

Study and installation of a VOIP service on iPAQ in Linux environment

Volkan Altuntas

Chaba Ballo

Olivier Dole

Jean-Romain Gotteland

ENSEIRB 2002

Summary

1. Introduction

2. Presentation

- a. iPAQ's characteristics
- b. use of Linux as operating system
- c. Interests of using a PC as workstation

3. VoIP's software

- a. Requirements
- b. Chosen application and reasons for this choice

4. iPAQ

- a. Communication between iPAQ and PC
- b. NFS
- c. Tasks done on PC
- d. Cross-compilation of software

5. Conclusion

6. Bibliography

1. Introduction

Our project is a part of Compose's project. This study's aim is to develop a programmable platform for multimedia communications. The chosen platform is the iPAQ H3660 from Motorola. Our study is to implement a VoIP service on iPAQ.

2. Presentation

a. iPAQ's characteristics

iPAQ is a PDA. Its cpu is an Intel StrongARM, rhythmic at 206 MHz. It has 64 Mo RAM and 16 Mo ROM flash. The operating system is installed on flash memory. Applications set up by user, will be stored on RAM memory. It has a TFT colour display, speaker and microphone, and its battery can supply power up to 12 hours.



b. use of Linux as operating system

The used operating system is Linux distribution Familiar. It is adjusted to iPAQ because of its light X11 graphic interface. Moreover, unlike Windows from Microsoft, Linux is not under copyright and source code is free.



c. Interests of using PC as workstation

We use a PC as workstation. This choice has been done because iPAQ is not powerful enough for compiling the application code. So PC and iPAQ are connected by a serial link and cross compiler's use enables us to compile application code on our workstation in a language understood by iPAQ, that is to say by its cpu.

3. VoIP's software

a. Requirements

The chosen service for validating programmable platform is a phone service: voice over IP. In one hand, we had to list all VoIP offers as iPAQ's characteristics were considered. Indeed, software had to be compatible with Linux Familiar and only occupy a bit space as it only left 5 Mo free. In the other hand, there was need to have access to software's sources in order they can be modified by adding new elements for example. Moreover, new elements' implementation would be eased by an API.

b. Chosen application and reasons for this choice

RAT has been chosen because it met constraints and it was Mr Escrig group's study subject. Moreover, Motorola's research team has generated executables for iPAQ and their download is available on the web what enables us to see the result to obtain and to test RAT. Robust Audio Tool (RAT) is an open-source audio conferencing and streaming application that allows users to participate in audio conferences over the internet. This can be between two participants directly, or between a group of participants on a common multicast group.

RAT requires no special features for point-to-point communication, just a network connection and a soundcard. For multiparty conferencing, RAT uses

IP multicast and therefore all participants must reside on a multicast capable network. RAT is based on IETF standards, using RTP above UDP/IP as its transport protocol. RAT features a range of different rates and quality codecs, receiver based loss concealment to mask packet losses, and sender based channel coding in the form of redundant audio transmission. It offers better sound quality relative to the network conditions than most audio tools available. It also features encryption so you can keep your conversations private.

RAT runs on a range of platforms: FreeBSD, HP-UX, IRIX, Linux, NetBSD, Solaris, SunOS, and Windows 95/NT.

RAT is *just* an audio application, it does not perform call services like user location, and neither does it listen to session announcements to discover advertised multicast sessions

4. iPAQ

a. Communication between iPAQ and PC

Minicom application is the main tool which enables us to communicate with our iPAQ. This application uses a file transfer protocol, called Kermit. Indeed, by using Kermit, a link can be established between two equipments, similar or not, and so connexion parameters can be configured.



```
A - Serial Device      : /dev/ttyS0
B - Lockfile Location  : /var/lock
C - Callin Program     :
D - Callout Program    :
E - Bps/Par/Bits      : 115200 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No

Change which setting? █
```

Moreover, connexion can be configured through Minicom's user interface. Thus, main commands can be used as logging on iPAQ or transferring data between iPAQ and our workstation.

The final step was to establish a PPP connexion between the iPAQ and our workstation what enables us to have all the TCP/IP protocols stack which is necessary to run RAT. This establishment is done in order to do all the necessary tasks for the project's achievement. Indeed, these protocols allow us to run different commands such as a VoIP application's transfer to iPAQ and to check that this application works by testing it. As a matter of fact, RAT is a piece of software which cannot run alone; I mean that RAT needs that other hosts are present on network to communicate with someone. As a comparison, a phone is useful only if it is connected to a network where other phones are connected too.

```

Minicom Command Summary

Commands can be called by CTRL-A <key>

Main Functions                                Other Functions
Dialing directory..D  run script (Go)...G | Clear Screen.....C
Send files.....S     Receive files.....R | cOnfigure Minicom..O
comm Parameters...P  Add linefeed.....A | Suspend minicom...J
Capture on/off....L  Hangup.....H       | eXit and reset....X
send break.....F     initialize Modem...M | Quit with no reset.Q
Terminal settings..T  run Kermit.....K   | Cursor key mode...I
lineWrap on/off....W  local Echo on/off..E | Help screen.....Z
                                           scroll Back.....B

Select function or press Enter for none.█

Written by Miquel van Smoorenburg 1991-1995
Some additions by Jukka Lahtinen 1997-1999
i18n by Arnaldo Carvalho de Melo 1998

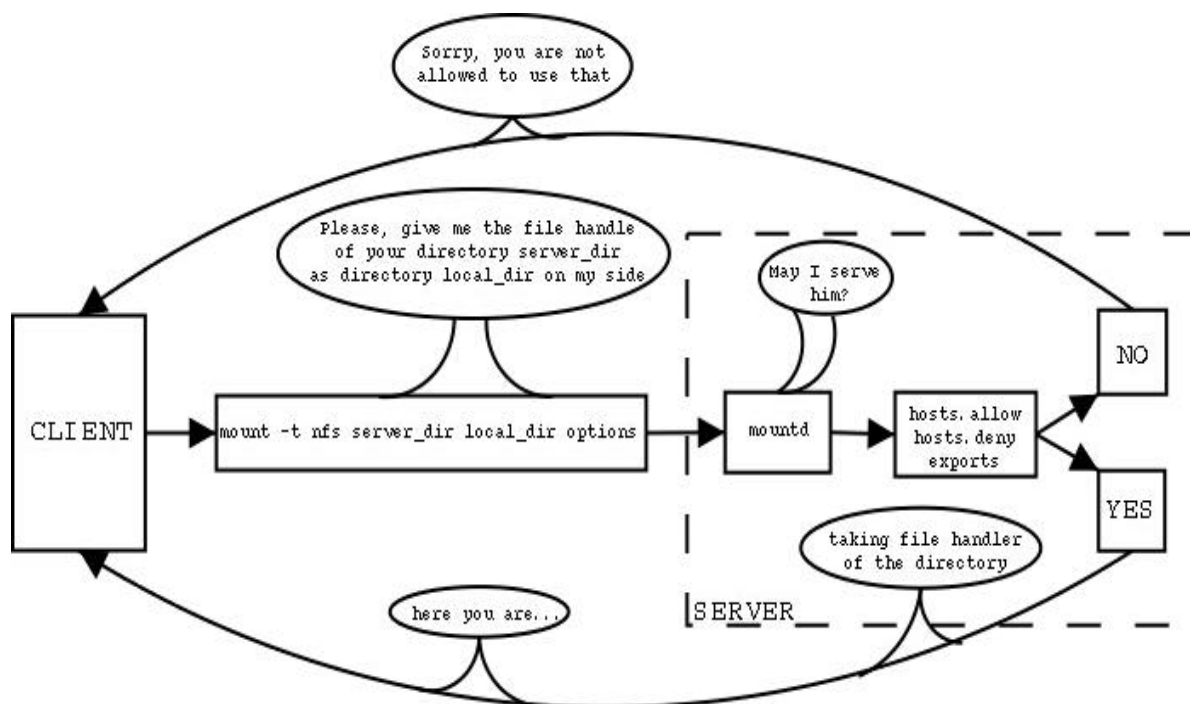
```

b. NFS

Sometimes it is wishable that two different machines share disk space. That is why there is network file system (NFS), the easiest way of sharing files and resources across a network. NFS is the way of working with files on a remote machine as if they are on your local. Thus, the use of such a service may considerably increase the iPAQ's memory storage capacities by mounting a pc's part of disk space as network file system. NFS is a great solution not only for the large available memory which is provided with the use of it, but also because it allows us not to load a program from pc to iPAQ each time a test is required for the program, furthermore only 624Ko are currently available on iPAQ. Let us see how NFS works. NFS is the most known service using remote procedure call (RPC). As an example, let us say a server machine, named tiger, keeps all \$HOME files in the directory /home. From your local machine, you would issue the command:

```
mount -t nfs tiger:/home /home (figure NFS_2.jpg)
```

According to this command, mount will try to connect the rpc.mount daemon on server tiger via RPC. The server will check if the request is permitted to mount /home from your local machine, and return a file handle that in turn may be used to serve all requests to files below /home. If the request is not allowed, the corresponding diagnostic message will be seen. When a user on your local machine tries to access a file on NFS, the kernel makes an RPC call to the NFS daemon, usually rpc.nfsd, using as parameters the name of the file, user and group ID. Thus the server, tiger, can prevent unauthorized access before sending the file handle.



c. Tasks done on PC

Since ppp link is established, iPAQ can easily communicate with other hosts through network. As a matter of fact, protocols like TCP/IP enable communications between pc and iPAQ to occur. For example X protocol uses TCP/IP to transmit information between server and client and so allows the iPAQ to manage a screen of its own on another host using the same protocol to manage his own display. Running an X program like rxvt on the iPAQ with display set to the host provides many advantages but drawbacks too. The interest of such an operation lies in having a large screen to work and using a physical keyboard to edit command or text. Of course, in such case, iPAQ's processor and operating system monitor the process, moreover the ppp link's pass-band is quite narrow due to the serial link that is why the application runs slower than usual. These drawbacks can really become annoying when a networking application is ran because such an application requires a wider pass-band and so cannot run correctly. Nonetheless the issue of the pass-band can be solved by connecting the iPAQ to a 10Mbit/s WLAN (Wireless Local Area Network) thanks to a PCMCIA extension card.

d. Cross-compilation of software

In order to adjust RAT software to iPAQ, cross compilation is required. Cross compilation principle is easy to understand: if sources of the piece of software to port are compiled with a common compiler, an executable will be generated for the processor of the pc. Whereas compiling them with a cross compiler for iPAQ will generate an executable for the iPAQ. Most software running on the iPAQ are written in C. Of course, C program's behaviour depends on the architecture on which the program is run. Its

behaviour can depend on particularities of operating system, compiler, libraries, and CPU. Nevertheless problems may occur, for example the name of a file representing a device may differ between OS. Endianless issues are interesting to understand how a processor interprets instructions generated by compilation. There are two basic memory layouts used by most computers, designated big endian and little endian. On big endian machines, the most significant byte of an object in memory is stored at the most significant address. Conversely, on little endian machine, the least significant byte is stored at the address closest to zero. For example:

```
int x = 0xaabbccdd
```

```
    unsigned char b = (unsigned char*)&x
```

On a big endian machine, b would receive the most significant byte of x, 0xaa.

On little endian machine, b would receive the least significant byte of x: 0xdd.

The pc architecture is little endian. Many ARM processors support both

modes,

and Linux distribution on the iPAQ is little endian. Endians problems can arise when sharing binary data between machines of different endianness.

RAT's executables have been downloaded on Motorola's web site. In fact this part was easier than expecting. Before trying cross compilation, we thought that many problems could occur, for example filenames of some devices could differ from Linux for pc to iPAQ, or some graphical difficulties could emerge. As regards devices, audio device filename is /dev/dsp on pc and /dev/sound/dsp on iPAQ, nonetheless iPAQ has a file named /dev/dsp which is a link to /dev/sound/dsp and which solves any possible problem. Another fear was that RAT is a graphical application and so a problem of display during cross compilation could have happened due to a missing library or something like that. In fact, by compiling exactly like for the Linux version expecting that Makefile has been modified (name of compiler and path where to find some libraries), few problems have been encountered. A tricky point was about the executable named tcl2c, as a matter of fact this programs gets a tcl's script to transform it into C code. This executable is created and then run during compilation, and so cross compilation if the piece of software is cross compiled, so to run correctly, this program has to be compiled with a compiler and not with a cross compiler. Because if cross compilation is used to generate tcl2c, pc will not be able to continue cross-compiling as soon as tcl2c will be required. Indeed tcl2c will be an executable for iPAQ and cross compilation will run on pc.

5. Conclusion

As a conclusion, few tests have been done for RAT on iPAQ. The reasons are that pass band is quite narrow due to the serial link which twists voice signal and so reduces voice's quality. Moreover, the workstation is not equipped with audio output and input like speakers or microphone, therefore tests were run at home. That is why it would be interesting to connect iPAQ with a wireless local area network (WLAN) in order to complete tests and to have a real mobility.

6. Bibliography

- iPAQ's characteristics
www.compaq.com/products/handhelds/pocketpc/
- Connection between pc and iPAQ
handhelds.org
- RAT
internet2.motlabs.com/ipaq/index.htm
- NFS
Starting share files with NFS www.linuxjournal.com (January 2002)