CONFIGURING MULTIBOOT ENVIRONMENTS ON SUN X64 SYSTEMS WITH AMD OPTERON PROCESSORS

Barton C. Fiske, NSG Technical Marketing

Sun BluePrints™ OnLine
— September, 2005

Part No. 819-2889-10
Revision 1.0, 9/2/05
Edition: September 2005

Please
Recycle

Table of Contents

Configuring Multiboot Environments on Sun x64 Systems with AMD Opteron Processors

This Sun BluePrints™ article gives detailed procedures for configuring Sun x64 workstations with AMD Opteron™ processors to boot more than one operating system from the same physical hard drive. This capability is referred to throughout this article as “multiboot.” Specifically, the three major operating systems in use today — the Solaris™ Operating System, Linux, and Windows operating systems — can be deployed on a single system disk, and configured to allow a user to choose between the different operating systems at boot time. Multiboot capability should not be confused with available virtualization technology that allows *simultaneous* operation of multiple operating systems (such as VMware, Xen, or other approaches).

In an era of increasingly tight IT budgets, many organizations are trying to optimize the utilization of their existing compute resources. To improve utilization, systems are now often shared among multiple users, or systems may be called upon to serve more than one function. Systems that operate by day as user desktops running personal productivity applications may be redeployed as compute nodes at night or on weekends, perhaps taking on complex simulation workloads as a part of a compute grid. Multiboot capabilities contribute directly to increased utilization by giving systems the ability to rapidly assume different roles by booting different operating systems.

In some cases, users are faced with a critical need to access multiple applications and different operating systems on an ongoing basis, such as the software developer who must compile, test, and deploy on multiple operating system platforms (or on multiple versions of the same operating system). Developers often require access to a range of dedicated test systems that are seldom used to their full potential. Target systems used for testing (each with a different OS) might otherwise sit idle while code development occurs primarily on a single machine. Multiboot capabilities let developers use fewer systems while utilizing them more effectively.

Sun’s family of x64 workstations and servers based on the AMD Opteron processor provide the flexibility of supporting multiple operating systems — including the Solaris Operating System for x64, the Sun Java™ Desktop System, the SuSE Linux operating environment, the Red Hat Linux operating environment, and Windows-based operating systems. These systems also offer low-cost, high-performance choices in both workstation and server platforms.

This Sun Blueprints article describes how Sun x64 systems based on the AMD Opteron processor can be configured with more than one operating environment on a single disk spindle, allowing a user to reboot and switch from one operating environment to another. Such a configuration provides tremendous flexibility for users that have requirements for more than one operating system, and can provide a high degree of convenience and flexibility. Moreover, multiboot capability can help to better optimize the utilization of valuable system resources.

In particular, this article describes a multiboot reference implementation for Sun x64 workstations that installs three different operating systems on a single disk spindle. The reference implementation includes the configuration of hardware and software along with system installation and booting techniques so that the reader may develop similar reference and derivative implementations. Additionally, this article presents integration steps and strategies that are known to work for installing and configuring Solaris and other operating environments on Sun x64 systems based on the AMD Opteron processor. An alternative and more complex reference implementation is described in an appendix to illustrate how far these techniques can be taken to support larger numbers of operating systems in multiboot configurations.

Assumptions and Scope

This Sun BluePrints article assumes that readers have a basic understanding of OS installation techniques, including a working familiarity with installing the various operating systems in a single boot configuration. Readers are also assumed to have previously installed (or be capable of installing) the Solaris, Linux, and Windows operating environments on Sun x64 systems in single-boot configurations. An acquaintance with basic boot disk partitioning methodologies is helpful but essential details are covered in the balance of the article.

The article also assumes that readers understand some basic principles of non-SPARC® based computer systems hardware and software practices, including BIOS updates, hardware driver installation, and familiarity with boot loader utilities. While it is not necessary to be deeply familiar with Sun's x64 systems, it is helpful to have a basic understanding of x86/x64-based system concepts as they relate to these topics. Anyone who has installed an x86/x64-based system from scratch should be able to follow the steps and techniques presented in this article, as well as being able to generalize the information presented here to create customized configurations.

Operating Environments Covered by This Document

This document focuses on the Solaris OS, Linux, and Windows-based operating environments. Specifically, the documented procedures use the following distributions:

- Solaris™ 10 OS on the x64 platform with Sun Java Desktop System, version 3, 32/64 bit
- Sun Java Desktop System, Version 3, 32 bit (based on SuSE Linux Enterprise Server)
- Microsoft Windows XP-Pro, 32/64 bit

Note – The more complex reference implementation described in the appendix features additional third-party software and variations of the above operating environments, including operating systems that Sun does not officially support.

Systems Covered by This Document

The procedures and methodologies described in this document are applicable to Sun's entire family of x64 systems based on the AMD Opteron processor (both servers and workstations). Details are given in this document for the following workstations:

- The Sun Ultra™ 20 Workstation
- The Sun Java™ Workstation W1100z
- The Sun Java™ Workstation W2100z

This document discusses Sun systems configured with IDE, SCSI or SATA disk drives, as supported on the respective platforms.

For customers already familiar with the Solaris OS and Sun SPARC based systems, this document will be helpful by highlighting subtle system differences at the hardware level. The discussion also seeks to leverage what users already know about installing the Solaris OS and typical deployment conventions.

For the reader more familiar with x86 32-bit systems running Linux or Windows operating environments, the BIOS and system bootstrap process may be very familiar. The nuances of Sun x64 systems and the procedures for installing and configuring the Solaris OS may be less familiar topics.

Non-Goals

This Sun BluePrints article does not replace or supersede any related documentation on the Solaris OS, Sun systems, and general operating system installation instructions. Customers are encouraged to obtain and read the appropriate installation manuals (see *References* and the Sun documentation Web site, *docs.sun.com*^{SM}). In addition, this document assumes that all disk partitions are on locally-installed disks, and does not cover booting from the network, storage arrays, RAID volumes, removable storage, or similar configurations.

System tuning is not covered in this document in any detail. While multiboot configurations offer significant flexibility and convenience, any performance-minded systems administrator will find the choice of putting the OS, swap and applications all on the same spindle as less than optimal. In this regard, the concepts and solutions herein are presented in as compact a form factor as possible, leaving the issues of performance optimization as an enhancement for the reader to perform. In the reference implementations, system performance parameters for each operating system are left at installation defaults.

In addition, all system security measures and mechanisms are left at installation defaults. Customers may require that additional security measures are implemented in actual deployments. OS patching may also be required to meet application and end user requirements.

How This Article is Organized

This Sun BluePrints article is divided into the following chapters:

- Chapter 1, “Getting Started,” provides an overview of the process used to configure multiple operating systems on a Sun x64 system, introducing the concepts of disk partitioning and describing several optional boot loaders. This chapter also introduces a reference implementations for Sun x64 systems with AMD Opteron processors supporting the ability to boot three different operating systems.
- Chapter 2, “A Basic Multiboot Configuration,” the reference implementation, summarizing the steps used to develop that configuration.
- Chapter 3, “Summary,” provides a brief summary of the article and lists references for further reading and information on related topics.
- Appendix A, “A Multiboot Solution to Support More Than Four Operating Systems,” discusses a second, more complex reference implementation. This configuration gives an example of partitioning that can support many separate operating system instances, and references a third-party partitioning tool (the Ranish Partition Manager) that might be used to achieve such a configuration.

Typographic Conventions

Table 1 shows the typographic conventions used in this article.

Table 1. Typographic conventions

Typeface	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the *User’s Guide*. These are called *class* options. You *must* be superuser to do this.
AaBbCc123	Command-line placeholder text; replace with a real name or value	To delete a file, type `rm` *filename*.

Chapter 1
Getting Started

The process of configuring a Sun x64 system based on the AMD Opteron processor to support multiple operating environments consists of the four basic steps shown in Figure 1.

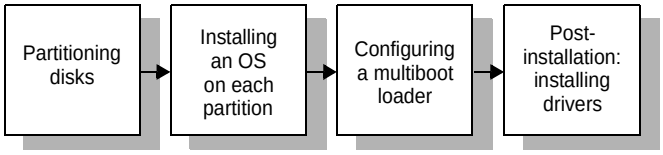

Figure 1. Steps to create a multiboot system

To configure a system capable of booting more than one operating system, the system disk must first be divided into multiple partitions that are subsequently installed with various operating systems. A boot loader program is then configured to allow the user to choose which operating system should boot. After operating systems are installed and the boot loader is functioning correctly, post-installation tasks are completed — such as installing and configuring device drivers.

Configuration Planning

For a successful multiboot implementation, initial configuration planning is a particularly important step. Planning the layout of partitions is a key requirement that must be completed prior to actual partitioning of the disk. The layout of disk partitions is impacted by important considerations, including the size and number of physical disks, the operating environments needed, and sizing estimates for applications and file storage. Taking the time to define the appropriate partition maps beforehand may prevent the predicament that the author encountered — namely, starting over from scratch and losing several hours worth of effort.

To build a usable configuration, it is best to think through many issues in advance and to try to understand the dependencies and areas of potential conflict. This document is designed to help with the planning process while imparting real world experiences encountered during actual multiboot installations.

To get started, this chapter first presents some useful background context and key concepts. Disk configurations for the Sun x64 AMD Opteron based systems are described along with some considerations for making partition layout decisions. An overview of the boot process is also provided, along with a summary of how partitions are described internally in the Master Boot Record (MBR). The chapter also includes a brief synopsis of available partitioning tools and boot loader technologies.

Disk Configurations

To plan the layout of disk partitions, it is often useful to know the disk geometry, including the number of cylinders on each disk. The Sun x64 workstations based on the AMD Opteron processor use the disk configurations listed in Table 2. The reference implementation described in this document use a single

physical hard disk although all Sun platforms listed can support up to two internal disks (of the same geometry) in each system chassis, and some systems can support larger numbers of disks.

Table 2. Basic disk configuration information

Workstation or Server	Disk Configuration
Sun Ultra 20 Workstation	80GB or 250GB SATA Hard Drive (up to two)
Sun Java Workstation W1100z	80GB Parallel ATA (IDE) Hard Drive (up to two)
Sun Java Workstation W2100z	73GB Ultra320 SCSI Hard Drive (up to four), or 146GB Ultra320 SCSI Hard Drive (up to four)

Disk Device Naming

Sun x64-based systems based on the AMD Opteron processor may use either SATA, IDE, or SCSI disks, depending on the system configuration. The Sun Ultra 20 Workstation features SATA disks and the Sun Java Workstation W1100z system uses IDE disks, while other Sun x64 systems use SCSI disk drives. During operating system installation, disk naming will vary depending on the operating system and the specific disk drives used. It is helpful to be familiar with these naming conventions when the time comes to select the appropriate drives and allocate the requisite partitions.

For Windows-based operating systems, device names use a drive letter format, such as `C:\`, `D\:`, `E:\`, etc., through `ZZ:\`. For Linux operating systems, IDE disks use device names `/dev/hda`, `/dev/hdb`, etc., while SCSI disks use `/dev/sda`, `/dev/sdb`, etc. For SATA drives in an Ultra 20 Workstation, RedHat Linux sees the drives as `/dev/sda`, `/dev/sdb`, etc., while SuSE Linux sees the drives as `/dev/hdg` and `/dev/hde`.

Partition names are as follows:

- `hda`<*1-n*> represents partitions 1-n for the first IDE or SATA disk, `hdb`<*1-n*> represents partitions 1-n for the second disk, etc.
- `sda`<*1-n*> represents partitions 1-n for the first SCSI or SATA disk, `sdb`<*1-n*> represents partitions 1-n for the second disk, etc.

For the Solaris OS on x64 or x86 systems, device names vary depending on whether the device represents a system disk *partition* (as defined in the MBR) or whether it refers to a Solaris *slice* (which is a part of the partition containing the Solaris OS and related file systems, such as `/`, `/opt`, etc.). For the Solaris OS on the x64 platform, device boot disk names use the form `/dev/dsk/c`*n*`t`*n*`d`*n*`s`*n*, where ***x*** corresponds to either ***p*** for partition or ***s*** for slice, and *n* corresponds to the number of the controller, target, disk number, and partition or slice, respectively. (See the *Solaris OS System Administration Guide* for more information on disk device naming.)

About Disk Partitioning

Caution – Prior to partitioning a disk, any data on the system should be backed up. Partitioning the disk will destroy any existing data and operating system information. If possible, use a blank disk or one that can be reformatted without concern for loss of data.

For the purposes of this Sun BluePrints article, a disk partition is defined as a contiguous portion of disk storage that may contain a file system or a logical volume.[1] Partitions may be primary partitions — those which span contiguous space and do not extend beyond that range of space — or extended partitions, which link multiple logical volumes together (the first partition holds the address of the second and subsequent volumes in the fashion of a singly-linked list). Extended partitions evolved from the need to support disk sizes greater than 2GB while providing compatibility with early DOS- and Windows-based systems. An extended partition can be divided into a maximum of 63 logical volumes.

An x64 boot disk can have up to four primary partitions, or three primary partitions and an extended partition (for reference, see `http://www.ata-atapi.com/hiwtab.htm`). The reference implementation in this Sun BluePrints article uses three primary partitions and a single extended partition to illustrate how to implement basic multiboot capability (Figure 2). An 80GB IDE disk was used for this reference implementation.

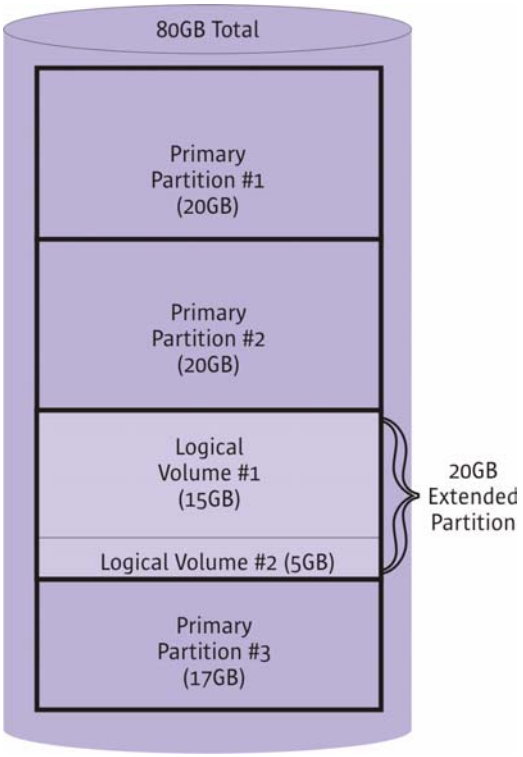

Figure 2. Three primary partitions and a single extended partition on an 80GB IDE disk

1.A logical volume refers to contiguous disk space, which together with other logical volumes may make up an extended partition

A second unsupported reference implementation (described in Appendix A, “A Multiboot Solution to Support More Than Four Operating Systems”), uses a partitioning tool that can overcome the four primary partition limitation without the use of extended partitioning. In the second reference implementation, the partitioning tool is employed to define twelve separate virtual partitions.

Partitioning and Operating System Layout

In a multiboot single-disk configuration, before partitioning the disk, it is necessary to consider the eventual layout of the operating systems. It is critical to carefully plan the partitioning and layout of the physical disks, taking requirements into consideration that extend beyond the disks to each individual operating system to be installed.

Some important operating system layout considerations include the following:

- In general, Windows operating systems can be installed on either a primary partition or a logical volume of an extended partition. In a multiboot environment, if a Windows-based operating system is to be installed, it is generally installed on the first primary partition so that the OS will treat that partition as the `C:\` drive. Although placement on the first partition is not an absolute requirement, some legacy Windows applications may require that the operating system is installed on the `C:\` drive or they may assume that it is installed there. In any case, installing Windows on the `C:\` drive is the safest route to pursue.
- The Solaris OS *must* be installed on a primary partition. This requirement exists because it is not possible for a UFS file system to reside on a logical volume of an extended partition. Before beginning the Solaris installation, it is also critical to understand that all of the UFS file systems required for Solaris (root (`/`), `/opt`, `/var`, `/export/home`, etc., and even swap space) are allocated on *slices* within the Solaris primary partition during installation[1]. This primary partition acts as the “disk” where the Solaris OS runs. (Refer to Figure 5 for a graphical depiction of this crucial insight.).
- For Linux operating environments, either primary partitions or logical volumes of an extended partition may be used for installation. There must be at least two partitions or logical volumes available — one for the Linux root (`/`) file system and one for Linux swap. Some boot loaders (notably LILO, briefly described later in this chapter) require root (`/`) and `/boot` to exist within the first 1024 cylinders.
- Swap space can be shared between two (or more) Linux operating system instances, but not between a Linux operating system instance and the Solaris OS, nor between a Linux operating system instance and a Windows-based operating system instance. In all cases, the general rule of thumb is to size swap space to be approximately twice the size of physical memory, and certainly never less than the amount of memory installed on the target system.

1.While performance tuning is outside the scope of this document, it is important to note that co-locating the Solaris OS, swap, and applications on the same disk spindle is not generally recommended for the best performance.

Reference Implementation

The reference implementation described herein supports three operating systems on a single 80GB IDE disk. Table 3 shows the proposed disk partitioning and suggested operating system layout, described in more detail in Chapter 2. The multiboot reference implementation uses a partitioning scheme with three primary partitions and one extended partition that has two logical volumes (refer back to Figure 2),

Table 3. Partition information for the reference implementation on an 80GB IDE disk

Partition #	Type	Intended OS	Partition Information
1	Primary	Microsoft Windows XP-Pro, 32/64 bit	Size: 20GB FS Type: NTFS
2	Primary	Solaris 10 OS (x64) 32/64 bit with Sun Java Desktop System, Vers. 3, 32/64 bit	Size: 20GB FS Type: UFS
3	Extended	Sun Java Desktop System, Vers. 3, 32 bit 1st logical volume - Linux (15GB) 2nd logical volume - Linux swap (5GB)	Size: 20GB total FS Type: EXT-3
4	Primary	Common data partition	Size: 17GB FS Type: FAT32

Common Data Partition

The reference implementation features a common data partition that can be used to share files and application install images between different operating environments. The common data partition is created as a FAT32 file system because the FAT32 file system is easily and reliably mountable between Windows, Linux, and the Solaris operating systems. (In contrast, a UFS file system can be mounted and read by most Linux operating systems, but may not prove stable for write access. Additional third-party software is required to mount Solaris UFS filesystems from Windows.)

The shared data partition must be individually mounted under each operating system — for Linux and the Solaris OS, mount point entries must be made after operating system installation in `/etc/fstab` and `/etc/vfstab`, respectively. Examples of mount point entries are given in subsequent sections of this document. For Windows, the FAT32 file system is mounted using drive letter semantics, and should be visible at Windows startup/login as `D:\`, `E:\`, `F:\`, `G:\`, etc.). For the Windows 2000 or XP operating systems, FAT32 partitions are limited in size to a maximum of 32GB and the maximum file size in a that partition is limited to 2GB.

Note that FAT32 file systems are subject to file naming restrictions, and may have issues with long filenames or filenames that contain spaces or other special characters. A FAT32 file system only recognizes filenames that adhere to “8.3” naming conventions (“8.3” naming refers to the first eight characters being used to represent the filename, while the 9th through 12th characters are the file type extension). Other file naming conventions may be supported by some operating environments on other file systems, but these filenames may not be recognizable on a FAT32 file system, and may be converted to a variant of the name (e.g., “Program Files” may become “PROGRA~1”). Also, remember that FAT32 file systems are not journaled, and as such, may be more susceptible to data errors.

Partitioning Tools

To configure a system capable of booting more than one operating system, the system disk must first be divided into multiple partitions that are subsequently installed with various operating systems. A boot loader program is then used to choose which operating system should boot. The disk partitioning utility provided with early x86-based PCs based on DOS was FDISK, which originally did not allow more than one primary partition. Over time, that limit evolved to four primary partitions, corresponding to the four partition descriptors in the Master Boot Record or MBR. Other partitioning tools have since evolved beyond FDISK, providing greater flexibility and ease-of-use with regards to partitioning. In addition to supporting extended partitioning and the definition of logical volumes, many new command line and graphical partitioning tools have emerged to work around the limitation of four primary partitions. Some offer ease-of-use features such as a graphical user interface, and convenient features, such as simplified ways to grow and shrink partition sizes.

Partitioning tools may also ship as part of an operating environment, such as `format` in the Solaris OS or the Windows Disk Management Tool in Windows-based operating environments. Other commonly available partitioning tools include Symantec Norton Partition Magic and the Ranish Partition Manager (see example in Appendix A).

To duplicate the reference implementation described in this Sun BluePrints article, any partitioning tool can be used as long as it can define three primary partitions and a single extended partition on the disk. For example, the Windows XP Disk Management tool can be used to partition the system disk. The process is discussed in Chapter 2, “A Basic Multiboot Configuration.”

Partitions and the Master Boot Record (MBR)

The first physical sector on a system disk contains the Master Boot Record (MBR), which holds both the partition table and a small amount of executable code that is the Initial Program Loader (IPL), or boot loader program. The partition table and the boot loader in the MBR are critical to the boot process and to the capability of booting multiple operating systems. As shown in Figure 3, the Master Boot Record is located in the first block (512 bytes) of the disk and contains the 446-byte initial program loader, a 2-byte header, and four 16-byte descriptors for each of the four possible disk partitions.

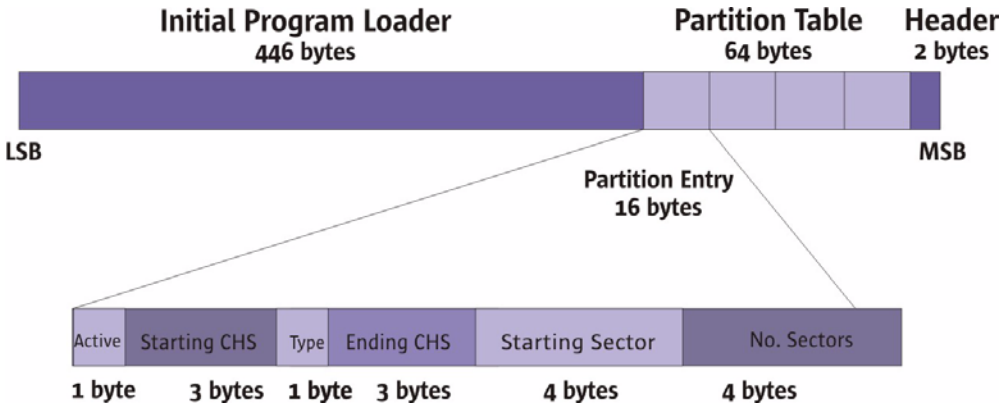

Figure 3. The Master Boot Record (MBR) and a Partition Descriptor.

The 16-byte partition descriptor defines whether the partition is “active” (that is, whether it is the active boot partition), its starting and ending CHS (Cylinder/Head/Sector) values, its starting sector, and its size (in number of sectors). In addition, a 1-byte “type” (or “partition ID”) field is used to identify the partition type. Note that only one of the four partition entries can be marked as the active boot partition at any one point in time (using the “active” field in the partition descriptor). Once the active field is set, this is the partition from which the system will attempt to boot.

By reading the MBR “type” field, some partitioning tools and boot loaders (discussed below) are able to recognize the underlying file system type. In many cases, the boot loader can then make a reasonable assumption about the operating system installed on that file system. Some common hex values and their corresponding file system types include:

- (0x07) Windows NT (NTFS)
- (0x0C) FAT32 (DOS/Windows)
- (0x82) UFS for Solaris x86 or Linux swap
- (0x83) Linux native (xiafs, ext2, ext3, reiserfs)
- (0xBF) UFS for Solaris 10 OS (and releases beyond Solaris 10)[1]

For a more complete list of partition type values, see `http://www.win.tue.nl/~aeb/partitions/partition_types-1.html`. Some boot loaders and partitioning tools (such as the Windows Disk Management tool and the Ranish Partition Manager) are not able to recognize the Solaris UFS file system, and will list its file system type as “unknown.”

The Boot Process

Generally, a Sun x64 system uses the following boot sequence when it is first powered on:

1. The BIOS (Basic Input Output System) in system ROM is read and executed, usually initiating a system Power On Self Test (POST).
2. The BIOS determines the “boot device,” which is normally a hard disk, but could also be the network, a CDROM, or USB device. In Sun x64 systems based on the AMD Opteron processor, the BIOS provides similar functionality to the Forth-language based Open Boot PROM in Sun SPARC based systems. Both mechanisms initiate POST diagnostic tests, and both define the specific boot disk device in a multiple disk system.
3. The BIOS invokes an interrupt (Int 19h) to load the contents of the first physical sector of the hard disk (the MBR) into executable memory (RAM).
4. The BIOS then passes control to the MBR boot code by issuing a jump to that location. The CPU then executes the primary boot loader code from the MBR. The MBR designates the active boot partition, and instructs the CPU to continue execution from the start of a specific file system boot sector, located at the starting cylinder, head, and sector for that partition.

1.The Solaris type identifier (OxBF) is recently defined. Previously, the Solaris OS for x86 was usually identified using the Ox82 identifier, but that could create issues during installation of other operating systems since it was also used for Linux swap

About Boot Loaders

To configure a system that can boot more than one operating system, the system disk must first be partitioned and subsequently installed with the appropriate operating systems. A boot loader program can then be configured to allow a user to choose between the different installed operating systems. The boot process generally consists of two separate phases, a primary boot phase and a secondary boot phase. The primary boot phase is implemented in system BIOS, as discussed above. The secondary boot phase occurs when a boot loader program is read into memory from the MBR and then executed.

There are several boot loader programs available that can support the selection of a boot partition and can initiate the boot sequence of the selected operating system. Some boot loader programs are packaged with an operating system, while others are available as shareware or as a commercial product. Regardless of whether a boot loader is delivered as part of an operating environment, it can often be used to select between multiple operating systems, including those from other vendors.

A sample list of boot loader programs follows:

- **The Solaris boot loader.** The Solaris master boot program, `mboot`, is a primary boot loader program that is included with the Solaris OS. During the installation of the Solaris OS, `mboot` is installed in the MBR. On subsequent boots, `mboot` reads the current disk partition information and displays a table with available partitions, allowing the user to boot from a particular partition. Like other MBR boot code, `mboot` reads the first sector of the boot disk partition, and then causes a jump to the first byte of that partition. If the boot partition contains the Solaris OS, the first sector of that partition contains the Solaris boot program (`pboot`) and the Solaris `bootblk` program. As a result of the primary boot process, the Solaris boot program is then loaded (either through `bootblk` if booting from a hard disk or `strap.com` if booting from a CDROM).
- **Grand Unified Bootloader (GRUB).** GRUB is an open-source boot loader available from GNU (see `http://www.gnu.org/software/grub/`). Through a command line interface (CLI), GRUB enables the booting of multiple operating systems, and attempts to standardize boot environments. A configuration file (`/boot/grub/menu.lst`) is used to define and facilitate booting from various partitions. During the installation of the Sun Java Desktop System, GRUB can be optionally installed in the MBR or on another partition. It is important to note that GRUB begins counting partitions starting at 0. For example, in the GRUB command line, partition (hd0,1) refers to the second partition on the first disk, corresponding to the Linux device name `/dev/hda2`.
- **The Linux Loader (LILO).** LILO allows a user to select a specific boot partition, and it can boot Linux as well as other operating systems. LILO can be installed into the MBR, or into a Volume Boot Sector (VBS) on a primary partition (or onto a logical volume of an extended partition). When LILO is installed into the MBR, it controls the boot process of the machine. LILO can be customized and configured during the initial Linux installation. By default, LILO knows nothing about other installed operating systems, and will only boot Linux. However, once configured, LILO will boot Linux as well as other operating systems. The LILO distribution is available at `http://lilo.go.dyndns.org/`.

- **The Windows boot loader.** Windows operating systems may be booted only from a primary partition or from a floppy disk or recovery CDROM. Installation of a Windows-based OS does not modify master boot code stored in the MBR of the boot disk, which uses a standard BIOS method for loading the Volume Boot Sector. First, the NT loader program, `NTLDR.EXE`, is loaded from the boot sector of the hard drive. Next, the `BOOT.INI` file is opened and used to present a menu of operating systems that can be booted. (See `http://www.comptechdoc.org/os/windows/ntwsguide/ntwsbooting.html`.)
- **OS-BS FreeBSD boot manager.** This boot manager is free, and can be downloaded from `ftp://ftp.freebsd.org/pub/FreeBSD/tools`. The main advantage of OS-BS is its small size, allowing it to fit into the MBR without requiring an additional partition. Like other boot loaders, OS-BS can select a different boot partition from a menu. During the configuration process, the user can assign labels for each partition and operating system.
- **The Ranish Partition Manager.** Partitioning and boot management capabilities are both included with the Ranish Partition Manager (`http://www.ranish.com/part/`). Two boot manager versions are included: a small manager that fits into the MBR, and an advanced manager that requires a small partition. Both boot managers allow different operating systems to be booted from multiple partitions. The Ranish boot manager dynamically remaps partition descriptors in the MBR at boot time, creating a “virtual” partition table and allowing the boot manager to select from more than four partitions. In the “virtual” partition table, the boot manager also designates one of the four selected partitions as the active boot partition. More details about the Ranish boot manager and partitioning tool are included in Appendix A.

Chapter 2
A Basic Multiboot Configuration

This chapter describes a basic multiboot configuration. In this reference implementation, a Sun x64 system based on the AMD Opteron processor is configured to boot one of three operating systems. The system disk is first divided into three primary partitions and one extended partition, that will subsequently house with the Windows XP operating system, the Solaris 10 Operating System, and the Sun Java Desktop System Version 3, respectively. A common data area is also created to enable data sharing between the three operating systems. Table 4 shows the partitioning layout and the device names for the three different operating environments.

Table 4. Partition information for the reference implementation

Partition #: Intended OS	**Size (MB)**	**Windows Partition Names**	**Linux Partition Names**	**Solaris Partition Names**
1: Microsoft Windows XP	20480MB	C:\	/dev/hda1	/dev/dsk/c0t0d0p0
2: Solaris 10 OS (x64)	20480MB	(n/a)	/dev/hda2	/dev/dsk/c0t0d0p1
3: JDS Linux OS Linux swap	20480MB total logical vol. #1 = 15360MB logical vol. #2 = 5114MB	(n/a)	/dev/hda3 (extended) /dev/hda5 (logical vol. #1) /dev/hds6 (logical vol. #2)	/dev/dsk/c0t0d0p3 (extended)
4: Common data partition	remaining disk space, ~17GB	D:	/dev/hda4	/dev/dsk/c0t0d0p4

The layout of the operating systems on the disk is important. Windows is installed on a primary partition and in general, Windows is installed onto the disk first so that its partition is recognized as the `C:\` drive. Another reason for installing Windows first is that it overwrites the MBR, and Solaris and JDS will install boot loaders in the MBR during their respective installations. The Solaris OS requires a primary partition for installation. The Linux operating environment is installed on an extended partition that has two logical volumes, with one logical volume designated for the operating system and the other for swap. The last primary partition is used for the common data partition. Additional disks may also be partitioned, although booting is supported on only one disk at a time.

Partitioning and OS Installation

Caution – Prior to this procedure, any data on the system should be backed up. Partitioning a disk destroys existing data and operating system information. If there is data that must be preserved, it may be best to remove the existing disk and start with a known blank drive.

Before You Begin

Before beginning, gather the CDROM images needed for installation. (In this reference implementation, the operating systems were all installed from CDROM images. Network installation should also work, but is beyond the scope of this document.) The following operating system images were used:

- Microsoft Windows XP-Pro, 32/64 bit
- The Solaris 10 OS for the x64 platform, 32/64 bit
- Sun Java Desktop System, Version 3, 32 bit

No matter the what system is involved, the system BIOS must be up-to-date before proceeding. This information can be determined by going to the relevant product specific web pages for either the Sun Ultra 20 workstation, or the Sun Java Workstations 1100z or 2100z at http://www.sun.com

Partitioning and Installation of Windows XP

In a multiboot environment, a Windows-based operating system is generally installed on the first primary partition so that the OS will treat that partition as the `C:\` drive. Although this is not an absolute requirement, some legacy Windows applications may assume that the operating system is installed on the `C:\` drive. The steps for partitioning a disk with Windows tools follow:

1. To begin, boot the system using a Windows XP bootable CDROM. In Setup mode, use FDISK to create new disk partitions, dividing the disk into the three primary partitions and one extended partition, as listed previously in Table 4.
2. Define two logical volumes in the extended partitions as shown. Assign drive `C:\` to the first primary partition, but do not assign drive letters to the other partitions or logical volumes. Format the first primary partition as an NTFS file system and the fourth primary partition as a FAT32 file system for the common data partition. (Other partitions will be formatted as the other operating environments are installed.)

   These partitions are generously sized to accommodate the various operating environments. Minimal requirements would be approximately:

   – Microsoft Windows XP — 6GB
   – Solaris 10 OS — 6GB (full OS installation takes about 3GB, plus swap, etc.)
   – Sun Java Desktop System — 6GB (5GB for Linux root, 1GB for swap)
3. After the initial partitioning is completed, install the Windows XP operating system on the first partition. When the installation of the Windows XP operating system is complete, reboot the system.
4. After the system has successfully rebooted, execute the Windows Disk Management utility (under Control Panel/Administrative Tools/Computer Management). This tool displays a graphical representation of the partition table, depicting the three primary partitions, the extended partition, and the two logical volumes (Figure 4). Confirm that the partitions shown match the desired partitioning sizes, that the C:\ partition uses the NTFS file system, and that the last primary partition (D:) uses the FAT32 file system. At this point, the initial steps of partitioning, and the installation of the Windows XP operating system, are complete.

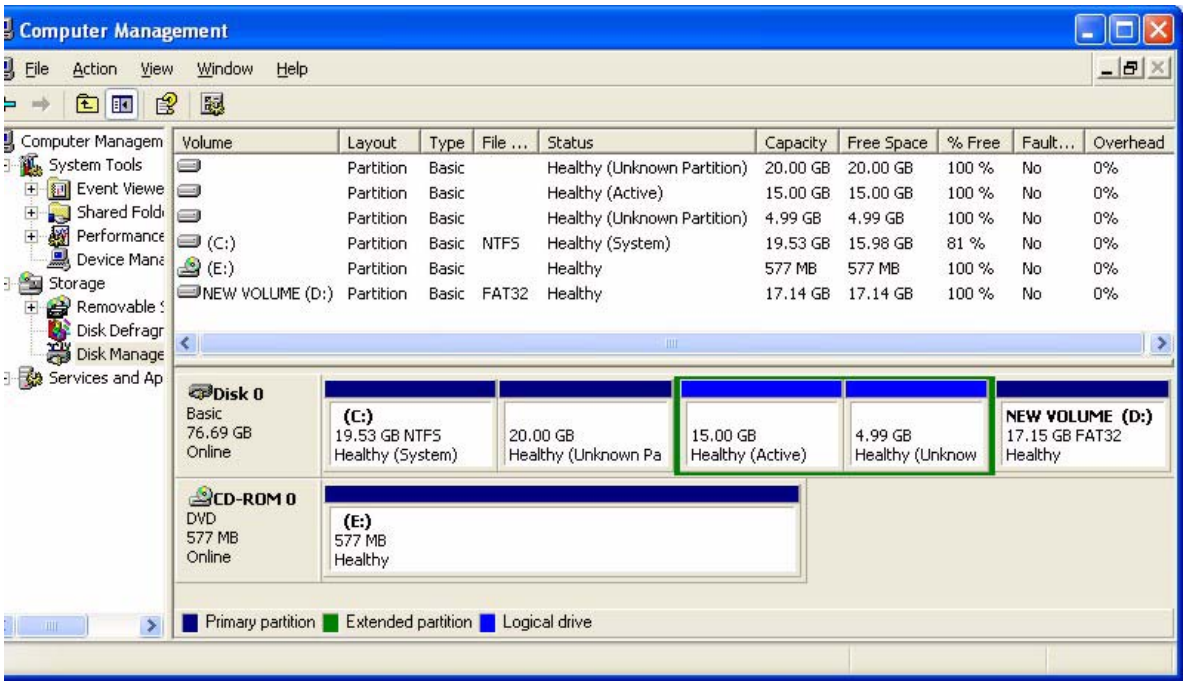

Figure 4. Configured partitions as shown in Windows Disk Management

Once Windows XP has been successfully installed onto the first primary partition, the other operating systems can be installed into the remaining partitions. Steps for installing the Solaris 10 OS for x86/x64 platforms and the Java Desktop System (JDS) are described in the following sections.

Installation of Solaris 10 OS on x64 Platforms

The reference implementation places the Solaris 10 OS on the second primary partition. Follow these steps to install the Solaris 10 OS:

1. Shutdown or restart the system, re-booting with the Solaris 10 OS CDROM #1 or DVD. Proceed with an interactive installation of the Solaris 10 OS.
2. Select the `c0d0` disk device as the target for the Solaris OS software, and select the menu to edit or customize fdisk partitions. To model the reference implementation, select the second partition as the installation target for the Solaris 10 OS. Delete the partition first, and then recreate it as a Solaris partition, which formats it as a UFS file system. Deleting the partition first allows Solaris to see the partition as available for installation.
3. Select manual layout of file systems and define the size and mount point for each slice. (A “slice” is the Solaris name for a logical section of a partition.) The “overlap” slice refers to the entire partition for the Solaris OS installation, and the size should match what was previously allocated for the partition. Define the root (`/`) slice as `c0d0s0`, a swap slice, and, if desired, an `/export/home` slice (Figure 5). In the reference implementation, approximate sizes are 3896MB for root (`/`), 514MB for swap, and 16059MB for `/export/home`. Although these sizes are individually specified for each mount point, all of the slices are created *within* the primary partition that has been allocated for the Solaris OS.

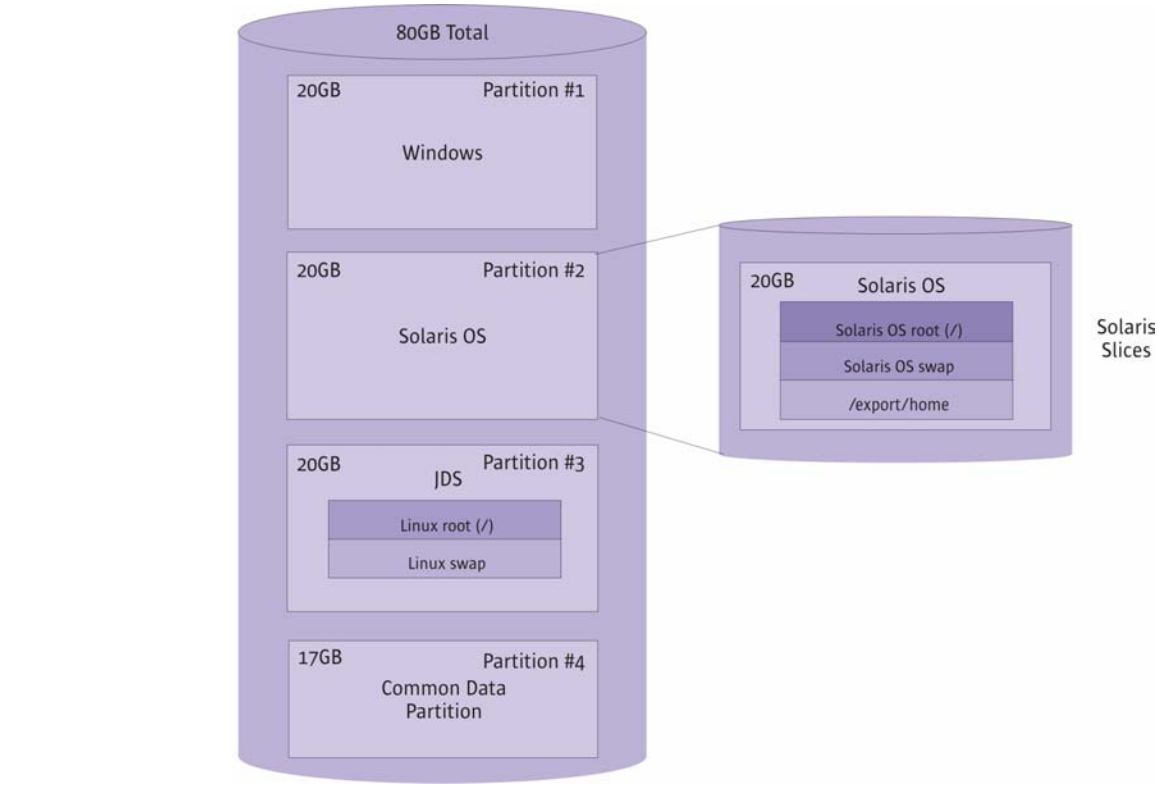

Figure 5. System Disk Layout and Solaris Slices on a Single Disk Partition.

4. Complete the Solaris installation using the other required CDROMs (or DVDs) and reboot. Upon reboot, note that the Solaris boot loader code now supersedes what had previously been installed in the MBR and allows the user to select the desired boot partition. (This is one reason why Windows is installed first in this reference implementation, followed by the Solaris OS.) Because of the installation of the Solaris boot loader, it is now easy to switch between Windows XP and the Solaris OS during a system reboot. If the boot partition is not changed within 10 seconds, the most recently used boot partition is rebooted.

Installation of JDS Version 3

To complete the operating system installation for the reference implementation, JDS version 3 is installed into the third partition. Steps for this process follow:

1. Shutdown or restart the system, and reboot with CDROM #1 of the Sun Java Desktop System Version 3. The installation begins with a “Welcome” screen in a variety of languages, followed by an installation screen that lists various installation options.
2. Select “Installation” from the menu. Press F2 and set the video output resolution from its current setting (likely 1280x1024) to Text Mode. Text Mode avoids potential video compatibility issues between the system graphics card and the monitor, and is the setting that is used for a server console. At this point, the Linux YaST installation program begins. (During post-installation procedures described later in this

chapter, the current NVIDIA graphics drivers can be installed, which then allows Graphics Mode to work properly for YaST installations.)

3. Follow the YaST prompts to select the installation language and other installation options. An “Installation Settings” screen summarizes installation parameters, allowing the settings to be modified prior to the actual installation. As shown in this summary, YaST has made some assumptions about partitioning based on the MBR partition descriptors previously installed by other operating systems. For the reference implementation, YaST displays the following partitioning proposal:

```
*   Format partition /dev/hda5 14.9GB (for /)
*   Format partition /dev/hda6 4.9GB (for swap)
*   Set mount point of /dev/hda1 to /windows/C
*   Set mount point of /dev/hda4 to /windows/D
```

   Be sure that the assumptions match the partitions where JDS will be installed. In this reference implementation, JDS is installed on two logical volumes in the extended partition, corresponding to /dev/hda5 and /dev/hda6. (The device /dev/hda3 represents the extended partition itself.)

4. Use the Tab key to highlight “Change” to modify the partitioning proposal and select “Partitioning” from the pull-down menu. Select the option to base partitioning on YaST’s proposed partitioning scheme (rather than re-partitioning).

```
(   ) Accept base proposal
( x ) Base partition setup on this proposal
(   ) Create custom partition setup
```

   Make sure that YaST accurately reflects the partitioning previously defined.

5. Create `/dev/hda5` as a partition, formatting it as *ext2*, and setting its mount point as the Linux root (`/`) partition. Define `/dev/hda6` as the Linux swap partition. Delete any mount point for `/dev/hda1` (the Windows XP partition), and set the mount point for `/dev/hda4` as `/common`. When the partition information looks correct, select “Accept” to save the changes. The installation settings will now appear as follows:

```
*   Format partition /dev/hda5 14.9GB (for /)
*   Format partition /dev/hda6 4.9GB (for swap)
*   Set mount point of /dev/hda1
*   Set mount point of /dev/hda4 to /common
```

6. At the “Installation Settings” screen, use the Tab key to highlight “Booting.” Choose the selection to write the GRUB boot loader into the MBR:.

```
( x ) Write GRUB to the boot disk (‘MBR’ on /dev/hda)l
(   ) Do not use GRUB (a different boot manager is required)
(   ) Write GRUB to a different partition
```

7. Complete the installation of the Sun Java Desktop System using the other CDROMs. The system should reboot when the installation is complete.

8. At the JDS login screen, log in as root and enter the appropriate password. Make a backup copy of the GRUB configuration file `/boot/grub/menu.lst` first before editing it. In the original, add in the highlighted lines so that the Solaris boot partition will be included as a boot option. (For more information about GRUB syntax, type `help` at the `grub` command line or see `http://www.gnu.org/software/grub/manual/`.)

```
gfxmenu (hd0,4)/boot/message
color white/blue black/light-gray
default 1
timeout 8
title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda5 hdc=ide-scsi
    initrd (hd0,4)/boot/initrd
title windows 1
    root (hd0,0)
    makeactive
    chainloader +1
title Solaris 1
    root (hd0,1)
    makeactive
    chainloader +1
    boot
title failsafe
    kernel (hd0,4)/boot/vmlinuz.shipped root=/dev/hda5
ide=nodma apm=off acpi=off vga=mormal nosmp noapic
maxcpus=0 3
    initrd (hd0,4)/boot/initrd.shipped
```

After a reboot, the user can select a bootable OS from the GRUB menu, choosing either Windows XP, the Solaris 10 OS, or the Sun Java Desktop System (Figure 6). The menu is displayed for a period up to that given by the `timeout` field in the GRUB menu file. If the timeout is reached without direction from the keyboard, then a default boot OS is chosen based on the `default` field in the menu file. In the example given above, after a timeout of 8 seconds, the first title in the list (namely Linux) will be selected. To change the default behaviour, entries can be moved around, or the default field can be changed to point to the desired entry. For more information on GRUB, please see `http://www.gnu.org/software/grub/`.

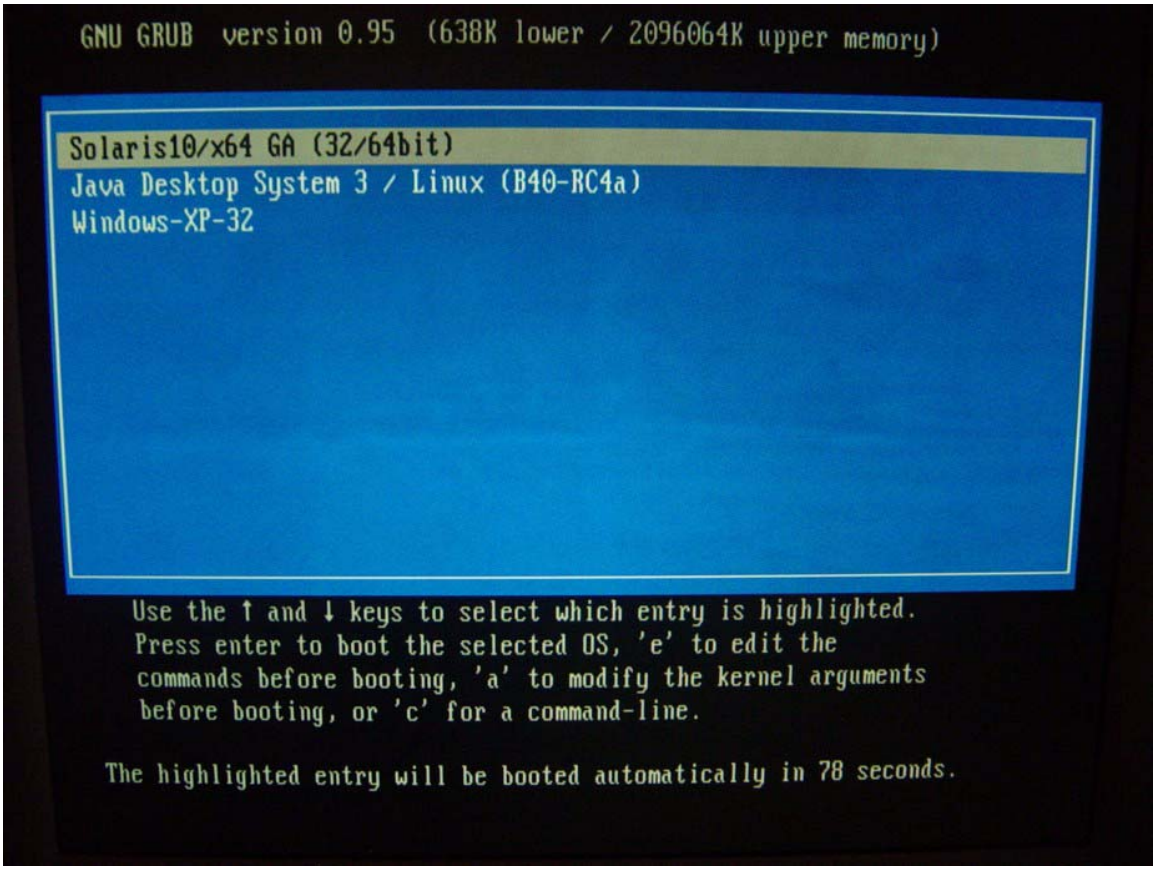

Figure 6. After installation of the reference implementation, the user can boot from any of the installed operating systems from the GRUB menu

Post-Installation

Once the individual operating systems are installed on their respective partitions, individual post-installation procedures are executed to tune and configure each operating system. Post-installation steps include:

- Installing and configuring drivers for each operating system
- The generation of a configuration file for use with the Xorg server (for the Solaris 10 OS)
- Mounting and configuring the common data partition.

To perform many post-installation steps, the Supplemental CDs for the Sun Ultra 20 workstation and Sun Java Workstations W1100z and 2100z can be quite useful. The Supplemental CDs ship with each system and contain recently released drivers for all the operating systems detailed in this document. As noted previously, there is a BIOS update delivered on the Supplemental CD for each platform described. The ISO image for these Supplemental CDs is also available on each product's Web page at `http://www.sun.com`.

Note that if software is used to burn the supplemental CD image, it must be capable of burning a bootable CD image, as well as creating a data partition. The resulting CD should be bootable into OpenDOS for the purposes of running either PC-Check, BIOS Upgrade, or escape to DOS. Alternatively, the CD image can also be mounted from the Solaris OS, Linux, or Windows operating systems. As a data CDROM, the Supplemental CD contains drivers and scripts for use with the workstations.

Installing Drivers

After installation of the base OS, device drivers can be installed for the individual operating systems

- **Drivers for Windows-Based Operating Systems**

  For Windows operating systems, the latest Windows drivers are available on the Sun Ultra 20 workstation Web site, the Sun Java Workstation Web site, or on the Supplemental CD. This CD also contains the "`XpReburn`" script, which can be used to create a Windows XP 32-bit recovery CD with the drivers included on the CD, or using updated drivers available on the system Web sites. To locate the appropriate Supplemental CD image, see `http://www.sun.com` and use the search facility to locate Web pages for the product of interest. NVIDIA graphics and video drivers can be installed from the Supplemental CD, or downloaded from the NVIDIA web site (`http://www.nvidia.com/`). (Be sure to select the driver for the NVIDIA Quadro board and the appropriate Windows operating system.)

- **Drivers for the Solaris 10 OS**

  The Solaris 10 OS requires the installation of video drivers and the creation of an X11 configuration file if the Xorg server is selected. (The configuration of the Xorg server is covered later in this chapter.) To achieve the optimal graphics performance running OpenGL applications on the Solaris OS, customers will also want to install the latest NVIDIA 32-bit and 64-bit 3D OpenGL drivers. NVIDIA graphics and video drivers can be installed from the Supplemental CD, or downloaded from the NVIDIA web site (`http://www.nvidia.com/`). (Be sure to select the driver for the NVIDIA Quadro board and the appropriate version of the Solaris OS.)

- **Drivers for the Solaris 9 x86 OS**

  Most releases of the Solaris 9 OS for x86 platforms pre-date the Sun x64 systems based on the AMD Opteron processor. Nonetheless, the Sun Java Workstations W1100z and W2100z have been verified and tested to run the Solaris 9 OS and are supported. If the Solaris 9 OS is required for x86 platforms, be sure to reference the relevant Web pages for this OS to determine what drivers, updates, or patches may be needed. Note that for all systems, if you plan to use the display, keyboard, and monitor, it is suggested you use only the "VESA Std" video driver that ships with the Solaris 9 OS.

- **Drivers for the Sun Java Desktop System Version 3 (Linux) and RedHat Linux**

  For the Sun Java Desktop System Version 3 or Red Hat Linux operating environments, the only drivers that are immediately required for the Sun Ultra 20 workstation, or for the Sun Java Workstations W1100z and W2100z, are for the NVIDIA graphics card, and either the AMD 8000 chipset (Sun Java Workstations W1100z and W2100z) or the NVIDIA CK804 chipset (Sun Ultra 20 workstation). Other drivers are either available on-line on the Sun product support pages, or on the Supplemental CDROM.

The Sun Ultra 20 workstation and the Sun Java Workstations W1100z and W2100z Supplemental CDs contain scripts to automatically install basic NVIDIA video drivers. To run the scripts, first boot with the Supplemental CD and select “Post-Install” option from the boot menu. The script will install either the 32-bit or 64-bit NVIDIA video drivers, and provides scripts to install other necessary drivers as well.

Configuring the Xorg Server for the Solaris 10 OS

The Solaris 10 OS now features the Xorg server (based on the Xfree86 server) as well as the Xsun server that shipped with previous versions of the Solaris OS. To get the Xorg server to recognize the NVIDIA video card properly, first the Xsun server must be unconfigured with the following command:

```
kdmconfig -u
```

Then, a revised X11 configuration file is required for the Xorg server. If this file is not generated automatically when Solaris is installed (or if the user switches from the Xsun to the Xorg server), then it must be recreated with one of these commands:

```
/usr/X11/bin/Xorg -configure
```

or

```
/usr/X11/bin/xorgconfig
```

Edit the copy of the resulting `Xorg.conf` file and make the necessary changes. Then copy the file to `/etc/X11/Xorg.conf`. For more information, see the *Solaris 10 Release Notes* and the `Xorg` man pages in `/usr/X11/man`.

Mounting the Common Data Partition (`/common`)

The multiboot reference implementation described in this document defines a common data partition. Each operating system (Windows XP, Linux, and the Solaris OS) can be configured to access the common data partition. Since the common data partition is formatted as a FAT32 file system, it is subject to file naming restrictions, and may only recognize filenames that adhere to “8.3” naming conventions.

For Windows XP, the common data partition can be mounted as the next logical drive letter (e.g., `D:`) For Linux, if the partition is not mounted automatically during installation (as usually occurs), then the following entry can be made in `/etc/fstab` :

```
/dev/hda4            /common             vfat     defaults   0    0
```

For the Solaris OS, the following entry must be made in `/etc/vfstab`:

```
/dev/dsk/c0t0d0p4      /dev/rdsk/c0t0d0p4   /common   pcfs      1    yes -
```

Note that in the Solaris OS, `/dev/dsk/c0t0d0p4` refers to the fourth system disk partition defined in the MBR, which is the common data partition. This numbering is in contrast to the naming convention for mounting a Solaris slice (such as `/`, swap, `/export/home`, `/opt`, etc.), that uses the syntax `/dev/dsk/c0t0d0s4` (**p** for partition and **s** for slice).

Reinstalling GRUB

If the Sun Java Desktop System is not installed last (or if Windows or the Solaris OS is reinstalled *after* the installation of JDS), it may be necessary to reinstall the GRUB boot loader. Use the following procedure to reinstall GRUB back into the MBR:

1. Shutdown or restart the system, and reboot with CDROM #1 of Sun Java Desktop System Version 3. The installation begins with a “Welcome” screen in many languages, followed by a display of various installation options.
2. Choose “Rescue” from the list of installation options, and log in as root. From the shell, perform these steps:

```
# grub
grub> find /boot/grub/stage1
(hd0,4)
grub> root (hd0,4)
filesystem is ext2fs, partition type 0x83
grub> setup (hd0)
Checking if “boot/grub/stage1” exists...yes
Checking if “boot/grub/stage2” exists...yes
Checking if “boot/grub/e2fs_stage1_5” exists...yes
Running “embed /boot/grub/e2fs_stage1_5(hd0)”...16 sectors
are embedded... succeeded
Running “install /boot/grub/stage1(hd0) (hd0)1+16p (hd0,4)
/boot/grub/stage2/boot/grub/menu.lst”... succeeded
Done.
grub> quit
# reboot
```

The `find` command locates the GRUB files in the fifth partition (counting from 0).This is the logical volume where Linux root was previously installed. Then the GRUB `setup` command is used to install the MBR with a pointer to the GRUB boot loader. After the system reboots, the GRUB boot loader menu should be displayed, allowing the user to select an operating system to boot.

Chapter 3
Summary

The ability to boot multiple operating systems on a Sun x64 workstation based on the AMD Opteron processor enables a great degree of flexibility and convenience for certain customers. This ability means that the three major x86 compatible operating systems in use today — the Solaris OS, Linux, and Windows operating systems — can all be deployed on a single system disk, allowing the user to choose between them at boot time.

This flexibility can increase the utilization of existing computing resources. Some users, such as software developers, can conveniently access multiple applications and different operating systems on an ongoing basis. With this ability, developers and others now have the option to consolidate a collection of dedicated test platforms into a single, well-utilized resource. The same benefits apply to anyone who needs quick access to multiple platforms without having to commit separate individual systems to the task.

This Sun BluePrints article has attempted to provide the reader with Sun recommended strategies for developing a multi-boot configuration. Using the tools and reference implementations discussed here, the reader can extend configurations to meet their own specific requirements. Beyond developing configurations based on the information presented here, the reader's logical next step is to adjust and fine-tune configurations to optimize performance and security characteristics.

About the Author

Barton Fiske has been with Sun for 12 years, in the industry for 17 years, and has a BS in Computer Science from the Rochester Institute of Technology. In his tenure at Sun, he has worked in a variety of challenging and creative capacities. Barton is the co-author of two books on Java: *Web Site Programming with Java* (Harms, Fiske, Rice), and *Advanced Java Techniques for Developers* (Berg, Fritzenger, et al). Most recently, Barton has been assigned to the Network Systems Group, where he works on developing customer-facing solution prototypes and demos featuring AMD Opteron based systems, including the latest workstation offering, the Sun Ultra 20 workstation.

Acknowledgments

The author would like to recognize the following individuals for their contributions to this article:

- Peter Pak is a contractor for Sun and has done extensive work using the Ranish Partition Manager on Sun Java Workstations W1100z and W2100z. Peter's focus has been on developing effective and versatile system configurations to meet the frequently required demonstration capabilities needed by the Sun corporate trade show group.
- Ginny Henningsen is a Sun contractor (and former Sun Systems Engineer) who writes technical white papers and other marketing collateral. She patiently and thoroughly worked through the procedures described herein and provided much needed translation of somewhat vague concepts into practical terms.

Ordering Sun Documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. Readers living in the United States, Canada, Europe, or Japan, can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` Web site enables users to access Sun technical documentation online. Users can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To obtain Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/`

References

Product-related download sites are as follows:

- `http://www.sun.com/desktop/workstation/w1100z/downloads.html`
- `http://www.sun.com/desktop/workstation/ultra20/downloads.html`

Solaris Installation Guide. 817-5768. To access this book online, go to `http://docs.sun.com.`

Preboot Execution Environment (PXE) Specification Version 2.1: `ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf`

Linux Network Install HOWTO: `http://www.linux.org/docs/ldp/howto/Network-Install-HOWTO.html`

Red Hat System Administration Guide — Installations: `http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/sysadmin-guide/pt-install-info.html`

SuSE YaST Auto Installer: `http://www.suse.de/~nashif/autoinstall/`

“Configuring JumpStart Servers for Solaris and Linux Provisioning on Sun x86-Based Servers” (Sun Blueprints article), `http://www.sun.com/blueprints`

“Multi-booting Solaris and other operating systems,” Mariusz Zynel, December, 2003. `http://multiboot.solaris-x86.org/`

Other useful Web resources include:

- `http://members.tripod.com/~diligent/harddisk.htm`
- `http://members.tripod.com/~diligent/hd-partn.htm`
- `http://www.trombettworks.com/multi-boot.htm`
- `http://www.ranish.com/part/primer.htm`
- `http://www.comptechdoc.org/os/windows/ntwsguide/ntwsbooting.html`

Appendix A

A Multiboot Solution to Support More Than Four Operating Systems

This appendix describes a second, more complex multiboot reference implementation. In this case, the system disk for a Sun x64 system with AMD Opteron processors is divided into twelve virtual partitions and configured with seven operating systems. The Ranish Partition Manager partitioning tool is used to configure the solution. Note that the use of this tool is not an exclusive means of achieving such a solution, and Sun makes no endorsement regarding the use of this software. This reference implementation is provided simply as an example of how a Sun x64 system can be configured to support more than four operating systems.

Caution – Prior to this procedure, any data on the system should be backed up. Partitioning the disk will destroy any existing data and operating system information. If possible, use a blank disk or one that can be reformatted without concern about loss of data.

The Ranish Partition Manager

For the this reference implementation, Version 2.44 of the Ranish Partition Manager was used to define partitions on the disk drive. (To download the Ranish Partition Manager, visit `http://www.ranish.com/part/`.) Although this tool is somewhat dated as of this writing, the Ranish Partitioning Manager is able to overcome the four-partition limitation by creating a virtual partition table, and has been found to work consistently for the purposes of this reference implementation. The tool keeps track of partition boundaries, dynamically inserting and deleting partition entries in the MBR at boot time.

Recall that the MBR contains a partition table with four descriptors, one for each primary or extended partition. Every cylinder range that is not included in the descriptors in the MBR is treated as unpartitioned space — random bytes that are ignored by all operating systems and partitioning programs. This unpartitioned space is not visible to the currently running operating system. The Ranish Partition Manager is able to exceed the four-partition limitation by manipulating partition entries in the MBR, and by treating all other areas of the disk as unpartitioned space. Although more than four partitions may be defined, the tool maps only four partition entries into the MBR at any one time. The tool allows the user to modify MBR entries, deleting and reinserting them to control visibility and the ability to boot these partitions. The Ranish Partition Manager can define up to 32 virtual partitions.

Partitioning the Disk

This larger reference implementation uses a partitioning layout similar to that shown in Table 5. The table reflects how the Ranish boot loader actually presents the disk layout in its user interface.

Table 5. Example multiboot partition information for more than 4 operating systems on a 73 GB disk

Part. # & Type	Row	File System Type	OS Target	Starting			Ending			Size [KB]
				Cyl	Head	Sect	Cyl	Head	Sect	
MBR		MBR	n/a	0	0	1	0	0	1	0
1 Pri		Unused	Unused	0	0	2	0	0	63	31
2 Pri		NTFS	Windows NT	0	1	1	1,275	254	63	10,249,438
3 Pri		Linux swap	Solaris 10, JDS Vers. 3	1,276	0	1	2,551	254	63	10,249,470
4 Pri		Linux swap	Solaris 9	2,552	0	1	3,827	254	63	10,249,470
5 Pri		Linux ext2fs	Fedore Core 3 x64	3,828	0	1	4,593	254	63	6,152,895
6 Pri		Linux ext2fs	JDS Vers. 2	4,594	0	1	5,359	254	63	6,152,895
7 Pri	2*	Linux ext2fs	JDS Vers. 3, b27	5,360	0	1	6,125	254	63	6,152,895
8 Pri		Linux ext2fs	RedHat AS8 x64	6,126	0	1	6,891	254	63	6,152,895
9 Pri	3	Linux swap	Linux swap	6,892	0	1	7,147	254	63	2,056,320
10 Pri	4	Windows FAT-32 LBA	/common	7,148	0	1	8,922	254	63	14,257,687
11 Pri	1	Boot Manager	Boot Manager	8,923	0	1	8,923	254	63	8,032
12 Pri		Unused	Unused	8,924	0	1	8,924	169	37	5,342

In this more complex multiboot reference implementation, a common swap partition is defined (partition #9) that can be shared between the different Linux operating system variants. This shared swap partition can be mounted to support either Fedora Linux (a free distribution of the Linux kernel from Red Hat), Sun Java Desktop System Versions 2 or 3, or RedHat AS8 Linux (partitions #5, 6, 7, or 8). However, the swap partition cannot be used in conjunction with a Solaris OS instance or a Windows-based operating system instance.

Note that due to its size, the Ranish Partition Manager requires its own partition. If this tool is used, it is recommended that the last complete cylinder of the disk be allocated for this partition, so that the partition manager has well-defined beginning and ending boundaries. In this implementation, the Ranish Partition Manager uses partition #11, and the last incomplete cylinder is set aside as partition number # 12.

The Ranish Partition Manager uses the “Row” column to indicate which four partitions currently have entries mapped in the MBR. In the boot scenario depicted in Table 5, those four partitions are currently partition # 11 (the Ranish Boot Manager), partition #7 (JDS Version 3 root), partition #9 (Linux swap), and partition #10 (the common data partition). JDS Version 3 in partition 7 is designated as the active boot partition, as indicated by the “*”. Note that the JDS or Linux swap partition must be defined as one of the four partition entries in the MBR in order for JDS or Linux to boot and run properly. (If Linux swap is not selected as one of the four partitions, then Linux may not function properly when applications exceed the limit of physical memory.) Since the common data partition is also a current entry in the MBR, it can also be mounted by JDS at boot-time.

A similar example is shown in Table 6. This example features the same partition table but with different entries in the “Row” column (partitions #11, 2, 3, and 10) for the Ranish Partition Manager. In this example, the partition entries in the MBR include the Ranish Boot Manager, the Solaris 10 OS, the Windows NT operating system, and the common data partition. The Solaris 10 OS is designated as the active boot partition. (Note that the Ranish Partition Manager reports the file system type of 0x82 as Linux swap, even though the Solaris OS is loaded in that virtual partition; see the earlier discussion of partition types in Chapter 2.)

Table 6. Partition information used by the Ranish partition manager

Part. # & Type	Row	File System Type	OS Target	Starting			Ending			Size [KB]
				Cyl	Head	Sect	Cyl	Head	Sect	
MBR		MBR	n/a	0	0	1	0	0	1	0
1 Pri		Unused	Unused	0	0	2	0	0	63	31
2 Pri	**3**	NTFS	Windows NT	0	1	1	1,275	254	63	10,249,438
3 Pri	**2***	Linux swap	Solaris 10, JDS Vers. 3	1,276	0	1	2,551	254	63	10,249,470
4 Pri		Linux swap	Solaris 9	2,552	0	1	3,827	254	63	10,249,470
5 Pri		Linux ext2fs	Fedore Core 3 x64	3,828	0	1	4,593	254	63	6,152,895
6 Pri		Linux ext2fs	JDS Vers. 2	4,594	0	1	5,359	254	63	6,152,895
7 Pri		Linux ext2fs	JDS Vers. 3, b27	5,360	0	1	6,125	254	63	6,152,895
8 Pri		Linux ext2fs	RedHat AS8 x64	6,126	0	1	6,891	254	63	6,152,895
9 Pri		Linux swap	Linux swap	6,892	0	1	7,147	254	63	2,056,320
10 Pri	**4**	Windows FAT-32 LBA	/common	7,148	0	1	8,922	254	63	14,257,687
11 Pri	**1**	Boot Manager	Boot Manager	8,923	0	1	8,923	254	63	8,032
12 Pri		Unused	Unused	8,924	0	1	8,924	169	37	5,342

Notes on Partitions

There are some notable differences with respect to partitioning in this reference implementation when using the Ranish Partition Manager, for example:

- ***Common Data Partition.*** To be accessible from an operating system, the common data partition (`/common`) must be marked as one of the visible partitions in the MBR.
- ***Linux swap.*** Be sure to mark Linux swap as an active partition whenever you are trying to boot a Linux operating system. Otherwise Linux will not function properly when applications exceed the limit of physical memory. Also, Linux swap may be shared between all Linux operating systems installed on the system. (Partition #9 in Table 6, for example, can be shared as Linux swap for JDS and Linux boot partitions — specifically as swap for Linux root partitions #5, 6, 7, or 8.)

- ***Ranish Boot Manager partition.*** The last complete cylinder is used for the Ranish boot manager partition, with the last incomplete cylinder left unused. It is recommended that the last complete cylinder of the disk be allocated for the Ranish boot manager partition, so that the partition manager has well-defined beginning and ending boundaries. In this implementation, the Ranish Partition Manager uses partition #11, and the last incomplete cylinder is set aside as partition #12.

Using the Ranish Partition Manager

The Ranish Partition Manager is installed from a floppy disk. Once the partitions are laid out, individual operating systems can be installed, along with appropriate post-install procedures as with the previous reference implementation.

Creating a Bootable Ranish Floppy

To use the Ranish Partition Manager, a bootable floppy disk must first be created by running the Ranish Partition Manager executable (`part244.exe`). This requires a USB-attached floppy device to be attached to the Sun Ultra 20 workstation, or to the Sun Java Workstations W1100z or W2100z.

To create a bootable Ranish floppy:

1. Boot the system with the existing or USB-attached floppy drive using a Windows operating system, or use a floppy on another Windows-based system. Download the Ranish zip file from the Ranish Web site (`http://www.ranish.com/part/`) and uncompress it.
2. Format a floppy disk. Run the executable `part244.exe` to create a bootable image on the formatted floppy. This executable file creates a bootable floppy disk with an executable called `part.exe` installed.

Partitioning with the Ranish Partition Manager

To partition the disk with the Ranish Partition Manager:

1. Insert the bootable Ranish floppy in the Sun x64 system's existing or USB-attached floppy drive. Reboot the Sun system, and press F8 during the boot sequence to load the Boot Menu. From the Boot Menu, select the option to boot from "Removable Devices" to boot from the floppy drive.
2. Run the Ranish Partition Manager executable that resides on the floppy:

```
A:\ part.exe
```

A Ranish Partition Manager window similar to that in Figure 7 appears on the screen. The Ranish Partition Manager will now allow you to create partitions for the installation of other operating systems.

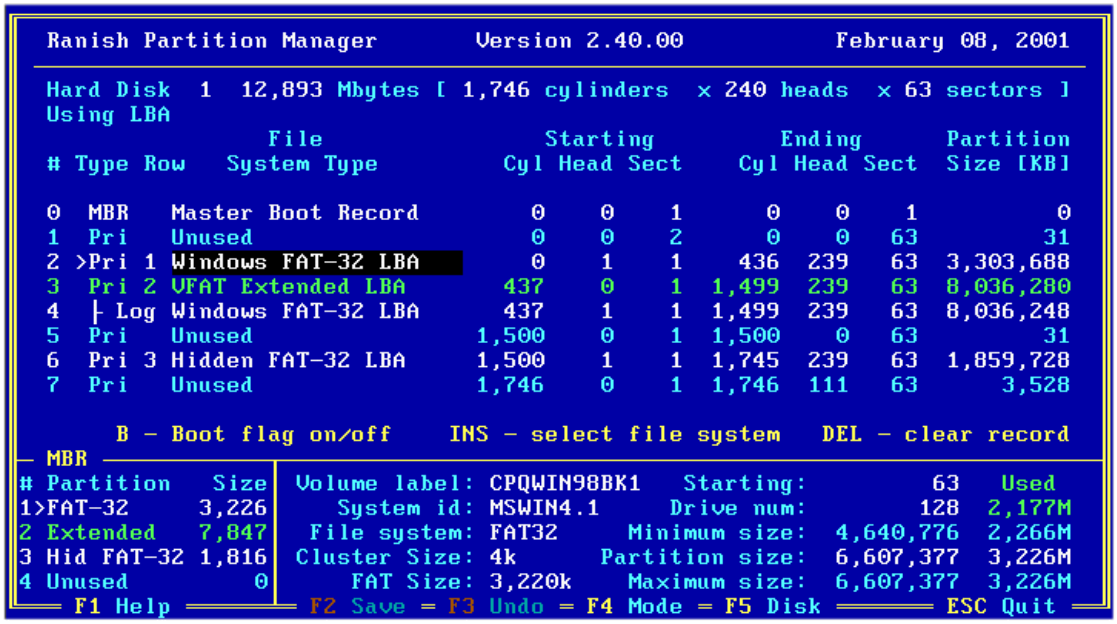

Figure 7. Example of the User Interface for the Ranish Partition Manager.

3. Define the virtual disk partitions as needed, and press F2 to save four active partition definitions (defined in the Row column) to the MBR. Pressing the ‘escape’ key exits the partition manager, and pressing ‘enter’ selects the highlighted partition and allows it to be changed. Note that function key F10 allows the simulation of partitioning steps without actually executing them. Using the Ranish Partition Manager, the partition table can be defined according to your specific partitioning goals. Remember that the “Row” column indicates which four virtual partitions are currently mapped and saved as entries in the MBR on the hard disk.

Operating System Installation

Once the Ranish Partition Manager is set up to select virtual partitions, the installation process consists of selecting four partition entries at a time and then configuring those four partitions with one or more operating systems. Select the first partition as the active boot partition, for example, and then install it with a Windows-based operating system. Continue with the other virtual partitions and operating systems until the desired configuration is reached.

Post-Installation

After the installation of the operating systems, be sure to perform the post-installation steps listed at the end of Chapter 2, “A Basic Multiboot Configuration.” The post-installation steps include:

- Installing and configuring drivers for each operating system
- The generation of a configuration file for use with the Xorg server (for the Solaris 10 OS)
- Mounting and configuring the common data partition.

Use the Supplemental CD that is shipped with each Sun Opteron-based workstation or server, or refer to the appropriate Sun system Web site to download available drivers and other information:

- `http://www.sun.com/desktop/workstation/w1100z/downloads.html`
- `http://www.sun.com/desktop/workstation/w2100z/downloads.html`
- `http://www.sun.com/desktop/workstation/ultra20/downloads.html`