unlock» your private key everytime you use time
- ▶ The key has been generated in
 - ▶ `~/.ssh/id_dsa`, the private key, that no one should have access to
 - ▶ `~/.ssh/id_dsa.pub`, the public key, that you can transfer publicly to everybody



Skipping the password with keys (2)

- ▶ Now, you need to transfer the public key to the hosts you want to connect to
 - ▶ `ssh-copy-id user@host`
- ▶ The public key has been transferred to the remote host, and you should see it in `~/.ssh/authorized_keys` on the remote host
- ▶ Trying to login to the remote host should ask you the passphrase of the private key
- ▶ This allows to replace our dozens of different passwords by a single passphrase, which is easier to remember.



Skipping the password with keys (3)

- ▶ `ssh-agent` allows to avoid giving the passphrase at every login. It keeps the passphrase in memory, either forever or for a limited time
- ▶ Run the agent: `$(eval ssh-agent)`
 - ▶ Will run the `ssh-agent` program
 - ▶ Will set a few environment variables so that the other `ssh` programs can connect to the agent
- ▶ Give the passphrase to the agent: `ssh-add`
- ▶ The other `ssh` programs can now login to remote hosts that know about your public key without entering the password



Skipping the password with keys (4)

- ▶ The environment variables set by ssh-agent disappear when you exit the current shell
- ▶ The best solution is to start the ssh-agent before starting the X server so that all your applications will have access to these environment variables
- ▶ This is usually done by default on most distributions, including Ubuntu
 - ▶ The file `/etc/X11/Xsession.options` sets the `use-ssh-agent` option
 - ▶ A script in `/etc/X11/Xsession.d/` starts the agent if the `use-ssh-agent` option is set



Skipping the password with keys (5)

- ▶ The process of telling the agent your passphrase can be further improved by
 - ▶ Installing a graphical ssh-add program: `ssh-askpass-gnome` for Gnome or `ksshaskpass` for KDE (only available in the next Ubuntu version)
 - ▶ Running `ssh-add` automatically when the graphical environment starts. The exact configuration depends on your window manager.



Port tunneling

- ▶ SSH can also be used to tunnel ports
- ▶ Create a local port that connects to a remote host through a SSH connection to another host
 - ▶ `ssh -L 12345:localhost:25 user@host`
 - ▶ Any connection on the local port 12345 will in fact reach port 25 on the destination, through an encrypted tunnel
- ▶ Create a remote port that connects to a host through a SSH connection to localhost
 - ▶ `ssh -R 4242:kernel.org:80 user@host`
 - ▶ Any connection on the remote host port 4242 will in fact reach port 80 of kernel.org through an encrypted tunnel



Configuration file

- ▶ SSH stores a configuration file in `~/.ssh/config`
- ▶ It can be used to set global options, but also per-host options, like
 - ▶ `Host openmoko`
 - ▶ `HostName 192.168.0.202`
 - ▶ `User root`
- ▶ Using these options, running “`ssh openmoko`” will connect automatically to IP `192.168.0.202` with the root login.



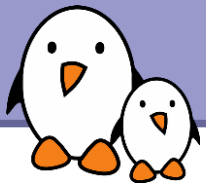
Practical lab – Using SSH

Time to start **Lab** !

- ▶ Ask your neighbor to create an account for you
- ▶ Login to your neighbor system using ssh
- ▶ Set up the keys to login without entering any password

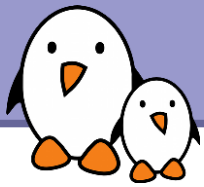


Thanks




To people who sent corrections, suggestions or improvements

▶ Guillaume Lelarge



Related documents



Free Electrons

Embedded Freedom

HOME DEVELOPMENT SERVICES TRAINING DOCS COMMUNITY COMPANY BLOG

Recent blog posts

ELC Europe in Grenoble

Free Electrons at ELC

Linux kernel 2.6.29 - New features for embedded users

The Buildroot project begins a new life

FOSDEM 2009 videos

USB-Ethernet device for Linux

Program for Embedded Linux Conference 2009 announced

Public session changes


Real hardware in our training sessions

Call for presentations for the LSM embedded track

Docs

Most of the below documents are presentations used in our [training sessions](#), or in technical conferences.

License

 All our documents are available under the terms of the [Creative Commons Attribution-ShareAlike 3.0 license](#). This essentially means that you are free to download, distribute and even modify them, provided you mention us as the original authors and that you share these documents under the same conditions.

Linux kernel

- [Embedded Linux kernel and driver development](#)
- [New features in Linux 2.6](#) (since 2.6.10)
- [Kernel initialization](#)
- [Porting Linux to new hardware](#)
- [Power management in Linux](#)
- [Linux PCI drivers](#)
- [Block device drivers](#)
- [Linux USB drivers](#)
- [DMA](#)

Architecture specific documents

- [ARM Linux specifics](#)
- [Linux on TI OMAP processors](#)

Embedded Linux system development

- [Embedded Linux system development](#)
- [Real time in embedded Linux systems](#)
- [Block filesystems](#)
- [Flash filesystems](#)
- [Free software development tools](#)
- [The U-boot bootloader](#)
- [The GRUB bootloader](#)
- [The blob bootloader](#)
- [Hotplugging with udev](#)
- [Introduction to uClinux](#)
- [Java in embedded Linux](#)
- [Embedded Linux optimizations](#)
- [Audio in embedded Linux systems](#)
- [Multimedia in embedded Linux systems](#)
- [Embedded Linux From Scratch... in 40 minutes!](#)
- [Building embedded Linux systems with Buildroot](#)
- [Developing embedded distributions with OpenEmbedded](#)
- [The Scratchbox development environment](#)

Miscellaneous

- [Introduction to the Unix command line](#)
- [SSH](#)
- [Linux virtualization solutions \(with an embedded perspective\)](#)
- [Advantages of Free Software and Open Source in embedded systems](#)
- [Introduction to GNU/Linux and Free Software](#)

All our technical presentations on <http://free-electrons.com/docs>

- ▶ Linux kernel
- ▶ Device drivers
- ▶ Architecture specifics
- ▶ Embedded Linux system development



How to help

You can help us to improve and maintain this document...

- ▶ By sending corrections, suggestions, contributions and translations
- ▶ By asking your organization to order development, consulting and training services performed by the authors of these documents (see <http://free-electrons.com/>).
- ▶ By sharing this document with your friends, colleagues and with the local Free Software community.
- ▶ By adding links on your website to our on-line materials, to increase their visibility in search engine results.

Linux kernel

- Linux device drivers
- Board support code
- Mainstreaming kernel code
- Kernel debugging

Embedded Linux Training

All materials released with a free license!

- Unix and GNU/Linux basics
- Linux kernel and drivers development
- Real-time Linux, uClinux
- Development and profiling tools
- Lightweight tools for embedded systems
- Root filesystem creation
- Audio and multimedia
- System optimization

Free Electrons

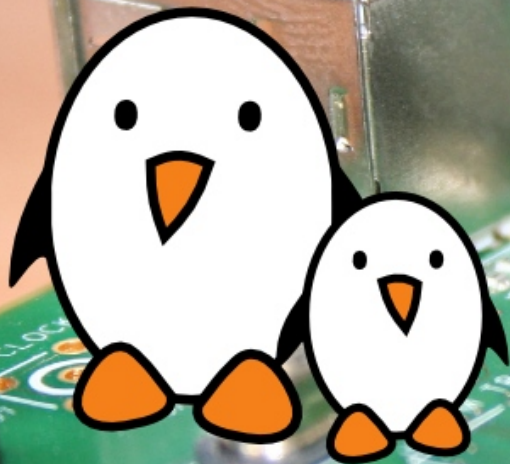
Our services

Custom Development

- System integration
- Embedded Linux demos and prototypes
- System optimization
- Application and interface development

Consulting and technical support

- Help in decision making
- System architecture
- System design and performance review
- Development tool and application support
- Investigating issues and fixing tool bugs



Free Electrons
Embedded Linux Experts

<http://free-electrons.com>