

**Laboratory for Advanced Planning
and simulation project**

**Configuring and Installing
Debian GNU/Linux
on 64 bit HP Itanium2
PC Architecture**

Alceste Scalas

CRS4, EIP/Geometric Modelling and Monte Carlo Simulations Area

Configuring and Installing Debian GNU/Linux on a 64 bit HP Itanium2 Architecture

Alceste Scalas <alceste@crs4.it>
CRS4

(Center for Advanced Studies, Research and Development in Sardinia)

April 2002

Abstract

This document describes the work performed to install the Debian™ GNU/Linux operating system on a Hewlett-Packard™ zx6000 workstation, based on the Intel™ Itanium2™ (McKinley) processor.

Contents

1	Hardware configuration	2
2	GNU/Linux™ distribution	3
3	Installation process	3
4	Hardware support problems	4
4.1	Graphics adapter	4
4.2	HP/3Dconnexion™ SpaceBall™ controller	4
5	Software problems	5
5.1	Problems due to 64-bit incompatibilities	5
5.2	Other problems and bugs	5
6	Conclusions	6

1 Hardware configuration

- Two Itanium2™ processors at 900 Mhz, 1.5 Mb cache;
- 2 GB RAM;
- CMD Technology™ Inc PCI0649 (rev 02) IDE interface;
- IDE ATAPI DVD-ROM drive;
- Intel™ Corp. Ethernet Pro 100 NIC;
- BROADCOM™ Corporation NetXtreme™ BCM5701 Gigabit Ethernet NIC;
- LSI Logic™ 53c1030 SCSI storage controller;
- 36 GB SCSI hard disk drive;
- ATI™ Fire GL4™ graphics card;
- USB keyboard and mouse;
- HP p1130 21" monitor;
- HP/3Dconnexion™ SpaceBall™ 6DOF controller.

2 GNU/Linux™ distribution

There are several GNU/Linux™ distributions that offer an Itanium™ port. The Debian™ GNU/Linux™ 3.0r0 system has been recognized as the ideal candidate to be installed on the workstation, since it is the only GNU/Linux™ distribution that satisfies all the following requirements:

portability the Debian™ GNU operating system runs on several combinations of hardware (twelve platforms) and kernels (Linux™, GNU Hurd, FreeBSD, NetBSD) [4]. It offers a standardised operating system suitable for various needs and tasks;

up-to-date ia64 support the latest versions of the Debian™ GNU/Linux™ system are available for ia64, while other distributions' ia64 ports are only available for older versions, with obsolete packages;

clear licensing policy the Debian™ packages are clearly distinguished for their licensing requirements. Every official package must follow the Debian™ Social Contract and Debian™ Free Software Guidelines [1], and it ensures that every component of the system is available in source format, and could be freely studied, modified and redistributed. Most GNU/Linux™ distributions often include non-free packages (e. g. that could not be modified or used for commercial purposes), and it may lead to legal uncertainties during common system usage;

standardization the Debian™ GNU/Linux™ system follows the Linux™ Standard Base (LSB) [7] and Filesystem Hierarchy Standard (FHS) [8] guidelines. It offers a guarantee of compatibility with other GNU/Linux™ distributions;

robust package system the Debian™ package system supports advanced dependency tracking and conflict resolution between packages;

package management via Internet the Debian™ packages are available from a number of different locations on the Internet, and they can be automatically managed and upgraded by using the `apt` suite [3]. This powerful system allows easy administration of security updates (available from <http://security.debian.org/>), and allows to easily synchronize the packages installed on different machines running Debian™ (it is possible to create a central `apt` repository with selected software packages). Other GNU/Linux™ distributions lack this kind of installation management (or are just starting to support it), requiring complicated download-and-install sessions for every software upgrade;

safe library management the API and ABI changes between newer releases of the same library are carefully tracked during the Debian™ development stage, eventually by fixing broken library sonames, to ensure that user-developed application don't stop working during an upgrade [9];

possibility of upgrade the Debian™ development policy [2] requires that the system could smoothly upgrade to a newer release without reinstallation. This task is achieved only by few other GNU/Linux™ distributions;

open development community the Debian™ GNU operating system is mostly developed by a number of volunteers which use the system for their own work. It ensures that the platform is adequate for various niche tasks that are not covered enough by other distributions (a very common situation in the research field), and that the system stability and availability is cared before marketing tactics or strategies;

huge number of available packages the Debian™ GNU/Linux™ system has over 8000 official packages available. A number of them is not present on other GNU/Linux™ distributions, and should be compiled by hand, because their request by the public is not big enough to justify the packaging investment (a common scenario for applications that are used in research fields).

3 Installation process

The system has been installed from the first bootable Debian™ CD-ROM (in its NON-US version, which includes cryptographic software that could not be exported from the USA [5]). Since we didn't download the remaining 6 CD-ROMs of the distribution, the packages that were not included in the first one were downloaded on-the-fly via `apt`, from <http://debian.fastweb.it/debian/>.

The installation process is straightforward, and is well documented on the Debian™ GNU/Linux™ Installation Manual [6]. The only problems may come during the creation of the fat32 booting partition: it will host the kernel images and the configuration files required by the ia64 Linux™ bootloader (elilo), that in turn will be invoked by the booting firmware (EFI) [10]. If that booting partition is too small, it won't be possible to store the

Linux™ images, and the system won't boot. A size of 100 Mb should avoid all problems, leaving enough room to install and experiment with several kernel versions.

The following partition scheme has been adopted for the installation (device file names are relative to the `/dev/scsi/host0/bus0/target0/lun0/` directory):

<i>Device</i>	<i>Start</i>	<i>End</i>	<i>Filesystem</i>	<i>Mount point</i>
part1	0,017	200,000	fat16	(EFI/elilo boot partition)
part2	201,000	500,000	ext3	/
part3	501,000	4500,000	ext3	/usr/
part4	4501,000	14500,000	ext3	/home/
part5	14501,000	19500,000	ext3	/var/
part6	19501,000	20000,000	ext2	/tmp/
part7	20001,000	22000,000	linux-swap	(none)

The ext3 journaled filesystem ensures data safety, while separated partitions for `/tmp/` and `/var/` help limiting global filesystem fragmentation (since these directories are the ones with the most frequent write accesses, e. g. for temporary files and system logs). Furthermore, a big `/var/` partition allows to maintain a quite complete local repository of Debian™ packages downloaded via `apt` (the program automatically stores them under `/var/cache/apt/archives/`).

Just after the installation, the system has been flawlessly upgraded from the “stable” release of Debian™ (“Woody”) to the “unstable” one (“Sid”), that supports more software packages in newer versions, and is solid enough to allow a daily desktop/workstation usage without big problems.

4 Hardware support problems

All the hardware components of the HP workstation zx6000 are supported by GNU/Linux™, with few noticeable exceptions.

4.1 Graphics adapter

The ATI™ Fire GL4™ drivers for GNU/Linux™ and XFree86™ [11] are distributed by the device vendor under a proprietary license, and could only be downloaded in binary format, compiled for the ia32 platform. Furthermore, they require a specific version of the GNU C Library [12] in order to run — which is older than the one that has been installed on the system [13]. It made the ATI™ graphics adapter completely unusable, and we had to replace it with an nVIDIA™ Quadro™ 4 900 XGL board.

The nVIDIA™ policy for drivers release is not different from the ATI™ one: they are available under a proprietary license in binary-only format. Fortunately, they are also compiled for the ia64 platform, and it makes them at least useable on the HP workstation zx6000 — even if they are quite buggy, and the lack of source code makes it impossible to track down some problems (e. g. spurious segfaults and frequent OpenGL visualization errors).

4.2 HP/3Dconnexion™ SpaceBall™ controller

The 3Dconnexion™ SpaceBall™ 4000 FLX is a 6DOF controller that allows easy navigation in a three-dimensional graphical environment (for example, to explore a 3D volumetric dataset or a geometrical model in CAD/CAM applications).

3Dconnexion™ provides proprietary, binary-only drivers for several operating systems. GNU/Linux™ is one of them, but, as usual, the drivers are compiled only for the ia32 platform.

Furthermore, the provided drivers don't follow any of the XFree86™ standards (more in detail, they don't support the XInput extensions [14]), so, even if they worked, the SpaceBall™ device could not be natively used by any graphical toolkit (Gtk+, Qt, GLUT...): it should be necessary to write ad-hoc device management code when developing an application that needs to supports the controller.

Fortunately, it has been possible to find some alternatives to the “official” driver. An unofficial XFree86™ 4.0.1 driver (that follows the XInput extension) has been written by Joe Krahn [15], and even if it doesn't compile cleanly against the XFree86™ 4.1.2 source code (the XFree86™ version that has been installed on the HP workstation zx6000), it should be portable without too much efforts.

We decided not to port the driver (yet) when we found that the 3Dconnexion™ SpaceBall™ controller was directly supported by the Linux™ Input Drivers [16], an hardware abstraction layer for generic controllers (joysticks, mice, etc.) that could be used via the XInput extension. Actually we're trying to make the device work this way (see below in section 5.2).

5 Software problems

Some software problems have been encountered while using the Debian™ GNU/Linux™ system on the HP workstation zx6000, mostly due to applications that are (or were) not 64-bit clean.

5.1 Problems due to 64-bit incompatibilities

ia64 is one of the platforms in which the GNU/Linux™ system (more in detail, the GNU Compiler Collection, gcc [17]) uses a 32-bit wide “int” type, and a 64-bit wide “long int”. There are other 64-bit platforms that use 32-bit wide “int” and “long int” types (for example, the ia64 port of Microsoft™ Windows) [18]. Pointers, however, are always 64-bit wide.

It creates a number of problems to applications that are not 64-bit clean, often characterized by:

- a. assuming that `sizeof(int) == sizeof(long) == sizeof(void*) == 4` bytes;
- b. assuming that a pointer could be safely stored in an “int” (or, on Microsoft™ Windows, even in a “long int”) value;
- c. using `vararg` function stacks without safe parameters casts, to ensure that the number of bytes read by `va_arg()` is the same that was stored during the function call.

Some applications and libraries included in the Debian™ system (and in other variants of the GNU system) suffer of one (or more) of these problems. They are usually limited to small and minor applications (e. g. games and toys), with few exceptions. One of them was the Gtk+ library [19], that suffered of unsafe `vararg` function stack management. However, when the problem was reported to the Debian™ developer that maintains the Gtk+ package, it was fixed in a few hours [20].

5.2 Other problems and bugs

Other bugs have been found in various parts of the system, not strictly related to 64-bit compatibility, but often caused by the young age of the ia64 port of some low-level applications.

One of these bugs seems to be a C compiler problem, that causes a segmentation fault in some GNOME [21] applications and applets compiled for ia64. A debugging session shows that a pointer is mangled (the upper 32 bits are set to zero) when it is returned from one function to another. The issue is being investigated, in order to generate a detailed bug report to be sent to the GCC developers.

Also, the Linux™ compilation stage requires some trickery: the official Linux™ 2.4 tree doesn't support ia64 platforms, and the ia64 kernel source packages provided with the Debian™ system include the patches from the IA-64 Linux™ Project [22]. These patches are not very clean, and enabling some configuration options may cause errors during Linux™ compilation, or even generate broken and unstable kernels. However, the situation is far better with the latest versions of both the kernel and the patches, and actually we aren't experimenting this kind of problems anymore.

Another problem is related to the Linux™ driver for the 3Dconnexion™ SpaceBall™ controller. The device initialization string was wrong (a single character was lowercase instead of uppercase, and the device could not be recognized by the driver), but fixing this problem was a matter of changing a line in the source code of an application (`inputattach`) and recompiling it. However, after that fix, the controller still doesn't seem to be registered in the kernel (no messages in the system logs, and no device is created under `/dev/` via the Device File System [23]). The problem is not ia64-specific (the same happens on the ia32 platform). We will investigate it in the next days, also contacting the Linux™ Input Drivers developers.

Finally, we've got a problem with the XFree86™ “nv” driver, which should support the nVIDIA™ cards (even if without 3D hardware acceleration): it causes the machine to reboot. Maybe its source code is not 64-bit clean — but, since we need 3D hardware acceleration and we're actually using nVIDIA™'s proprietary drivers, we have not investigated the reasons of the problem.

6 Conclusions

The HP workstation zx6000 is actually working without major problems (the ones that have been described, and that have not been resolved yet, are not critical). It has the potential to be an excellent GNU/Linux™ workstation: the performance is very good, and will improve further with the forthcoming adoption of gcc-3.2 as the standard system compiler, since it can generate more optimized code (actually, Debian™ GNU/Linux™ on ia64 is based on gcc-2.96).

Another big improvement should come with the Linux™ 2.5 development tree (that will eventually become the stable 2.6/3.0 Linux™ series), since it has several optimizations and native support for the ia64 platform. Unfortunately, we can't even experiment with these kernels yet, since the nVIDIA™ proprietary drivers require the 2.4 kernel series to work.

The driver releasing policies adopted by some major hardware vendors (like ATI™ and nVIDIA™) are just the problem that slows down the GNU/Linux™ development and support for the ia64 architecture. In fact, their approach makes it impossible to ensure that a new and powerful platform (like ia64) will gain all the capabilities of the pre-existing ones (like ia32): it will be necessary to wait that the vendors decide to port their proprietary drivers, or that someone obtains or reverse-engineers the device specifics, writing a free driver that could be freely recompiled and adapted to every system. Until it happens, people with specific needs may be forced not to adopt a new architecture because of the lack of supported hardware.

References

- [1] The Debian™ Social Contract and Free Software Guidelines: http://www.debian.org/social_contract.
- [2] The Debian™ Policy Manual: <http://www.debian.org/doc/devel-manuals#policy>.
- [3] The APT HOWTO: <http://www.debian.org/doc/user-manuals#apt-howto>.
- [4] The Debian™ GNU operating system ports: <http://www.debian.org/ports/>.
- [5] The Debian™ GNU/Linux™ FAQ: “What about the US export regulation limitations?” (<http://www.debian.org/doc/FAQ/ch-software.en.html#s-useexports>).
- [6] The Debian™ GNU/Linux™ 3.0 Installation Manual: <http://www.debian.org/releases/stable/installmanual>.
- [7] The Linux™ Standard Base homepage: <http://www.linuxbase.org/>.
- [8] The Filesystem Hierarchy Standard homepage: <http://www.pathname.com/fhs/>.
- [9] The Debian™ Library Packaging guide: <http://www.netfort.gr.jp/~dancer/column/libpkg-guide/libpkg-guide.html>.
- [10] A brief guide to the EFI/elilo booting process on ia64: <http://ftp.heanet.ie/mirrors/ftp.redhat.com/pub/redhat/linux/7.1/kr/doc/RH-DOCS/rhl-ig-itanium-en-7.1/sl-guimode-post-installation.html> (even if it is Red Hat™-oriented, it could be applied to every GNU/Linux™ distribution).
- [11] The XFree86™ Project: <http://www.xfree86.org/>.
- [12] The GNU C Library: <http://www.gnu.org/software/glibc/>.
- [13] The download page of the ATI™ Fire GL4™ drivers for GNU/Linux™ and XFree86™: <http://www.ati.com/support/products/workstation/firegl4/linux/firegl4linuxdrivers.html>.
- [14] XInput resources and documentation (for the Gtk+ widget toolkit): <http://www.gtk.org/~otaylor/xinput/>.
- [15] The “unofficial” 3Dconnexion™ SpaceBall™ 4000 FLX driver for XFree86™ 4.0.1, written by Joe Krahn: <http://www.geocities.com/joekrahn/>.
- [16] Linux™ Input Drivers homepage: <http://atrey.karlin.mff.cuni.cz/~vojtech/input/>.

- [17] The GNU Compiler Collection homepage: <http://gcc.gnu.org/>.
- [18] “Porting Linux Software to the IA64 Platform” by Moshe Bar: http://www.byte.com/documents/s=7182/byt1021067742738/0513_moshe.html.
- [19] The Gtk+ library: <http://www.gtk.org/>.
- [20] Bugreport and fix for the Gtk+ Debian™ package, on ia64: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=166442>.
- [21] The GNOME (GNU Network Object Model Environment) homepage: <http://www.gnome.org/>.
- [22] The IA-64 Linux™ project: <http://www.linuxia64.org/>.
- [23] The Device File System for Linux™ : <http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html>.