How did Linux become a mainstream embedded operating system?

Chris Simmonds
2net Limited

Winchester, UK
chris@2net.co.uk

***Abstract*—This paper is based on a timeline showing the various stages of development that Linux went through as it evolved from a desktop operating system for PCs to a flexible operating system which could be scaled down to embedded devices. Since the early beginnings of embedded Linux in the late 1990's, many people have contributed time and know-how that has resulted in today's Linux-based embedded operating systems. This paper highlights some of the key milestones, and the events and people behind them.**

Keywords—Linux

I. Introduction

The birth of Linux can be pinpointed to 25th August 1991 when Linus Torvalds, then a student at the University of Helsinki, announced his project on the comp.os.minix newsgroup in a famous post that began:

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

Linus' motivation was to run a Unix-like operating system on his desktop. However, the open source license and availability of the code via the Internet, meant that many developers and enthusiasts were able to get access to the code. Pretty soon Linux was running on all sorts of different systems

This paper is about a significant sector of the Linux installed base, generally known as “embedded Linux”. It is hard to pin down exactly what “embedded” means. One definition is “everything that contains a computer but doesn't look like one”, such as printers, wireless routers, smart TVs, smart thermostats and even smart light bubs. Another definition emphasizes the lack of system resources: embedded devices tend to have slow processors, and small amounts of memory and storage. Looking at it this way brings into the definition of embedded devices such as smart phones and tables, which are obviously computers, but which use many of the same techniques as the first group. I have taken a rather broad definition, which includes both definitions. With that in mind, let's see how all this came about.

II. The Early Days: 1995 to 1999

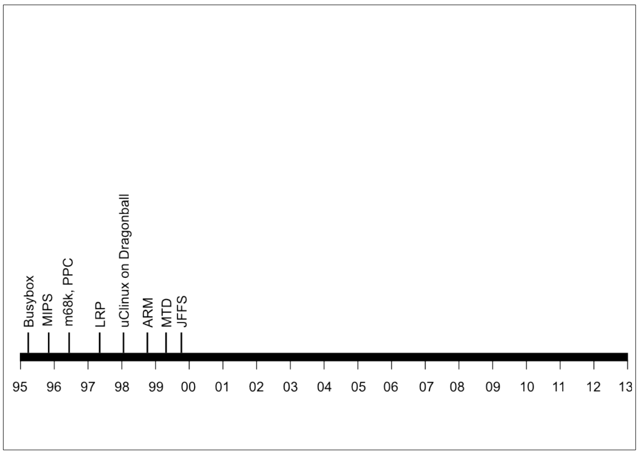

Illustration 1: Early days

During the period 1995 to 1999 the foundations were laid that would enable Linux to become an embedded operating system.

In 1995, Bruce Perens began work on a utility cammed Busybox that would enable him to create a simple Linux system that would fit onto a 3.5 inch floppy disk. His aim was to make it easy to install the Debian Linux distribution by allowing it to be booted from a floppy disk drive, which was fitted to almost all desktop PCs at that time. Both Busybox and a Linux kernel would fit into at 1.44 MiB floppy disk.

Although it was not on his mind at the time, embedded systems also tend to have a small amount of storage, and so Busybox was the ideal solution to creating a small, useful subset of a Linux distribution. Even today, a large proportion of embedded Linux systems use Busybox.

An important stepping stone was the Linux Router Project (LRP) [1], created in 1997 by Dave Cinege. He took the idea of a bootable floppy disk and added some networking utilities which could turn any PC with two or more network cards, and the right drivers, into a router. While not strictly an embedded application, it demonstrated that a useful system could be put together in a small footprint. It also formed the basis that would lead to Linux being used on many network devices.

A further step in that direction came a year later when David Täht and Greg Retkowski published a document titled the "Arlan Wireless Howto" [2], which described the process of making a wireless connection between two points using Arlan cards, and so created what was arguably the first embedded Linux wireless router [3].

As well as having a small amount of storage, another characteristic of embedded devices is the variety of CPU architectures employed, for reasons of cost, low power and the range of on-chip features. During this period, Linux was ported to several architectures that are important for embedded Linux:

1995: MIPS

1996: Motorola 68000 and the IBM/Motorola PowerPC

1999: ARM

In 1998, support for the Motorola Dragonball processor was added, as part of the effort to run Linux on the 3Com Palm Pilot PDA. The Dragonball did not have a memory management unit (MMU), and so this work created a fork of Linux that was known as micro controller Linux, or more commonly, uClinux [4]. It also lead to the creation of a new, small C library called uClibc [5], which is still in use today. uClinux was merged into mainline Linux with the 2.6 release in 2003.

The final piece of the jigsaw was support for memory based storage technologies, especially flash memory. David Woodhouse added the Memory Technology Devices (MTD) layer [6]. MTD creates a framework for accessing flash memory and is the basis of the various flash-based file systems that follow. Flash memory is a key component of most embedded devices. Shortly afterwards, Axis Communications used the MTD later to create a robust file system for flash memory. It was called jffs (journalling flash filesystem) and later evolved into jffs2 which is used in a large number of devices through to today

III. 1999: Things Start to Happen

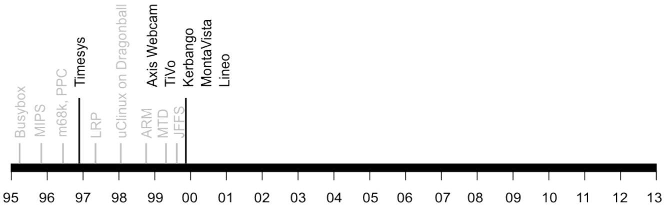

Illustration 2: Things start to happen

1999 is the year that embedded Linux started to make an impact as the first high-profile commercial products began to emerge.

TiVo [7] shiped their first-generation DVR (Digital video recorder) in March of that year. TiVo was one of the first consumer products to run Linux, and established Linux in a product segment that it dominates today. Under the hood, the first TiVo had a PowerPC 403 processor, 16 MiB RAM and a 14 GiB hard drive.

Axis Communication [8] released the 2100 network camera. Axis created the jffs filesystem, as already noted. The hardware spec for the 2100 was ETRAX-100 risc cpu, 8/16 MiB RAM, 2 MiB NOR flash, uClinux 2.0.33 [9].

The Kerbango 100E Internet radio [10] was announced. Although it never shipped as a product, it did generate a lot of press coverage, especially when 3com, which at that time was very large company, bought the product a year later. The Kerbango was important in another way. Much of the software was implemented by a new Silicon Valley company called Montavista, and it proved a good training ground for many developers.

As well as Montavista, several other embedded Linux software companies came into being in 1999. Lineo was founded by Bryan Sparks. Lineo did not last as long as Montavista, but also made important contributions to embedded Linux. In Germany, Software consultancy Denx was formed. It began work on PPCBoot, merging code from fadsboot (Dan Malek) and 8xxrom-0.3.0 (Magnus Damm). PPCBoot 0.41 was released July 2000, initially for MPC 8xx boards. PPCBoot evolved into U-Boot which is used to boot Linux on many millions of embedded boards today.

Linux Goes Mobile: 2000 to 20011

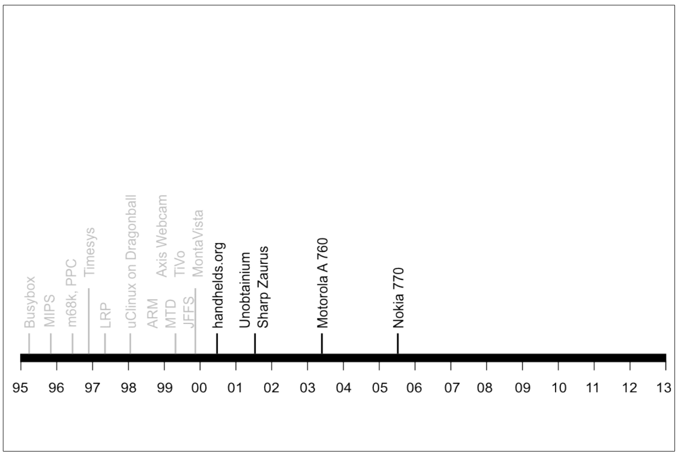

Illustration 3: Linux goes mobile

In this period, embedded Linux made inroads into the mobile market, at first hesitantly but later with more confidence.

One project has already been mentioned: the port to the Palm Pilot. The next step came in 2000 with the release of the Compaq iPAQ H3600. This was a powerful system, with an ARM architecture StrongARM processor, 64 MiB RAM, 16 MiB of flash memory and a color touch screen, which meant that it was capable of running a full Linux kernel, with memory management. The Linux port was coordinated by handhelds.org, which launched in June. This reference [11] shows the site as captured by archive.org in August 2000.

Showing the shape of things to come was **project Mercury** at Compaq CRL [12], where Jim Gettys, and others put together a prototype mobile handset consisting of an iPAQ H3600 with an add-on sleeve called the BackPAQ, which provided GSM, CDMA/802.11b, and Bluetooth wireless communication, a full VGA (640x480) camera, accelerometer and 1GiB of storage on an IBM Microdrive. The accelerometer was used to re-orientate the screen from portrait to landscape as the device was rotated. Jim named it "the Unobtainium" when talking to journalists who had heard of but not seen the device. A few hundred were made for research and development but it never went into production.

Sharp shipped the Zaurus SL-5000D PDA running Linux 2.4.10 (provided by Lineo). It had a StrongARM SA-1110 CPU @ 206 MHz, 64 MiB RAM, 16 MiB NOR flash. Note that this is the developer edition (hence the 'D' in the product name). The first commercial Linux Zaurus was the SL5500 which shipped in 2002 [13].

In 2003, Motorola launched the first commercial Linux-powered mobile handset, the A 760. It was put on sale in SE Asia and was followed up in 2004 with the A 780 in the USA. This marks the start of Linux as a cell phone operating system. The A 780 used an ARM 925 chip (an Intel PXA270) at 316 MHz, had 48 MiB flash storage and a QVGA (240 x 320 pixel) touch screen. The Linux port was provided by Montavista.

In 2007 Nokia released the 770 Internet Tablet running their Maemo Linux distribution. Maemo was built - indirectly - on the base provided by Familiar Linux/GPE and used some of the technology from a British company named OpenedHand. The 770 was the first of a line of devices leading up to the N9 phone in 2011. Although Nokia no longer exists, the Maemo operating system, now knows as Mer, continues to be as part of the Sailfish OS developed by Finnish company Jolla.

In 2008, Google launched the Android operating system, which runs on top of Linux. The first Android handset, the T-Mobile G1 (aka the HTC Dream) had a modified Linux 2.6.25 kernel. For several years, the Android kernel remained separate from mainline Linux, but at a conference in 2011, a decision was taken to work with the Google developers to integrate Android more closely into the Linux community, to the benefit of everyone. That work is still in progress, but most functions, including Android logging, wakelocks and the low memory killer ,have been integrated.

Android has turned out to be the most successful Linux distribution so far, with well over one billion current users.

IV. Embedded Linux Build Systems

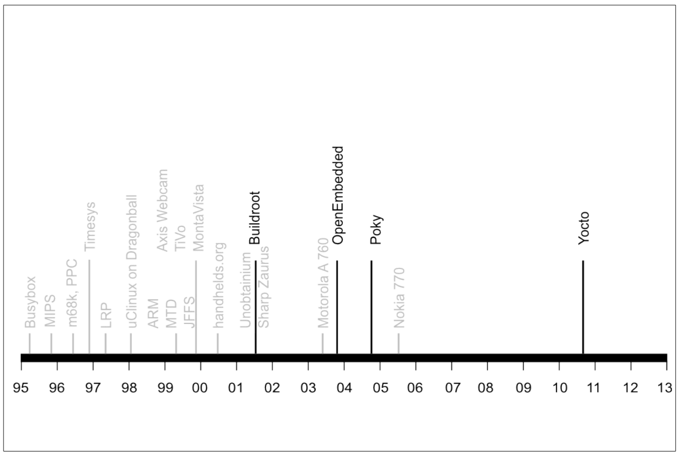

Illustration 4: Evolution of embedded build systems

In the early days of embedded Linux each project was hand crafted from scratch, which required a high level of skill. This approach is known as “Roll Your Own” Linux. Starting in 2001, a number of open source tools were developed to automate much of the process. Interestingly, they all had a basis in mobile Linux projects.

The first was Buildrot [14]. In December of 2001, it emerged from the uClinux/uClibc project as a way of building small embedded Linux systems to test uClibc. Buldroot has continued to be maintained and is widely used today.

In 2003, the OpenEmbedded Project was born, creating a common build system and code base for Familiar Linux, OpenZaurus and OpenSIMpad. OpenEmbedded is the basis of several embedded distributions today, including the Yocto Project.

Also in 2003, Matthew Allum formed OpenedHand [15], a UK-based consultancy, developing software for mobile platforms. Matthew had been working for some time at handhelds.org, especially on the problem of making effective use of the small screens then common. One thing to come out of this was a simple, full-screen X11 window manager named Matchbox and the Clutter UI library. OpenedHand worked with

Nokia on the 770 tablet and with One Laptop Per Child on the OLPCx1. A little later on, OpenedHand employee Richard Purdie created Poky Linux based on OpenEmbedded. In 2008 OpenedHand was purchased by Intel. In 2010, Intel spun off Poky Linux as the Yocto Project [16] and transfered ownership to the Linux Foundation. The Yocto Project now forms the basis for many open source and commercial build tools for embedded Linux.

V. Real-Time Linux

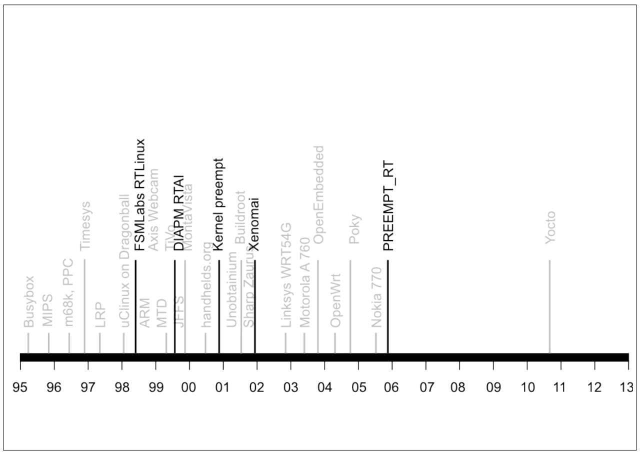

Illustration 5: Real-time Linux

The software for many embedded systems has real-time requirements, that is, they have deadlines for completing various computational tasks. Real-time systems range from processing streaming media, such as a smart TV, to controlling a robot on a production line. Throughout the nineteen nineties embedded devices often employed an RTOS (Real Time Operating System), such as VxWorks and Nucleus, even if the device itself was not real-time. Consequently, at the beginning of the era of embedded Linux, many pointed to its apparent lack of real-time facilities and questioned whether it could be considered an embedded operating system.

The first movement on improving the determinism of Linux, and so making it more suitable for real-time systems, occurred in 1998. At that time, FSMLabs released their product, called RT Linux [17]. Their approach was to partition the system into two: a real-time part running real-time application and a non-real-time part running Linux. This is known as the “dual kernel” approach. By creating a new execution environment for time critical computing it completely bypassed the problems inherent in the Linux kernel of that time. RT Linux was acquired by Wind River (now part of Intel) in 2007.

RT Linux was proprietary, and as a consequence somewhat controversial among open source developers. In 1998, the Dipartimento di Scienze e Tecnologie Aerospaziali created an open source dual-kernel real-time system, called RTAI - the RealTime Application Interface for Linux [18]. It is still being developed today.

In 2001, a second open source dual kernel system became available: Xenomai [19]. Although Xenomai has a lot in common with RTAI, and the two projects share code and ideas, there are differences. The major ones derive from the goals the projects aim for, and from their respective implementation. While RTAI is focused on lowest technically feasible latencies, Xenomai also considers clean extensibility (RTOS skins), portability, and maintainability as very important goals.

The second strand of development focused on improving Linux rather than bypassing it, and so making it available to everyone who uses Linux. There are many aspects to making the Linux kernel more deterministic, but central to everything is reducing the interrupt latency such that an interrupt that signals an external event can be processed within a given deadline.

Work began in 2000 when Ingo Molnar and Andrew Morton created the "voluntary preempt patch". This patch reduces worst-case interrupt latencies by breaking up lengthy processing cycles in the kernel into smaller chunks and re-scheduling if an interrupt handler has set the "need_resched" flag. In 2001, Robert Love took a different approach when he released the kernel preemption patch, which allows the kernel to reschedule in response to a hardware interrupt so long as it is not in a critical section. Both the voluntary preempt and kernel preemption patches were included in Linux 2.6 in 2003.

In 2005, Ingo Molnar, Thomas Gleixner and others began work on the PREEMPT_RT kernel patch which further improves the real-time behavior of Linux, bringing it close to hard real-time. Parts of this patch have been integrated into mainline Linux in the years since, including high resolution timers, threaded interrupt handlers and real-time mutexes. However the changes to make the kernel fully preemptible have not been merged, although there is hope that they will be by the end of 2016.

VI. Conclusion

In a span of a little over 20 years, Linux has become established as the foremost embedded operating system, with large volume applications in mobile, streaming media devices (smart TVs and set top boxes), and domestic WiFi routers. It is now deployed in over two billion devices.

The reasons for this adoption are first that it is available using an open source license, GPLv2. This has encouraged developers to take Linux and adopt it to their needs, many of which have been far away from the original intentions of Linus Torvalds when he created it. Second, there has been a very strong developer community around the kernel and the many other projects that provide the software needed to create an embedded Linux distribution.

It seems likely that the adoption of Linux will continue, at least for a while yet. It is being driven by the increasing power of embedded hardware and the increasing expectations of the end users. As we enter the age of smart devices interconnected into an “Internet of Things”, Linux will continue to provide the power and flexibility to make them work.

References

[1] http://en.wikipedia.org/wiki/Linux_Router_Project

[2] http://www.rage.net/wireless/wireless-howto.html

[3] http://the-edge.blogspot.co.uk/2010/10/who-invented-embedded-linux-based.html

[4] http://www.uclinux.org

[5] http://uclibc.org

[6] http://lists.infradead.org/pipermail/linux-mtd/1999-May/000000.html

[7] http://www.tivo.com

[8] http://www.axis.com

[9] http://www.axis.com/25years/milestones.htm

[10] http://en.wikipedia.org/wiki/Kerbango

[11] http://web.archive.org/web/20000815063255/http://www.handhelds.org

[12] http://web.archive.org/web/20010620015914/http://crl.research.compaq.com

[13] http://en.wikipedia.org/wiki/Sharp_Zaurus

[14] https://buildroot.org

[15] http://web.archive.org/web/20070401185003/http://projects.o-hand.com

[16] https://www.yoctoproject.org

[17] https://en.wikipedia.org/wiki/RTLinux

[18] http://www.rtai.org/

[19] https://xenomai.org/