Hound Dog Terminal Manager

User Guide

Release V1R0M1

Table of Contents

Table of Figures

Introduction

Computer labs in today's businesses contain many Linux systems to run their operations. Computer labs can contain many computer systems in a single 19 inch rack, in lab rows of hundreds of feet, accounting for hundreds or thousands of computer systems. Often these systems are only available thru IP access, using such tools as Putty, requiring command line syntax to view and monitor system health, activity, and configuration.

While web server interfaces can be deployed, these tools are usually designed to perform specific tasks. As a result, terminal sessions are often resorted to, using such tools as Putty to satisfy various queries and maintenance.

Because terminal sessions are character-based, it is assumed that benefits found in GUI interfaces such as point-and-click cannot be implemented in terminal sessions. The Hound Dog Terminal Manager addresses this need.

The Hound Dog Terminal Manager provides comprehensive insights into the configuration and activity of a Linux System using a "point and click" user interface developed specifically for a Terminal Window. Instead of having to remember command line syntax and keystrokes, the HDog Terminal Manager allows these input queries and commands to be issued using Menus, Pulldowns, and Push Buttons and presents its results in a structured, easy to understand format.

The Hound Dog Terminal Manager provides three distinct features:

- A GUI-like user interface for easy, error-free user interaction.
- A user modifiable and extendable menu structure built from an XML file.
- A public API that includes SDK functionality for functionality extensions.

- An additional feature of the Hound Dog Terminal Manager is that it provides a consistent user interface across different variants of Linux. This release of HDog provides support for Ubuntu and Fedora, but future releases will provide support for other Linux variants such as openSUSE, PCLinuxOS, Debian, Mandriva, and Sabayon/Gentoo. Instead of having to remember the variations of command line syntax associated with these Linux variants, HDog accounts for those differences while providing a consistent, easy-to-understand user interface.

Product Versions

The Houng Dog Terminal Manager is installed as an Evaluation Version. This Evaluation Version can be used on the installed system for 30 days.

You must purchase a registration key within this 30 day window or it will expire. After expiration, if/when you invoke the Houng Dog Terminal Manager, only the Registration option can be performed.

You can purchase this registration key on the www.hounddogtech.us website or by requesting via email to hdogtech@comcast.net. Paypal is the preferred payment type but other forms of payment can be made.

You can purchase either of the two Registered Versions:

Local Version

The Houng Dog Terminal Manager Local Version can be installed on up to two computers. This is a complete, independent installation on each system.

Network Version

The Houng Dog Terminal Manager Network Version can be installed on up to four computers and contains all the features and characteristics of the Local Version.

Each installation of the Houng Dog Terminal Manager Network Version can support up to 100 remote Linux systems in the network. The Houng Dog Terminal Manager will securely perform its functions on the Remote Systems and return its results to the user on the installed system.

This Network Version has the advantage of not needing to install the Houng Dog Terminal Manager on each Linux system it needs to run on. It may provide an ideal solution for configurations that contain multiple embedded Linux systems connected in a private network with a single Linux system having public network access. In those cases, the Houng Dog Terminal Manager would be installed on the Linux system with public network access and use private network to perform actions on the embedded Linux systems.

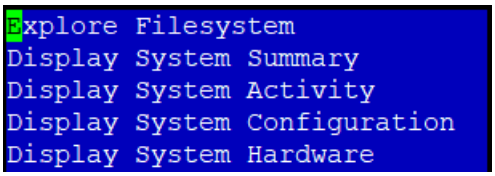
The Network Version also provides for Secure Socket Trust to be established between the Houng Dog Terminal Manager signon User ID on the local Linux system with a User account on a remote Linux system using user-friendly point and click actions.

Note: The Evaluation Version supports up to one Remote System to be specified.

User Interface

The Hound Dog Terminal Manager, (**HDog**), user interface utilizes colors and attributes to depict point-and-shoot screen objects like Menu Items, Push Buttons, and Pulldowns. Keyboard use of **Tab** and **Arrow** keys, as well as an attached mouse, allow the user to point to the task s/he would like to perform.

Consider the Menu Items on the right →



The cursor indicates that the **Explore Filesystem** menu item is the current selection. Pressing **Enter** or clicking the mouse on this entry will cause that function to be performed. Likewise, Tabbing to another menu item and/or clicking on another menu item will cause that selected menu item function to be performed.

Consider the Action Bar on the right →

The cursor indicates that the current Push Button is for the Forward action. Tabbing will jump to the **Return**, **Save**, **Print**, **Help**, and **Exit** Push Buttons. Pressing **Enter** or a mouse click on a Push Button will cause that function to be performed.

When **HDog** starts up the user will be presented with the **HDog Main Menu**. Selecting a Submenu such as **Display System Hardware** will cause that submenu to be presented. Submenus will have an additional Push Button displayed called **Return**. Upon selection, the **Return** Push Button will cause the next 'Higher' menu to be presented.

Pressing the **Exit** Push Button will cause **HDog** to immediately terminate normally.

Here is some information of the other **HDog** Push Buttons.

Help: Help information is provided for the panel.

Forward: This Push Button is only displayed when HDog realizes that there is more information left. Selecting this Push Button will cause HDog to page forward.

Save: Saves the current action's contents as a new dataset in the /tmp folder.

Print: Prints the current action's contents using the **lpr** utility.

A typical Putty screen is comprised of 24 rows with 80 columns each. These screens can be expanded to over 50 rows and 180 columns. **HDog** makes use of any screen expansion to provide as much information for the user as possible.

Main Menu (Local Version)

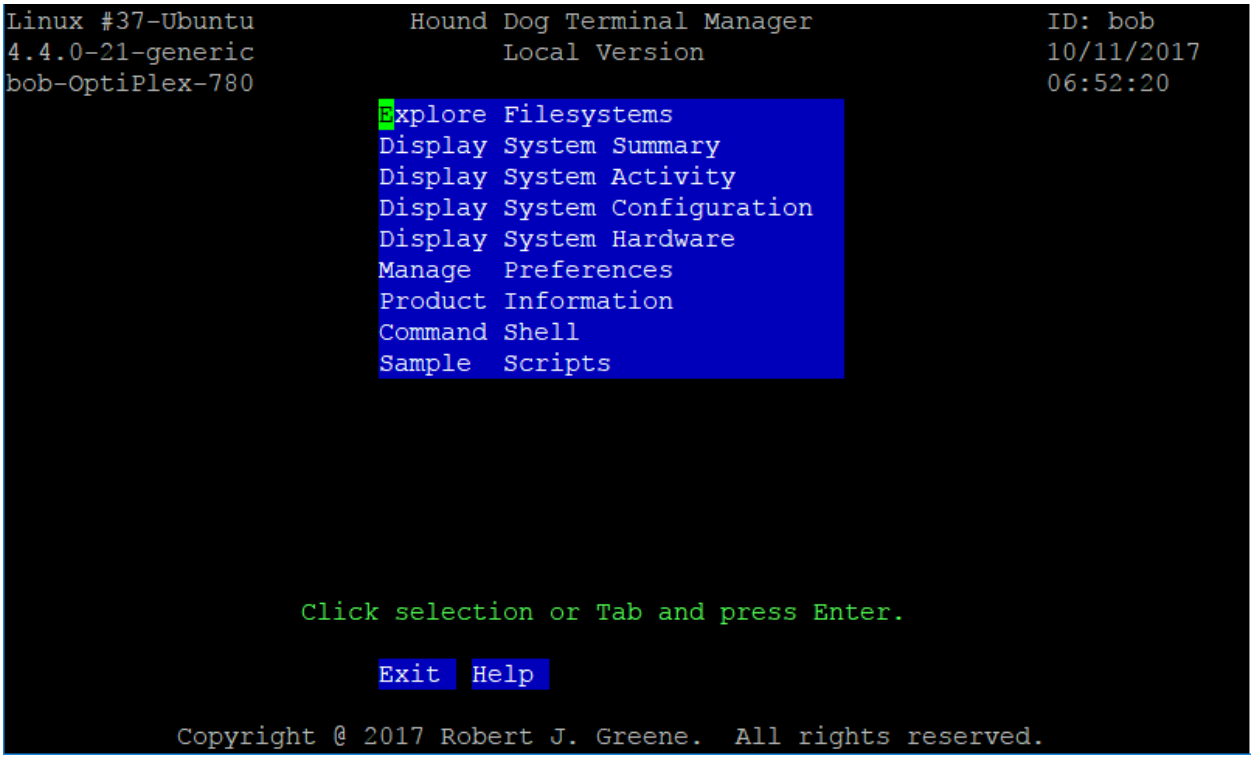


Figure 1: Main Menu.

Above is the Hound Dog Terminal Manager, (**HDog**), **Main Menu**. The content of the **Main Menu** is dynamically built from the hdog.xml file, which can be modified by the user as required. Information about the system is displayed in the upper left and the user id and time is displayed in the upper right.

The user can click on a particular Menu Item, or **Tab** to a desired Menu Item and press **Enter**.

Among the above selections:

- **Explore Filesystem**: Beginning at the root folder, display the files and folders of the system.
- **Display System Summary**: Provide a summary of the system's hardware components.
- **Display System Activity**: Submenu of System Activity Menu Items.
- **Display System Configuration**: Submenu of System Configuration Items.
- **Display System Hardware**: Submenu of System Hardware Components.
- **Manage Preferences**: Customize color and attribute preferences.
- **Product Information**: Provide information about the HDog product.
- **Command Shell**: Enter the Linux Command Shell.
- **Help**: Provide Help information.
- **Exit**: Exit the Hound Dog Terminal Manager.

Main Menu (Network Version)

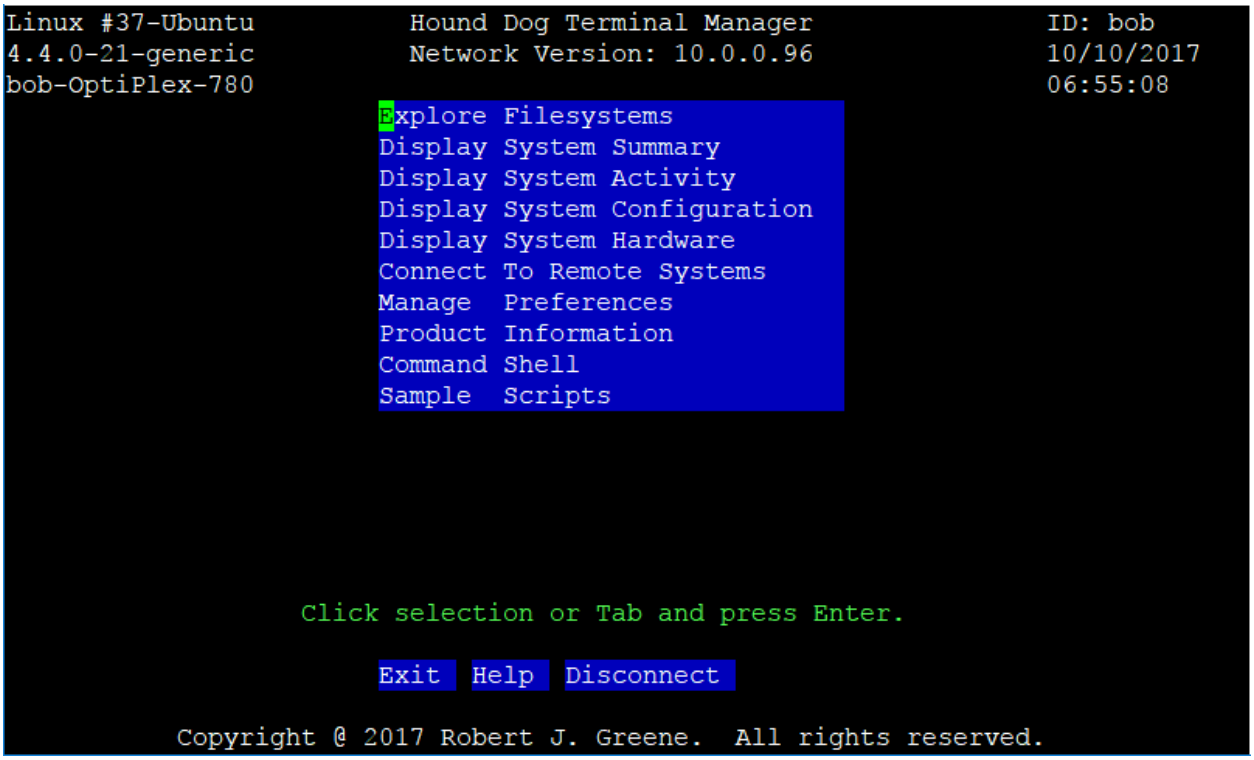


Figure 2: Main Menu (Network Version)..

The Hound Dog Terminal Manager Network Version **Main Menu** is identical to the Local Version with the following exceptions:

- An additional Menu Item is displayed with the title **Connect to Remote Systems**.
  When selected, a panel will be presented to allow you to select a Remote System to Connect to using its predefined IP, Port, and User ID. Once connected, the actions that you select will be performed on that Remote System with the output being presented back on this panel.

- When a Remote System is connected, its IP will be appended to the above panel title. In the sample panel above, this IP is 10.0.0.96.

- A **Disconnect** push button is added to the Action Bar if a Remote System is currently connected. When the **Disconnect** Push Button is selected, the currently connected Remote System will be disconnected and subsequent actions will be performed on the Local Linux system.

The Network version allows you to install the Hound Dog Terminal Manager on one system and perform actions on up to 100 Remote Systems that are network attached to the local system.

Note: The **Manage Preferences** and **Product Information** actions will be performed on the Local Linux system regardless of whether or not a Remote System has been connected.

Explore Filesystems

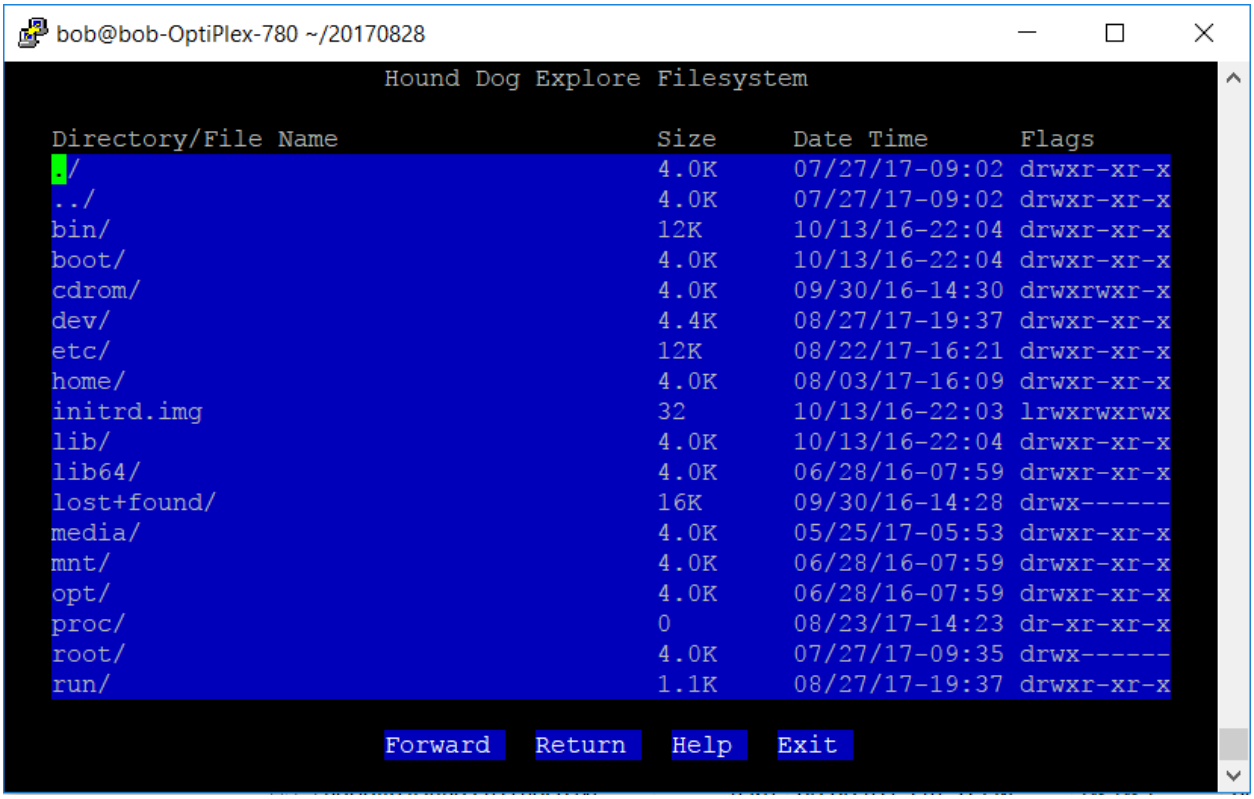


Figure 2: Explore Filesystems.

The term **filesystem** used here refers to the entire hierarchy of directories (also referred to as the **directory tree**) that is used to organize files on this system.

The directories start with the **root directory**, (**/**), which contains a series of subdirectories, each of which, in turn, contains further subdirectories, etc. Subdirectories are displayed with a trailing '**/**' in their name.

Clicking on a **Subdirectory** will result in a display of that **Subdirectory's** contents.

Clicking on a **File** will return detail information about that **File** along with the option to **browse** or **edit** the file if you are authorized to do so.

Clicking on the **../ Subdirectory** name will result in a display of the most recent 'upper' **directory**.

Exploring Folders

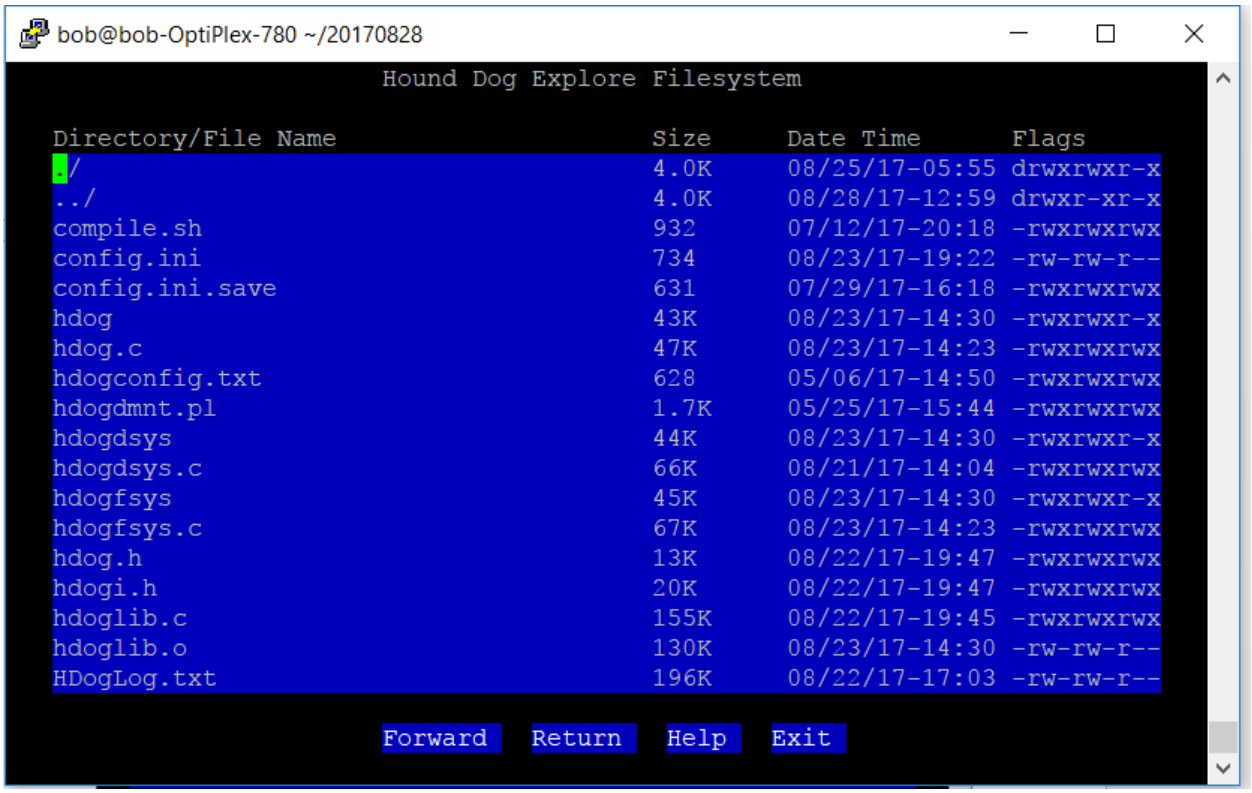


Figure 3: Explore Folder.

Above you will see a subdirectory 'below' the root directory. This subdirectory happens to only contain files.

Clicking on a **File** such as **hdog.h** will return detail information about that **File** along with the option to **browse** or **edit** it if you are authorized to do so.

On the next page you will see the response when you click on the **hdog.h** file entry.

Exploring File Information

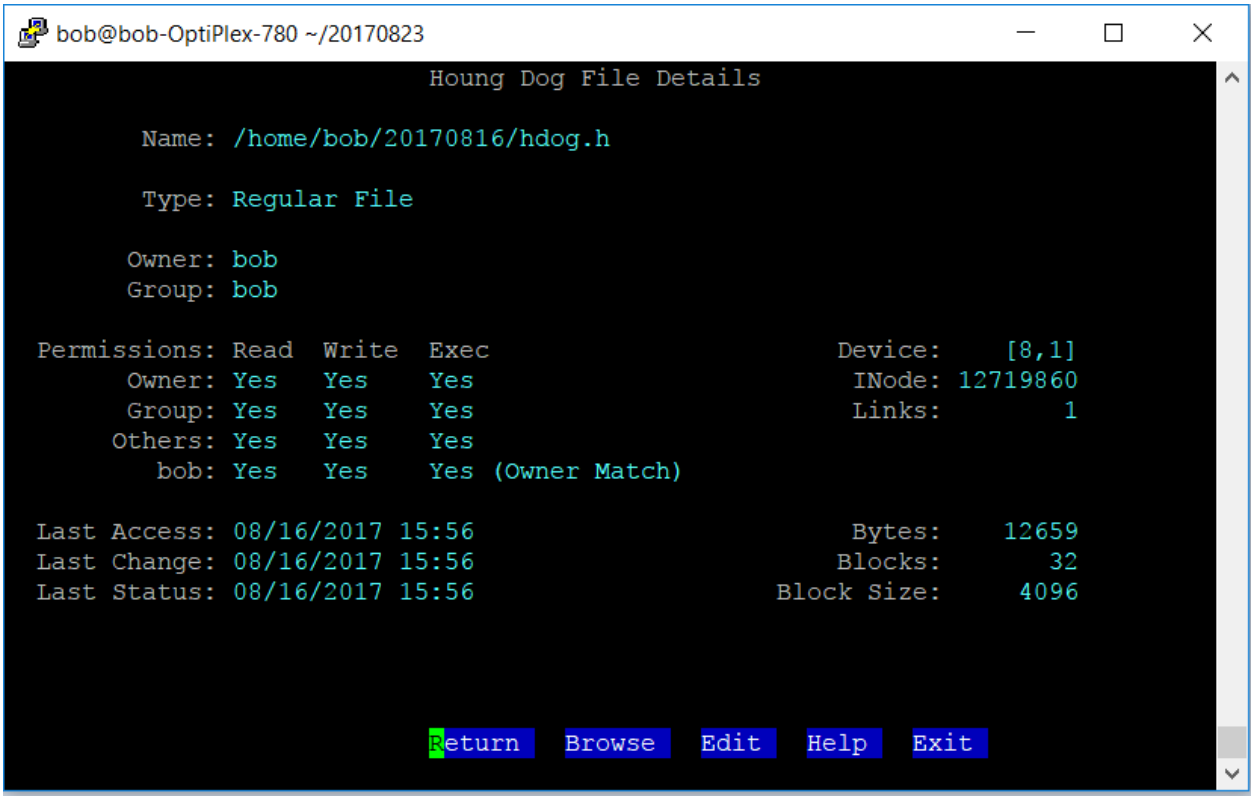


Figure 4: Explore File Information.

Above you will see detail information above a file called **hdog.h**.

The file can be of the following types:

Block Device:	A file that refers to a **block device**.
Character Device:	A file that refers to a **character device**.
DirectoryFIFO/Pipe:	A named pipe.
Symlink:	A symbolic link.
Regular File:	A regular file.
Socket:	A special **file** used for inter-process communication.
Unknown Type:	An unknown file type.

Read, **Write**, and **Execute Permissions** are listed for **Owner**, **Group**, **Others**, and the current logged on **User**. The logged on **User ID** is listed along with the type of **Permission** match that applies to this File.

File Size and **Last Access, Last Change**, and **Last Status Dates** are also displayed.

Browse and **Edit Push Buttons** will be displayed if the user is authorized to perform these actions. Selecting **Browse** or **Edit** will invoke a **vi** session of the file.

Clicking on **Return** will result in a redisplay of the **directory** that contains the above file.

Browsing and Editing Files

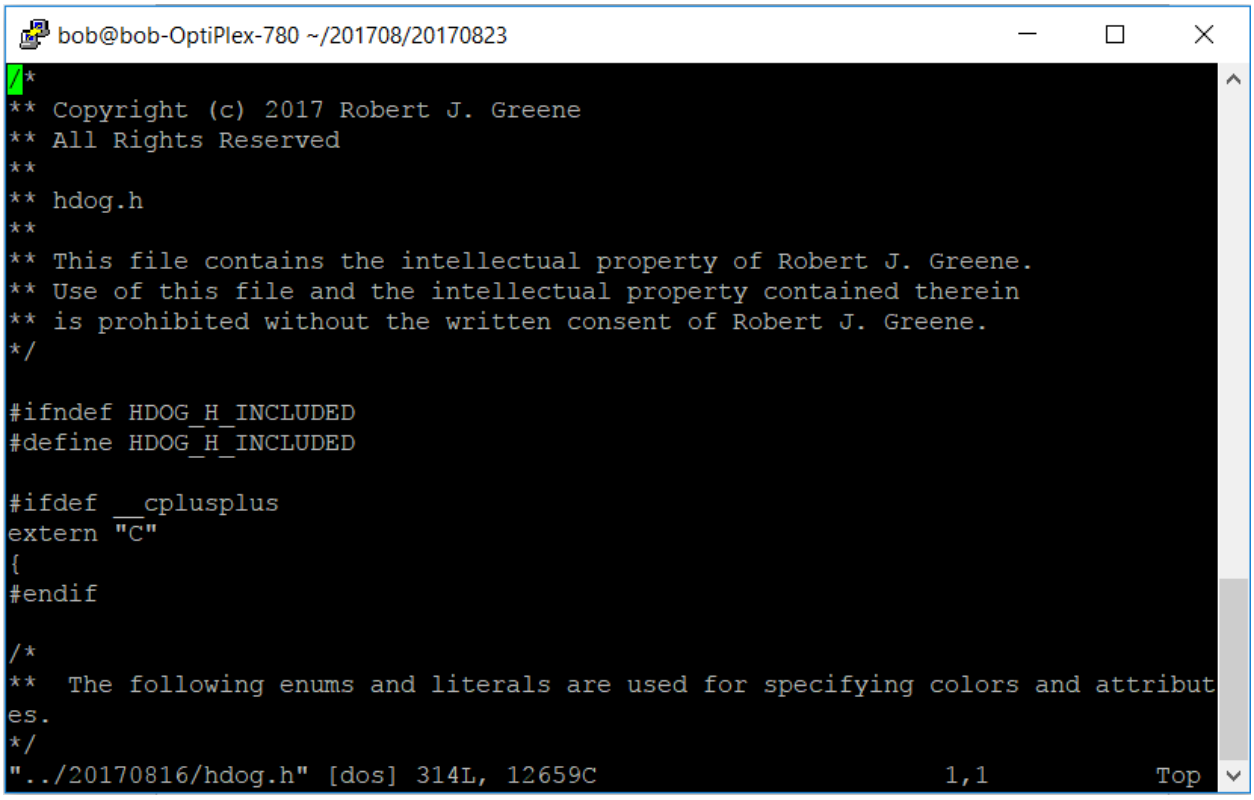


Figure 5: Browsing and Editing Files.

vi is invoked when a user selects a **File** to be browsed or edited.

The above example displays a **vi** session with the hdog.h file when the user has selected Browse in the previous panel.

When the user terminates the session, s/he will be returned to redisplay of the **directory** that contains the above file.

Display System Hardware Summary

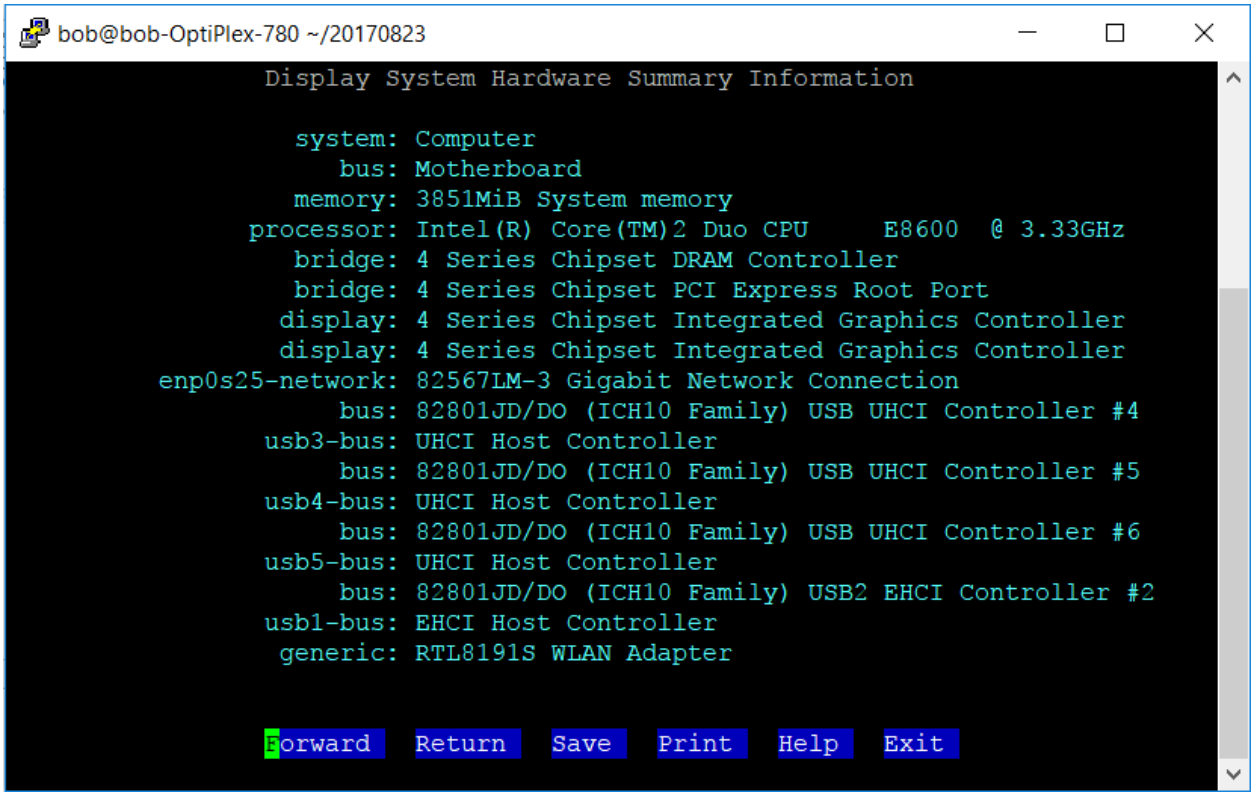


Figure 6: Display System Hardware Summary.

The **System Hardware Summary** display describes the basic components of this system. This includes information about the **Processor**, **Bridges**, **Buses**, **Display Controllers**, **Input Devices**, **Storage Disks**, and **Controllers**.

Display System Activity Submenu

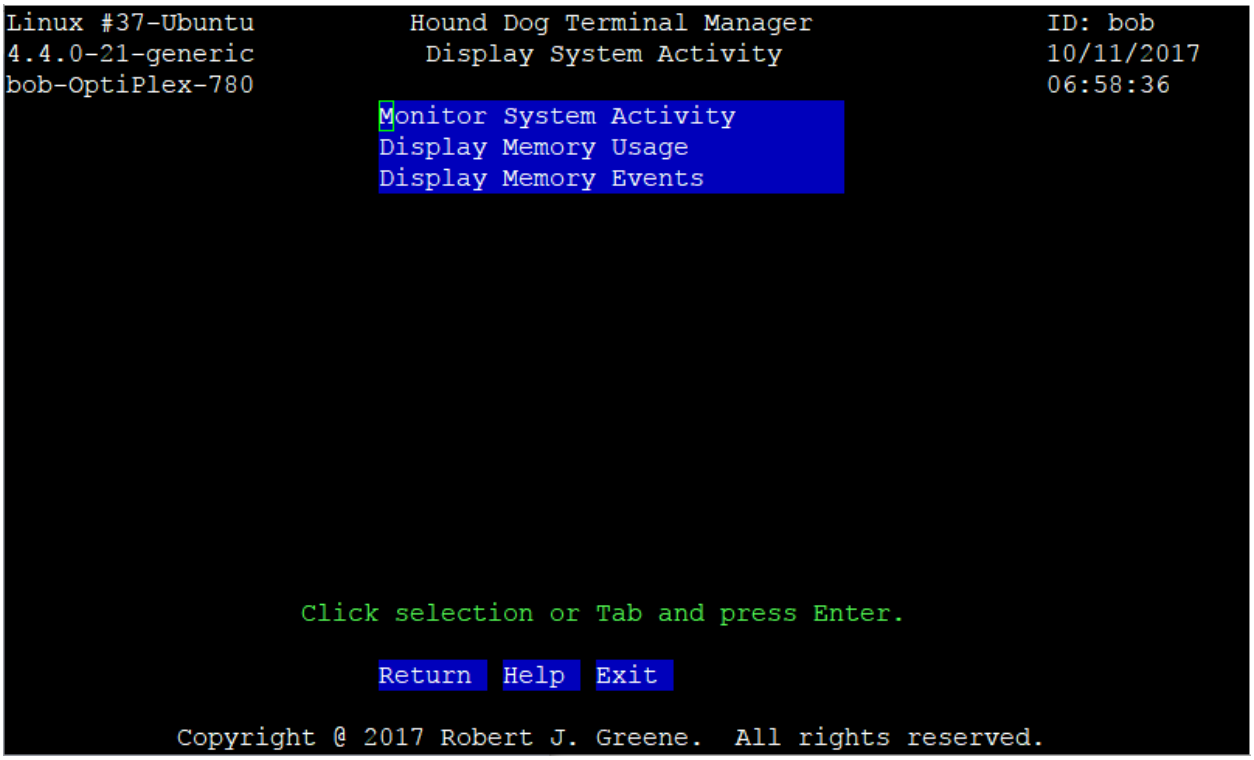


Figure 7: Display System Activity Submenu.

The **Display System Activity Submenu** provides the following menu items:

- **Monitor System Activity**: This is a dynamic real-time view of the CPU and memory activity of this running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel. This uses **Htop**, which is a free (GPL) ncurses-based process viewer for Linux.

  Note: This menu item is currently not supported when connected to a Remote System.

- **Display Memory Usage**: This displays information about the systems **RAM** usage.

- **Display Memory Events**: Reports virtual memory statistics.

Monitor System Activity

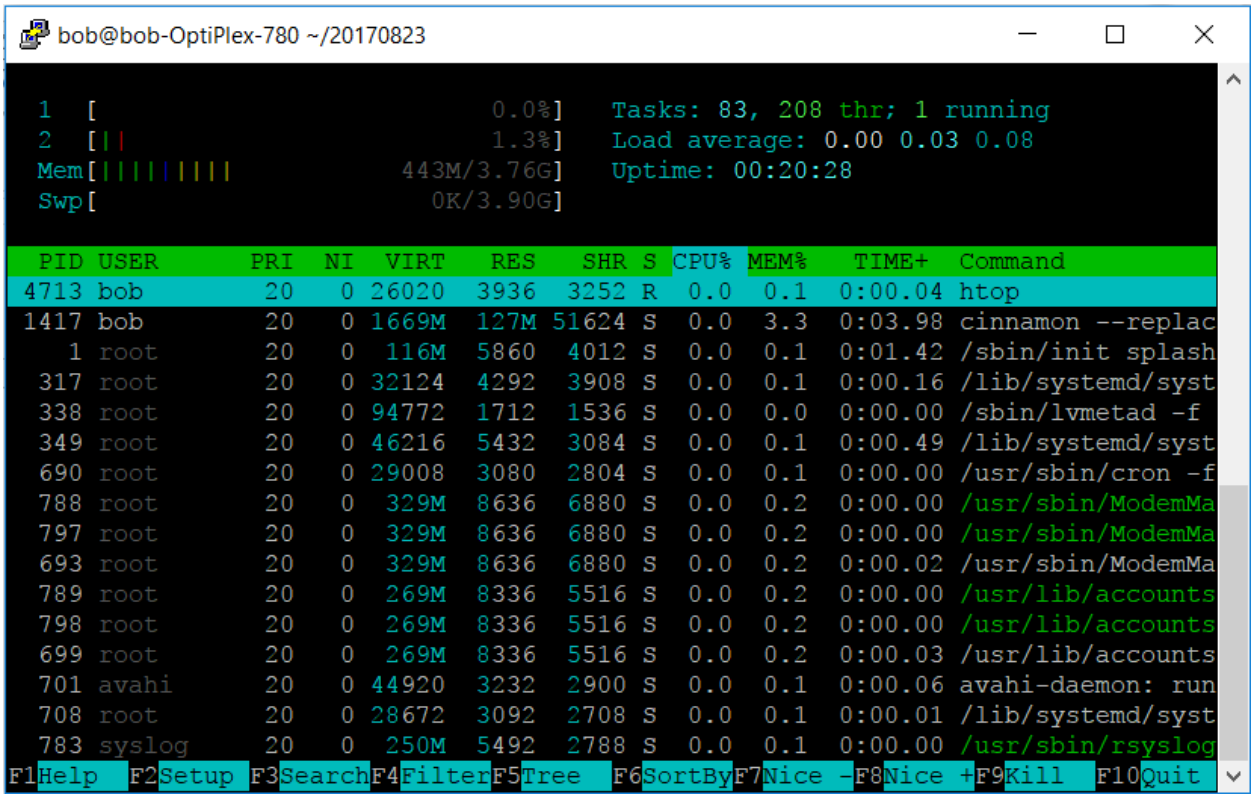


Figure 8: Monitor System Activity.

This is a dynamic real-time view of the **CPU** and memory activity of this running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel. This uses **Htop**, which is a free (**GPL**) ncurses-based process viewer for Linux.

Note: This feature is not supported on Remote Systems.

Display Memory Usage

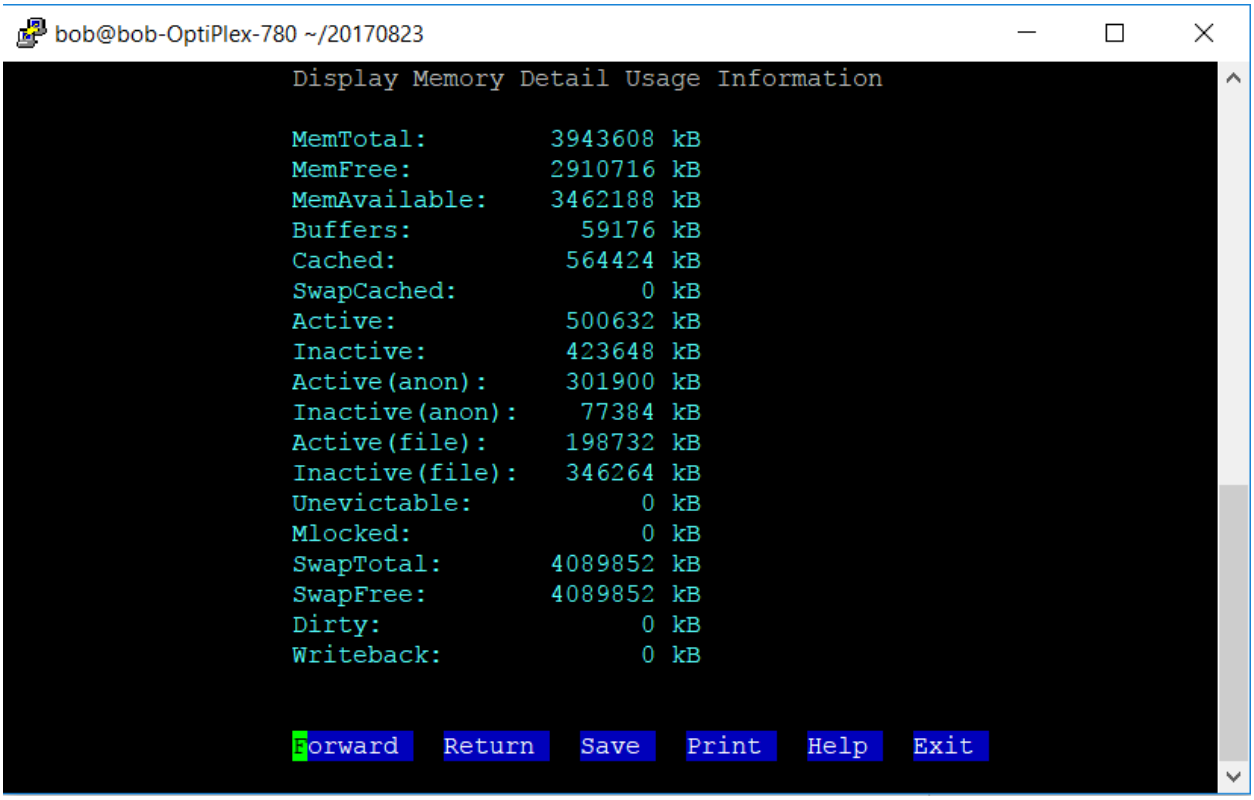


Figure 9: Display Memory Usage.

This panel displays information about this system's **RAM** usage in Kilobytes. Here are some descriptions of the output:

- **MemTotal**: Total amount of physical **RAM**.
- **MemFree**: Amount of physical **RAM** left unused by the system.
- **Buffers**: Amount of physical **RAM** used for file buffers.
- **Cached**: Amount of physical **RAM** used as cache memory.
- **SwapCached**: Amount of swap used as cache memory.
- **Active**: Amount of buffer or page cache memory in active use.
- **Inactive**: Amount of buffer or page cache memory that are free.
- **SwapTotal**: Amount of swap available.
- **SwapFree**: Amount of swap free.
- **Dirty**: Memory amount waiting to be written back to the disk.
- **Writeback**: Memory amount actively being written back to the disk.
- **VMallocTotal**: Total allocated virtual address space amount.
- **VMallocUsed**: Used virtual address space amount.
- **VMallocChunk**: Largest contiguous virtual address space block.

Display Memory Events

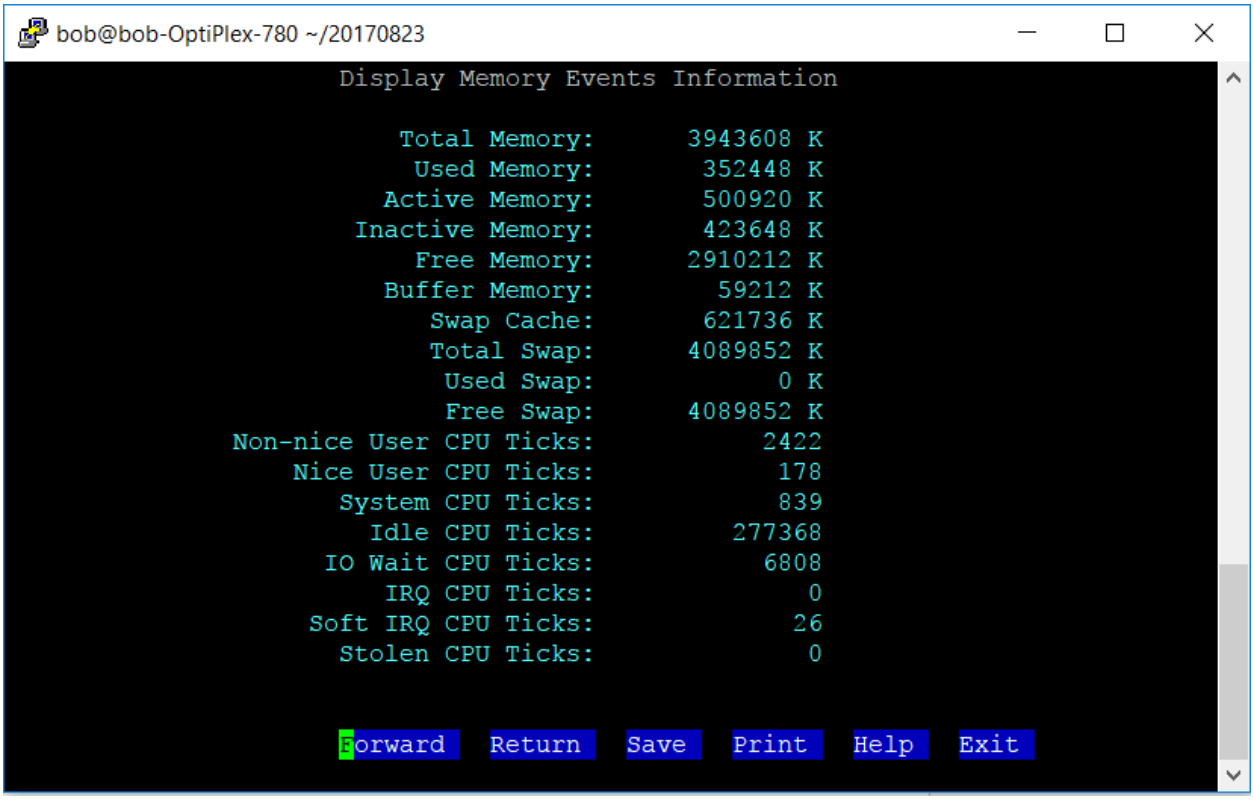


Figure 10: Display Menu Events.

This panel displays information about this system's Memory and CPU usage. Here are some descriptions of the output:

Memory Usage:

- **Total Memory**: Total amount of physical **RAM**.
- **Free Memory**: Amount of physical **RAM** left unused by the system.
- **Buffer Memory**: Amount of physical **RAM** used for file buffers.
- **Total Swap**: Amount of swap available.
- **Free Swap**: Amount of swap free.

CPU Usage:

- **Non-Nice User CPU Ticks**: User **CPU** time that was not niced.
- **System CPU Ticks**: **CPU** Time spent running kernel code.
- **Idle CPU Ticks**: **CPU** Time spent running kernel code.
- **IO Wait CPU Ticks**: Time spent waiting for **IO** to complete.

Display System Configuration Submenu

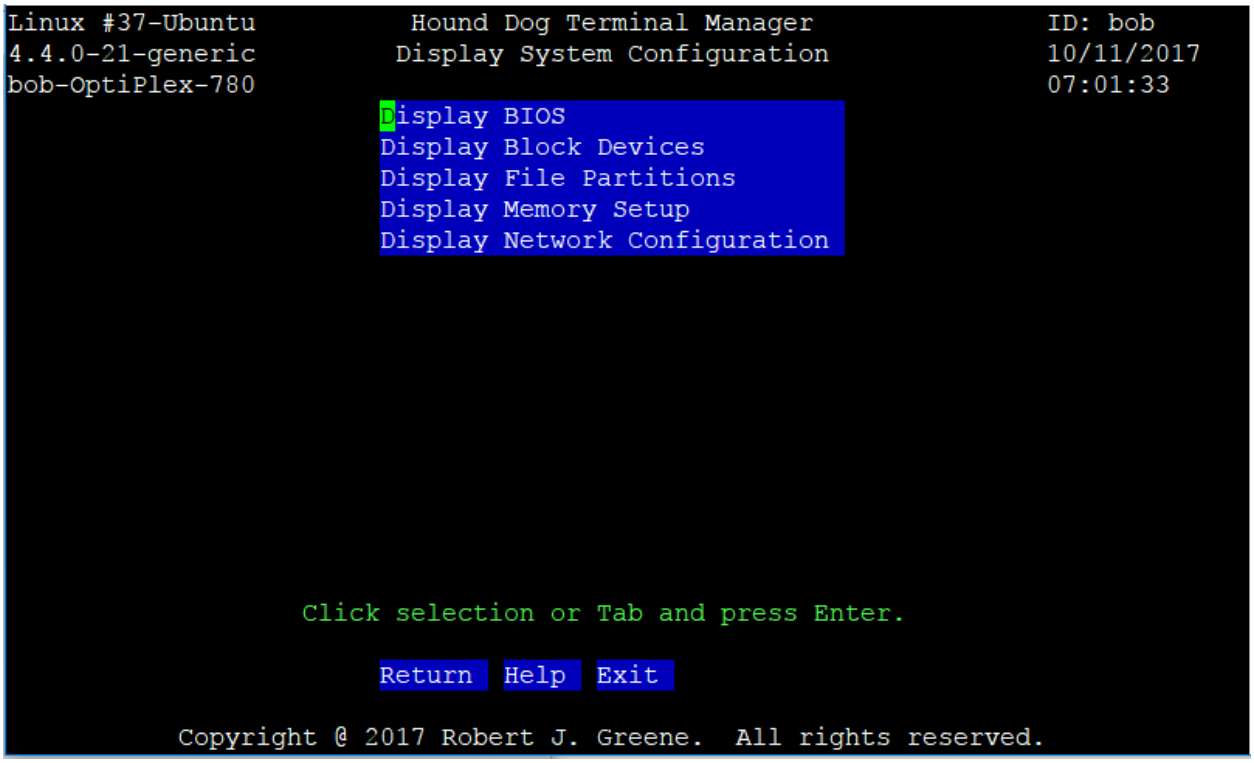


Figure 11: Display System Configuration Submenu.

This Submenu displays various configuration setup of this system:

- **Display BIOS**: Lists non-volatile firmware settings that used to perform hardware initialization during the booting process.
- **Display Block Devices**: Lists all block devices (except **RAM** disks) in a tree-like format.
- **Display File Partitions**: Lists the system partition table.
- **Display Memory Setup**: List physical memory devices.

Display Basic Input Output System (BIOS) Information

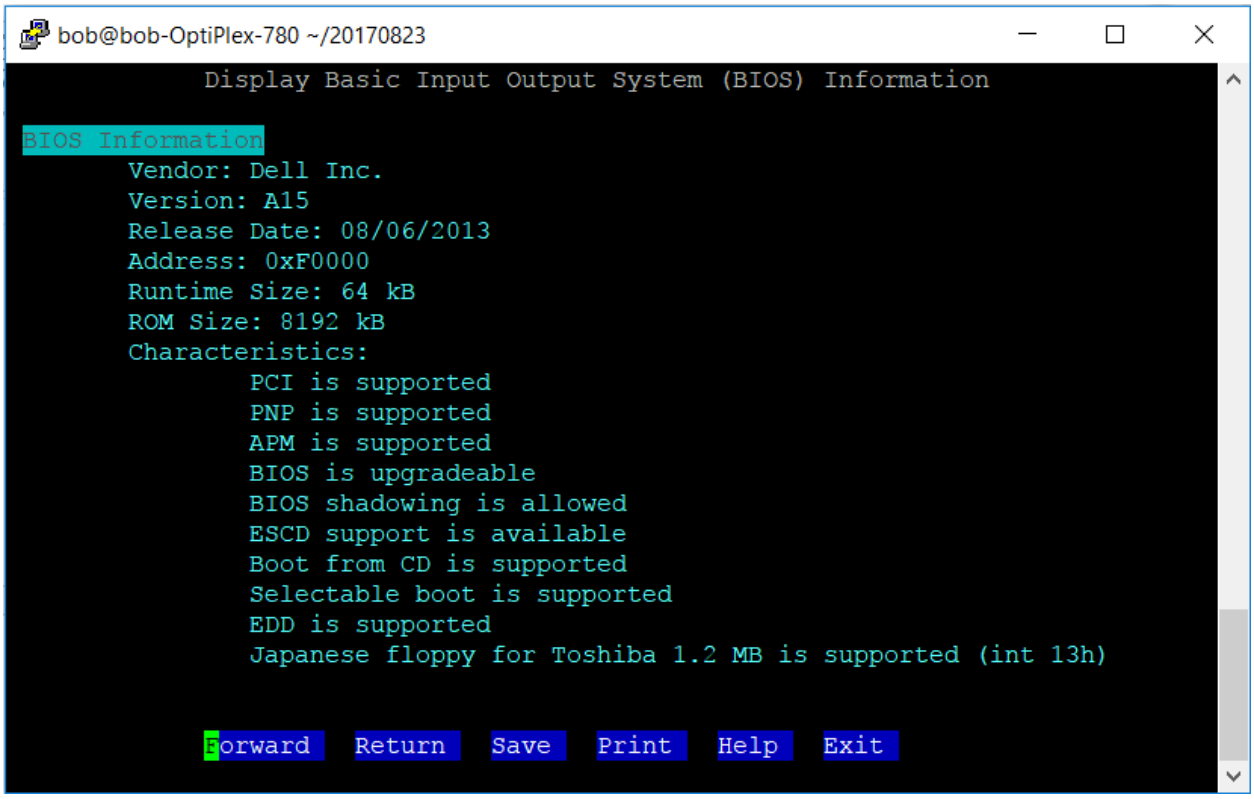


Figure 12: Display Basic Input Output System (BIOS) Information.

BIOS (an acronym for **Basic Input/Output System**) is non-volatile firmware used to perform hardware initialization during the booting process, and to provide runtime services for operating systems and programs. The **BIOS** firmware comes pre-installed on a computer system board, and it is the first software run when powered on. The **BIOS** in modern PCs initializes and tests the system hardware components, and loads a boot loader or operating system from a mass memory device. Most **BIOS** implementations are specifically designed to work with a particular computer or motherboard model, by interfacing with various devices that make up the complementary system chipset.

Unified Extensible Firmware Interface (**UEFI**) is the successor to **BIOS**. **UEFI** also works with a new partitioning scheme called **GUID Partition Table** (**GPT**). Support for **GPT** in Linux is enabled by turning on the option **CONFIG_EFI_PARTITION** during kernel configuration. This option allows Linux to recognize and use **GPT** disks after the system firmware passes control to Linux.

Display Block Devices Information

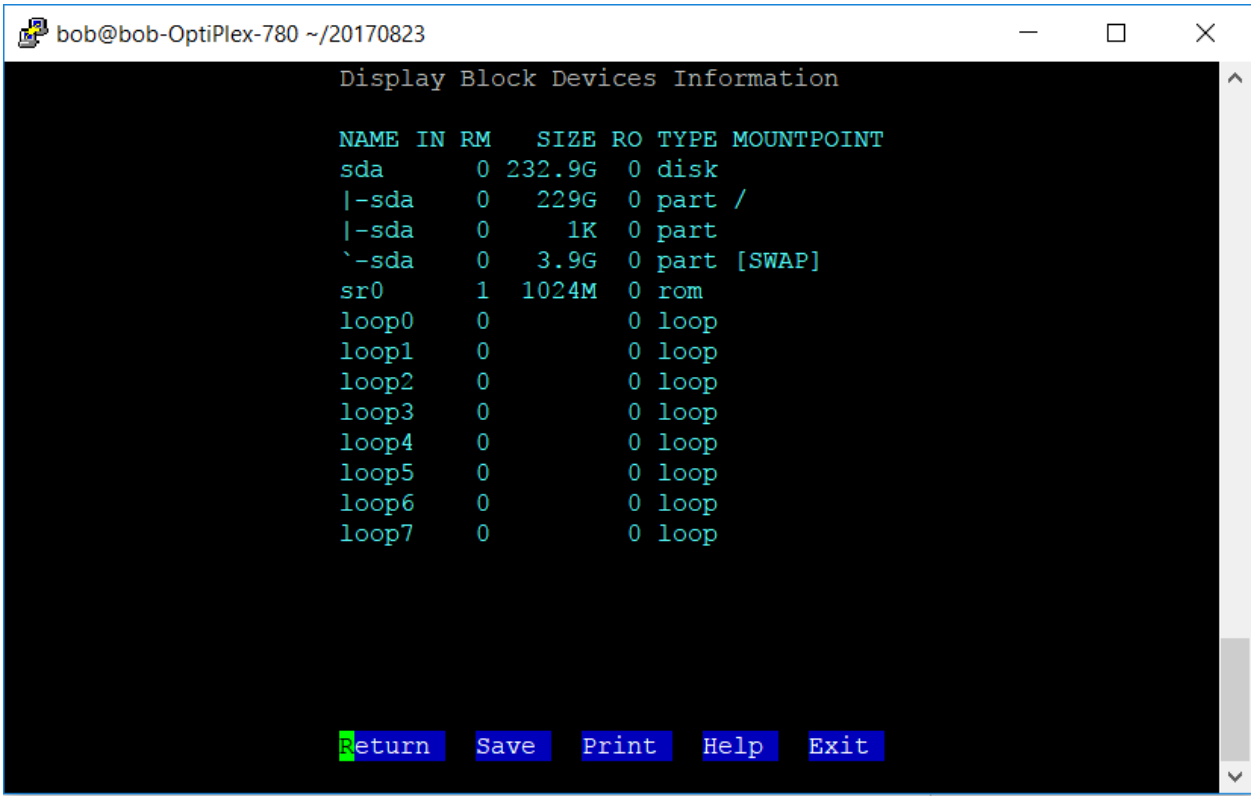


Figure 13: Display Block Devices Information.

A **Block Device** is a computer data storage device that supports reading and (optionally) writing data in fixed-size blocks, sectors, or clusters. These blocks are generally 512 bytes or a multiple thereof in size.

This panel displays information about all available **Block Devices**. It includes all **Block Device**s (except **RAM** disks) in a tree-like format. Here is a description of the output columns:

NAME:	The device name.
SIZE:	The size of the device.
RM:	Whether the device is removable. (0 = No, 1 = Yes).
RO:	Whether the device is read-only. (0 = No, 1 = Yes).
TYPE:	Type of device.
MOUNTPOINT:	Mount point for device.

Display Disk Partition Information

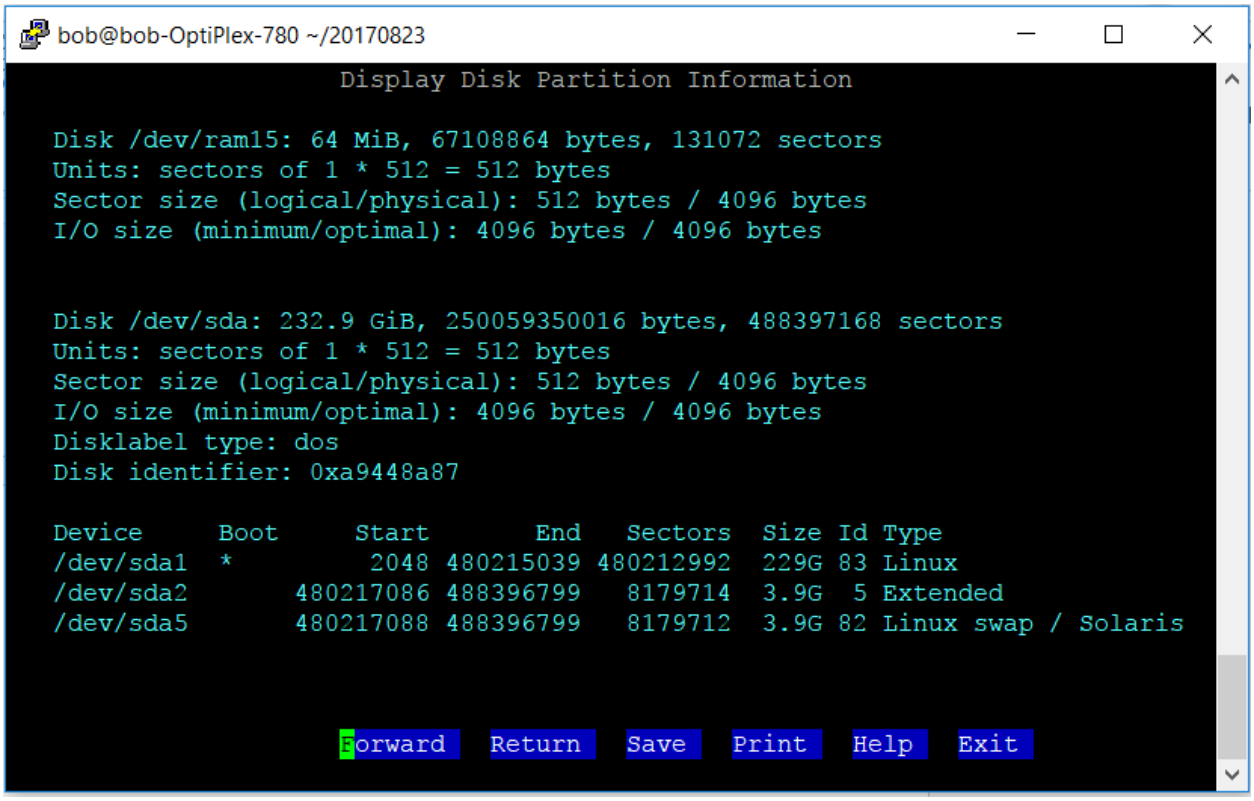


Figure 14: Display Disk Partition Information.

A hard disk can be divided into several **partitions**. Each **partition** functions as if it were a separate hard disk. This panel displays information about all available **Disks** along with any **Partitions**.

Here is a description of the output columns for Disk Partitions:

Device: Name of the **Partition**. This is typically the name of the Disk with a suffix of 1, 2, 3, etc.
Boot: Whether this is a bootable **Partition**. * indicates Yes.
Start: Starting sector.
End: Ending sector.
Sectors: Number of sectors.
Size: Size of the **Partition**.
Id: ID of the **Partition**.
Type: Type of **Partition**.

Display Memory Setup Information

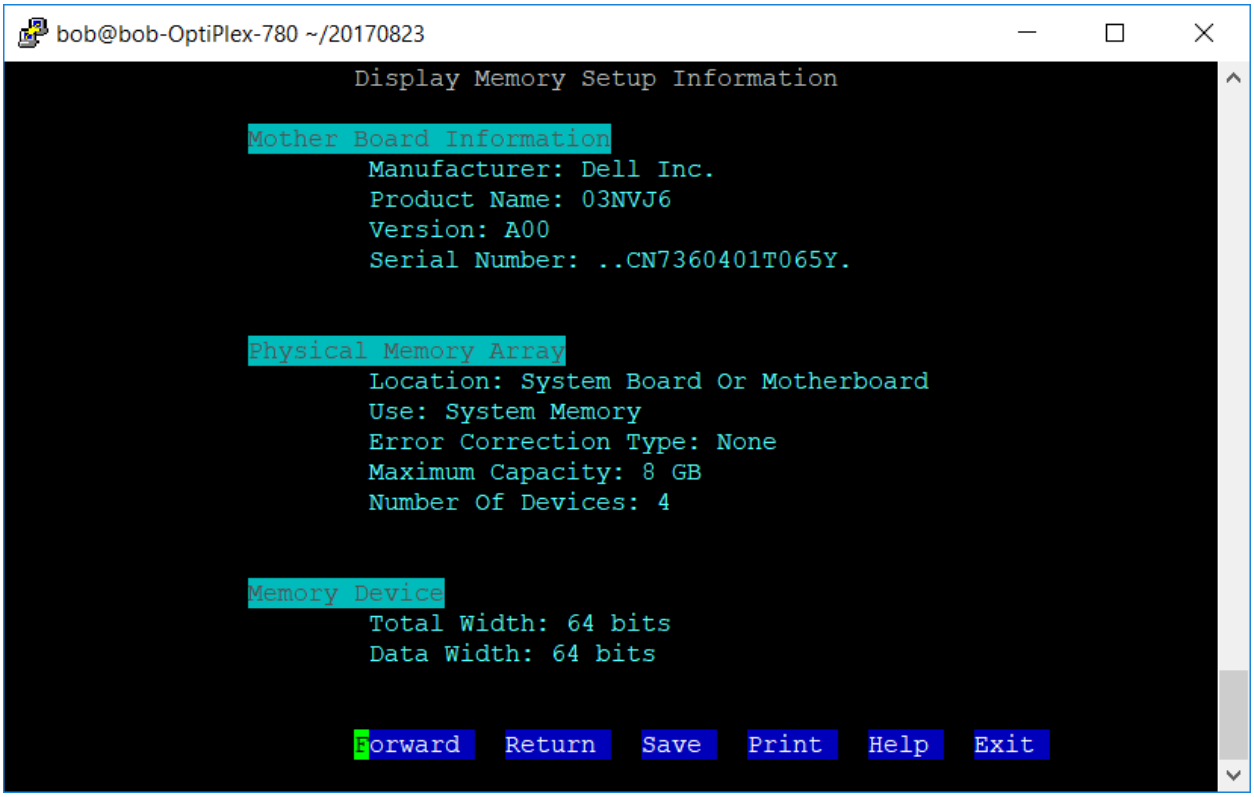


Figure 15: Display Memory Setup Information.

The two widely used forms of modern **Random Access Memory** (**RAM**) are static **RAM** (**SRAM**) and dynamic **RAM** (**DRAM**). **SRAM** is more expensive to produce, but is generally faster and requires less dynamic power than **DRAM**. In modern computers, **SRAM** is often used as cache memory.

Because **DRAM** is less expensive to produce than static **RAM**, it is the predominant form of computer memory used in modern computers.

Synchronous dynamic random-access memory (**SDRAM**) is any dynamic random-access memory (**DRAM**) where the operation of its external pin interface is coordinated by an externally supplied clock signal. **SDRAM** is currently widely used for computer memory. Beyond the original **SDRAM**, further generations of double data rate **RAM** have entered the mass market - **DDR** (also known as **DDR1**), **DDR2**, **DDR3**, **DDR4**, and **DDR5**.

Computer memory (**RAM**) is stored in **Memory Devices** in **Physical Memory arrays** on a **Motherboard**.

Display Network Configuration

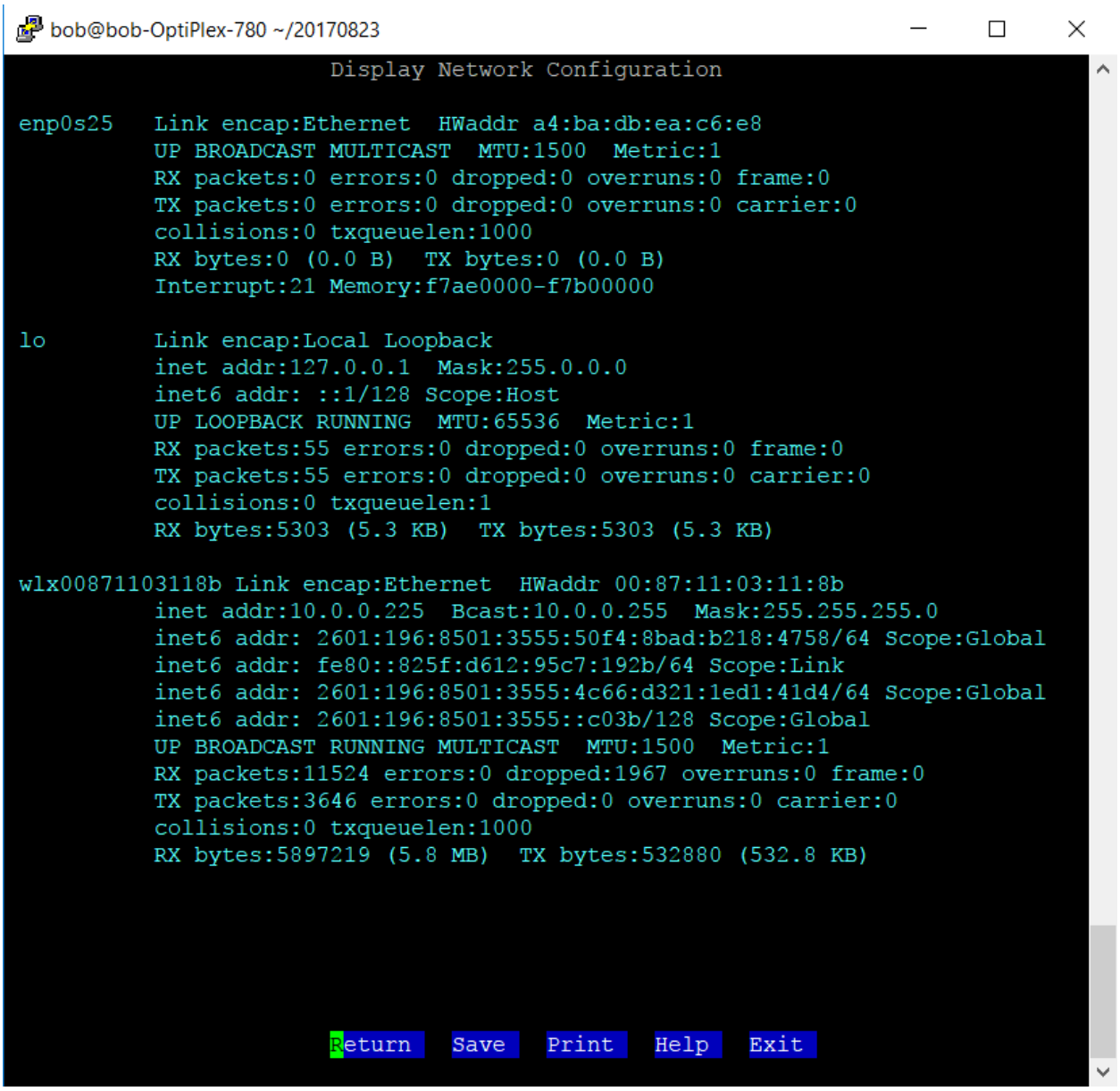


Figure 16: Display Network Configuration.

This panel displays shows details of the **network interfaces** that are up and running in your computer. Here is a description of some of the output columns:

Link encap:Ethernet:	The interface is an Ethernet related device.
HWaddr:	This is the hardware address or MAC address.
inet addr:	This system's IPv4 address.
Mask:	The network mask.
UP:	Ethernet interface Kernel modules are loaded.
BROADCAST:	The Ethernet device supports broadcasting.
RUNNING:	The interface is ready to accept data.

Display System Hardware Submenu

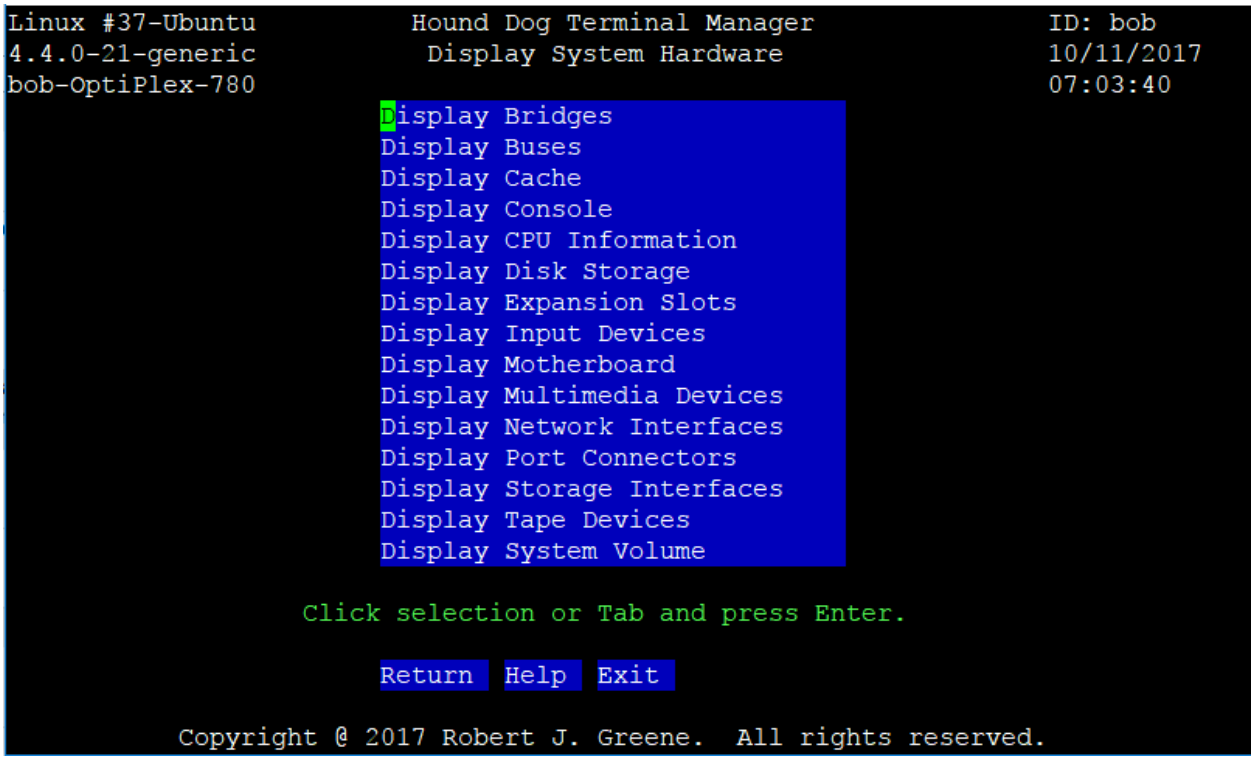


Figure 17: Display System Hardware Submenu.

This Submenu displays various hardware components of this system:

Display Bridges:	Bridges connect buses with different capacities.
Display Buses:	Transfers data between computer components.
Display Cache:	High-speed computer memory.
Display Connectors:	Connectors such as **USB** and parallel ports.
Display Console:	Computer display components.
Display CPU:	Computer processor.
Display Slots:	Expansion slots.
Display Input Devices:	Input components such as keyboard and mouse.
Display Motherboard:	Printed circuit board.
Display Multimedia:	Devices such as audio, video, and Text devices.
Display Networking:	Networking components.
Display Storage:	Storage components and Interfaces.
Display Tapes:	Attached Tape devices.
Display System Volume:	Storage volume system the system filesystem is based upon.

Display System Bridge Information

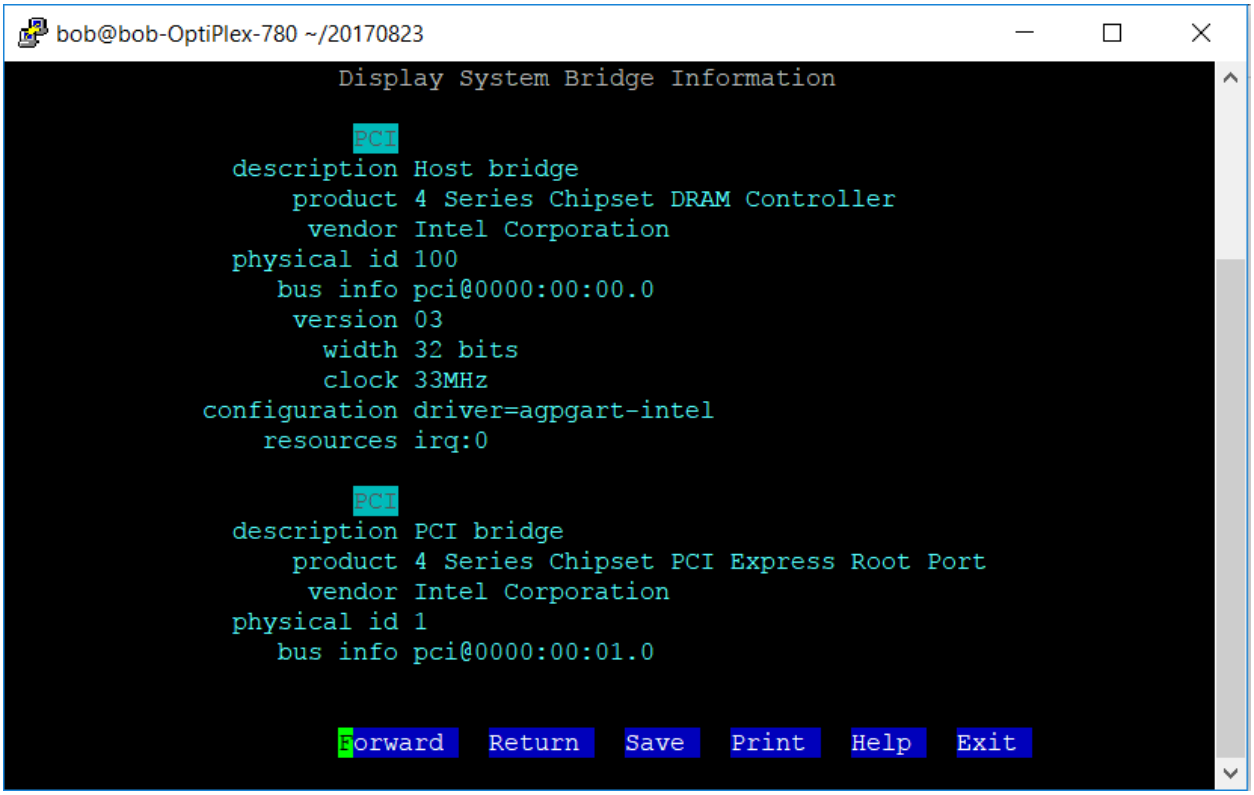


Figure 18: Display System Bridge Information.

A **Bridge** connects two **Buses** with different capacities (bandwidths) inside a computer.

A **Bus** is a communication system that transfers data between computer components. The entire **Bus** System starts close to the **CPU**, where the traffic is greatest.

A **Northbridge** or **Host Bridge** is one of the two chips in the core logic chipset architecture on a PC motherboard, the other being the **Southbridge**. Unlike the **Southbridge**, **Northbridge** is connected directly to the **CPU** via the **Front-Side Bus** (**FSB**) and is thus responsible for tasks that require the highest performance, such as the flow of data between the **CPU** and **RAM**, and to the **Accelerated Graphics Port** (**AGP**).

Southbridge is an Intel chipset that manages the basic forms **of Input/Output** (**I/O**) such as **Serial**, **Universal Serial Bus** (**USB**), **Integrated Drive Electronics** (**IDE**), **Audio**, and **Industry Standard Architecture** (**ISA**) **I/O** in a computer.

Display System Bus Information

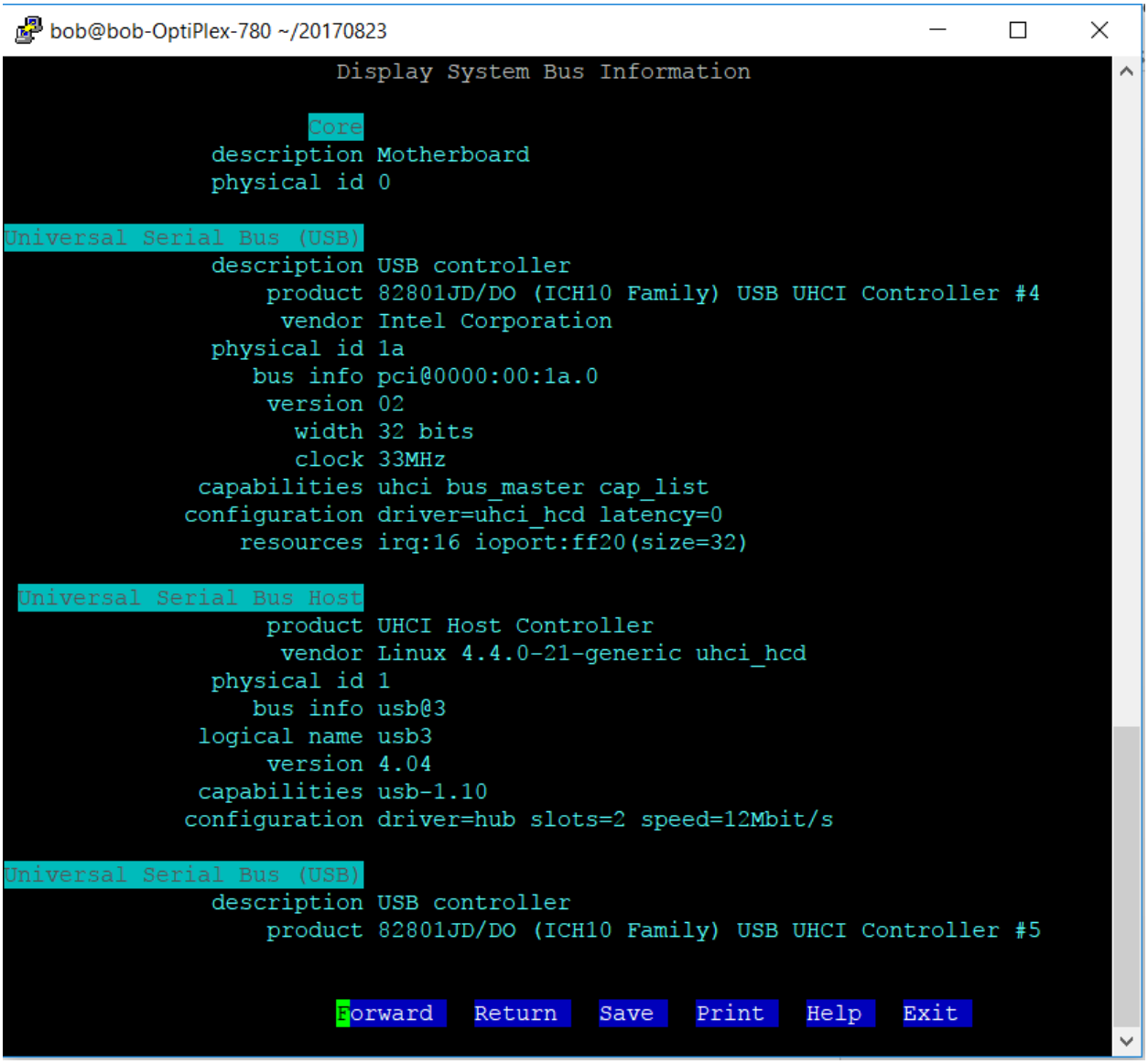


Figure 19: Display System Bus Information.

A **Bus** is a communication system that transfers data between components inside a computer. It is these **Buses** which connect all of the components to each other on a **Motherboard**.

The entire **Bus** System starts close to the **CPU**, where the traffic is greatest. **RAM** is the component which has the very greatest data traffic, and is therefore connected directly to the CPU by a particularly powerful **Bus** that is called the **Front Side Bus** (**FSB**).

The **Buses** connecting the motherboard to the **PC** peripheral devices are called **I/O Buses**. Data packets (of 8, 16, 32, 64 or more bits at a time) are constantly being moved back and forth between the **CPU** and all the other components. There are several **Buses** on a **Motherboard** but they are all connected, so that data can run from one to another and reach all of the **Motherboard** components.

Display System Cache Information

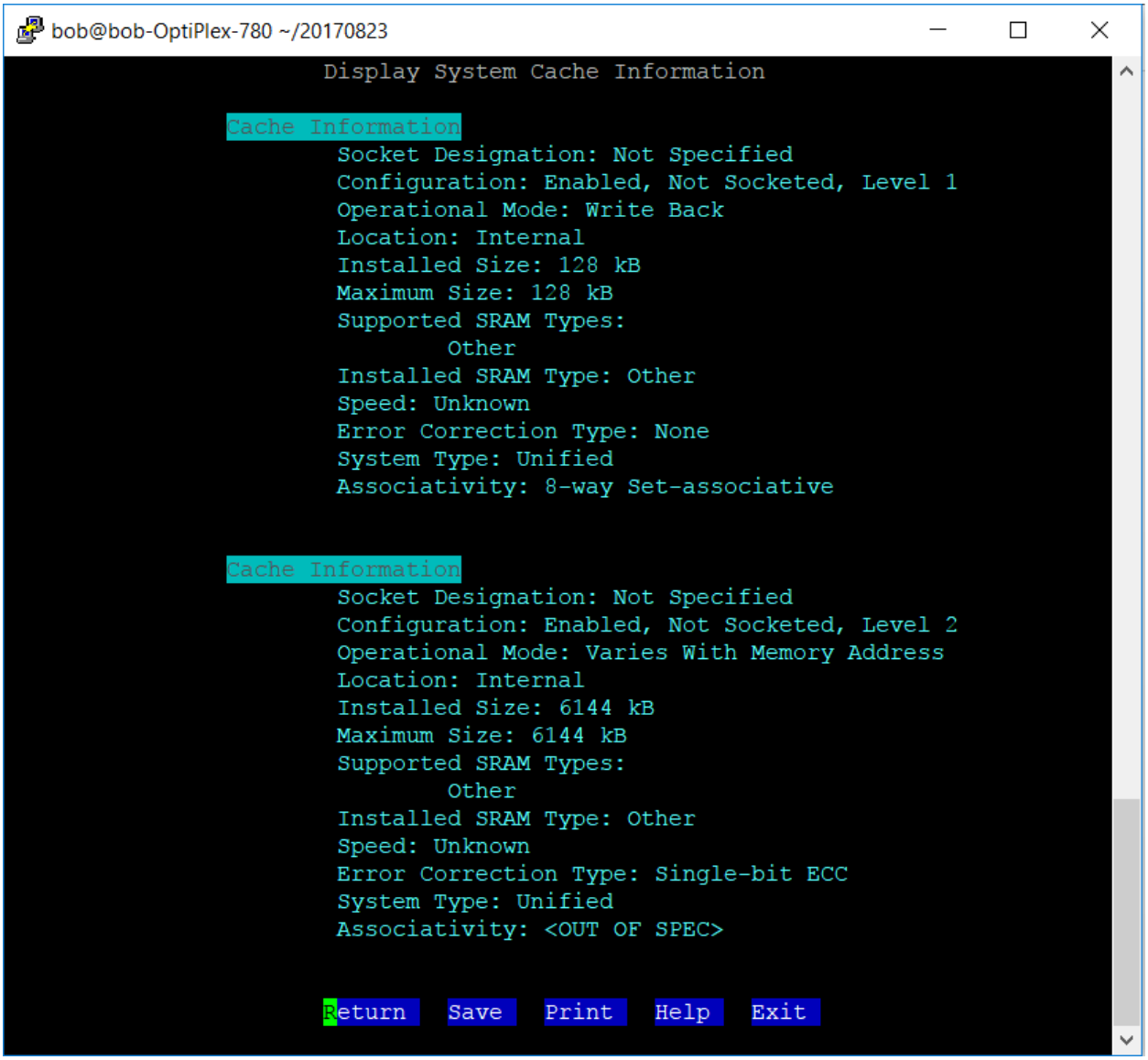


Figure 20: Display System Cache Information.

A **Memory Cache**, sometimes called a cache store or **RAM** cache, is a portion of memory often made of high-speed static **RAM** (**SRAM**) instead of the slower and cheaper dynamic **RAM** (**DRAM**) used for main memory.

In **SRAM**, a bit of data is stored using the state of a six transistor memory cell. This form of **RAM** is more expensive to produce, but is generally faster and requires less dynamic power than **DRAM**.

Memory caching is effective because most programs access the same data or instructions over and over. By keeping as much of this information as possible in **SRAM**, the computer avoids accessing the slower **DRAM**.

Display System Display Information

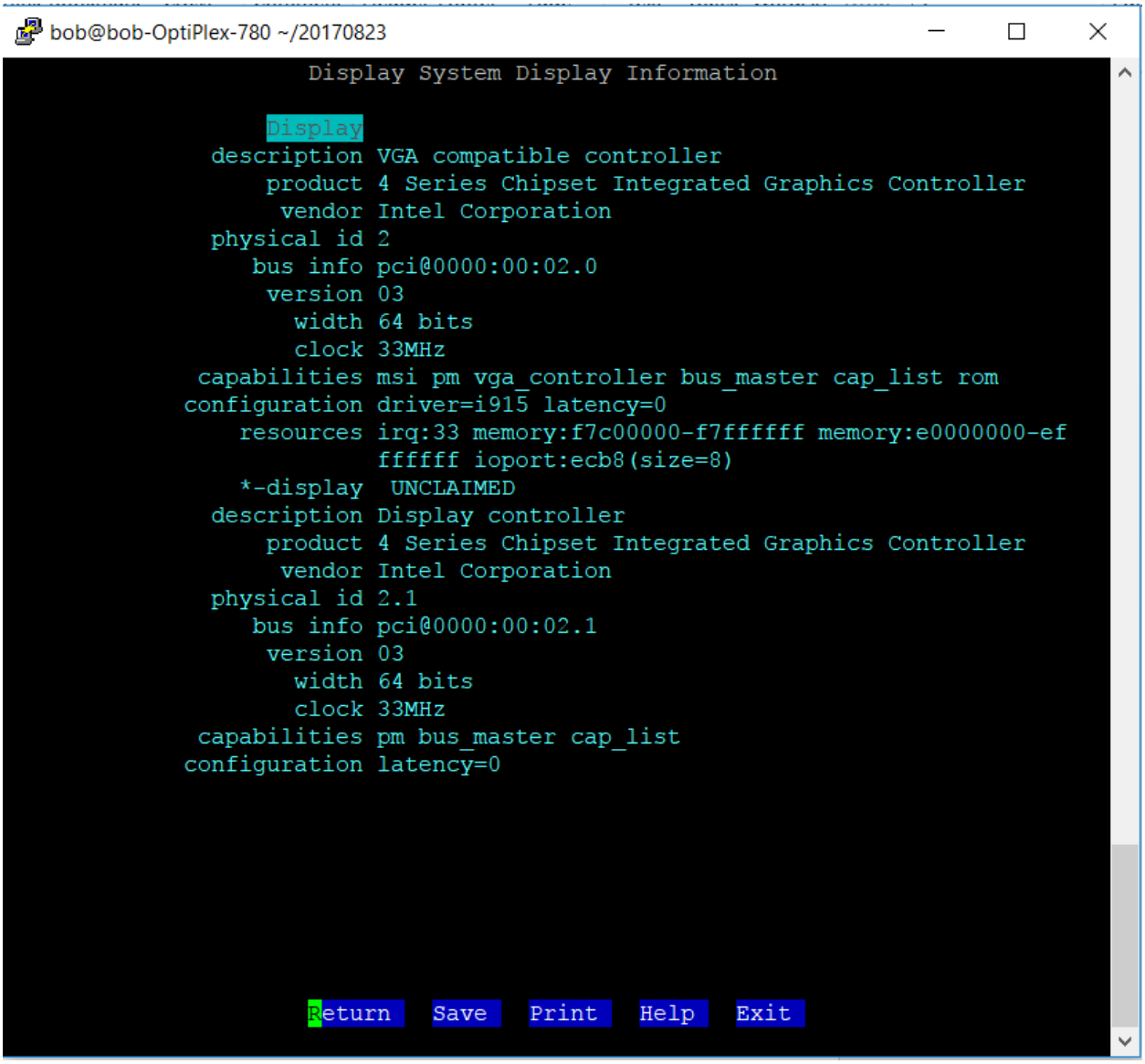


Figure 21: Display System Display Information.

The **Linux Console** is a **System Console** internal to the **Linux Kernel**. A **System Console** is the device which receives all **Kernel** messages and warnings and which allows logins in single user mode. The **Linux Console** provides a way for the kernel and other processes to send text output to the user, and to receive text input from the user. The user typically enters text with a **Computer Keyboard** and reads the output text on a **Computer Monitor**.

The **Linux Kernel** supports virtual consoles - consoles that are logically separate, but which access the same physical keyboard and display. The **Linux Console** (and **Linux Virtual Consoles**) are implemented by the **VT Subsystem** of the **Linux Kernel**, and do not rely on user space software. This is in contrast to a **Terminal Emulator**, which is a user space process that emulates a terminal, and is typically used in a graphical display environment.

Display System CPU Information

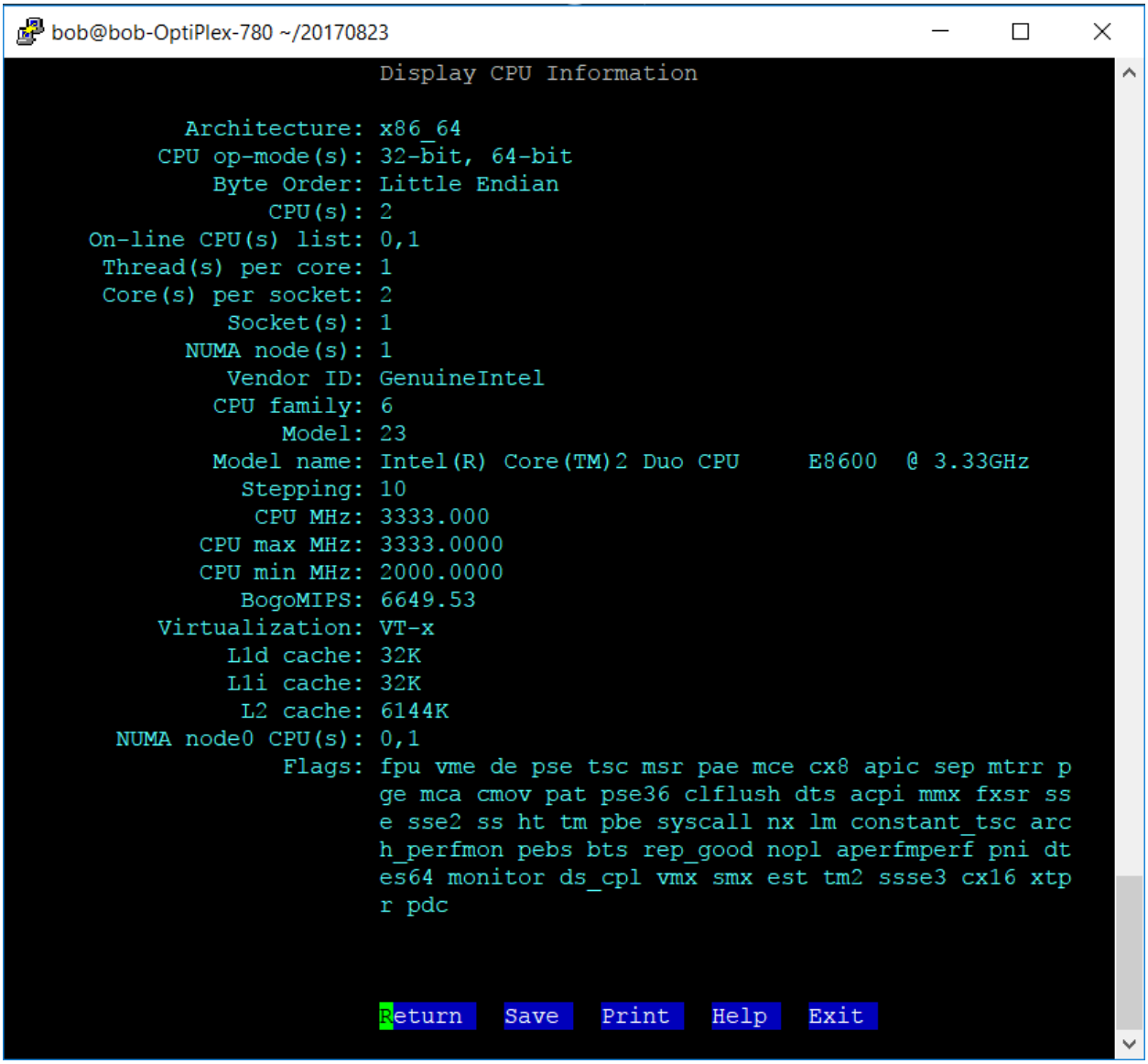


Figure 22: Display System CPU Information.

A **Central Processing Unit** (**CPU**) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control, and **Input/Output** (**I/O**) operations specified by the instructions.

Principal components of a **CPU** include the **Arithmetic Logic Unit**, (**ALU**), that performs arithmetic and logic operations, **Processor Registers** that supply operands to the **ALU** and store the results of **ALU** operations, and a **Control Unit** that performs the fetching (from memory) and execution of instructions by directing the coordinated operations of the **ALU**, **Registers**, and other components.

Most modern **CPUs** are **microprocessors**, meaning they are contained on a single integrated circuit chip. Many computers contain a **multi-core processor**- a chip containing multiple **CPUs** called **Cores**.

Display Disk Storage Information

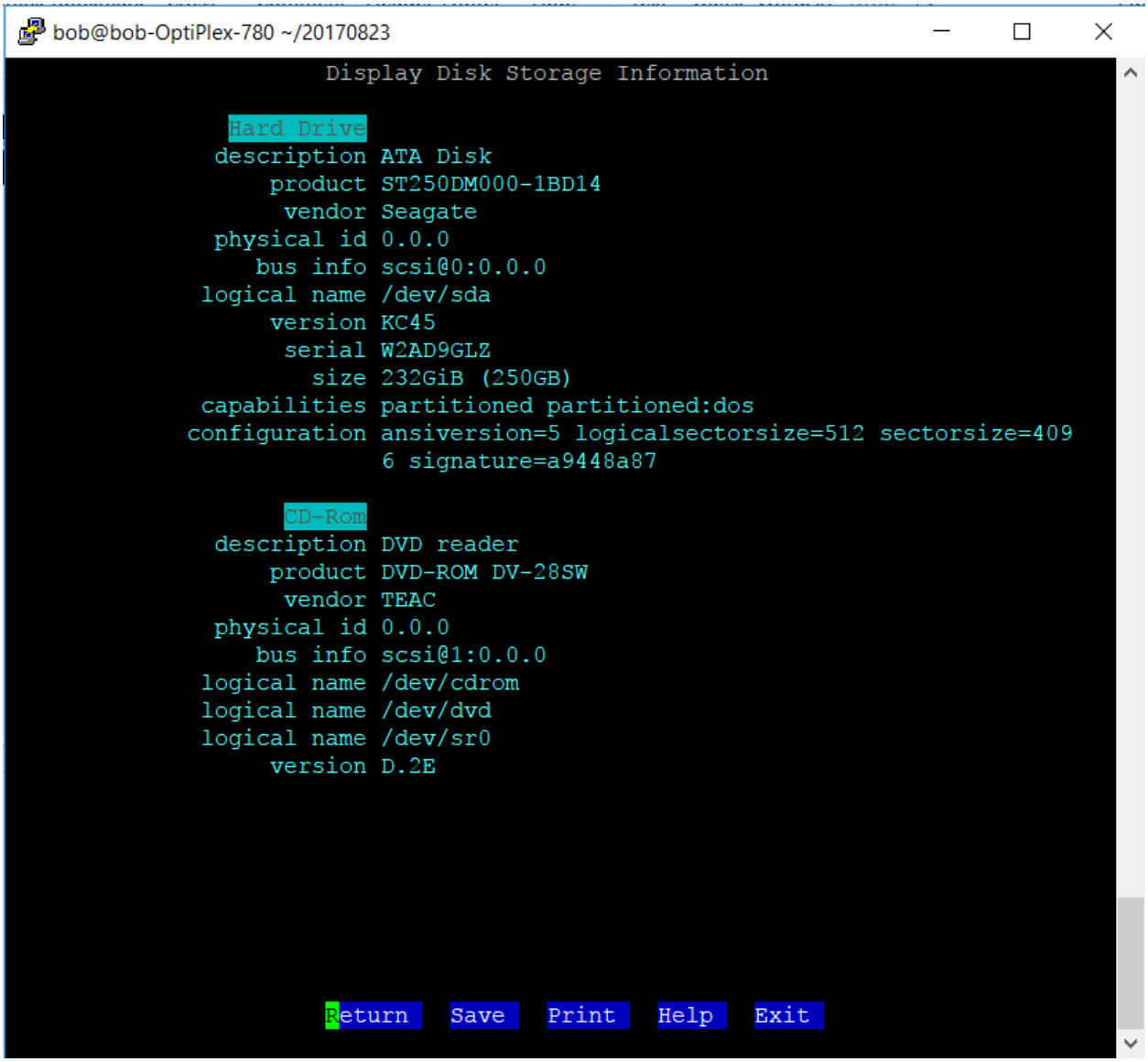


Figure 23: Display Disk Storage Information.

Disk Storage is a general category of storage mechanisms where data are recorded by various electronic, magnetic, optical, or mechanical changes to a surface layer of one or more rotating disks. Notable disk drive types are the **Hard Disk Drive** (**HDD**), the **Floppy Disk Drive** (**FDD**), and various **Optical Disc Drives**.

Many modern Computers contain **Solid-State Drives** which are solid-state **Storage Devices** using Integrated circuit assemblies as memory to store data persistently. **SSD** technology primarily uses electronic interfaces compatible with traditional **Block input/output** (**I/O**) hard disk drives, which permit simple replacements in common applications. New **I/O** interfaces like **SATA Express** and **M.2** have been designed to address specific requirements of the **SSD** technology.

Display Expansion Slot Information

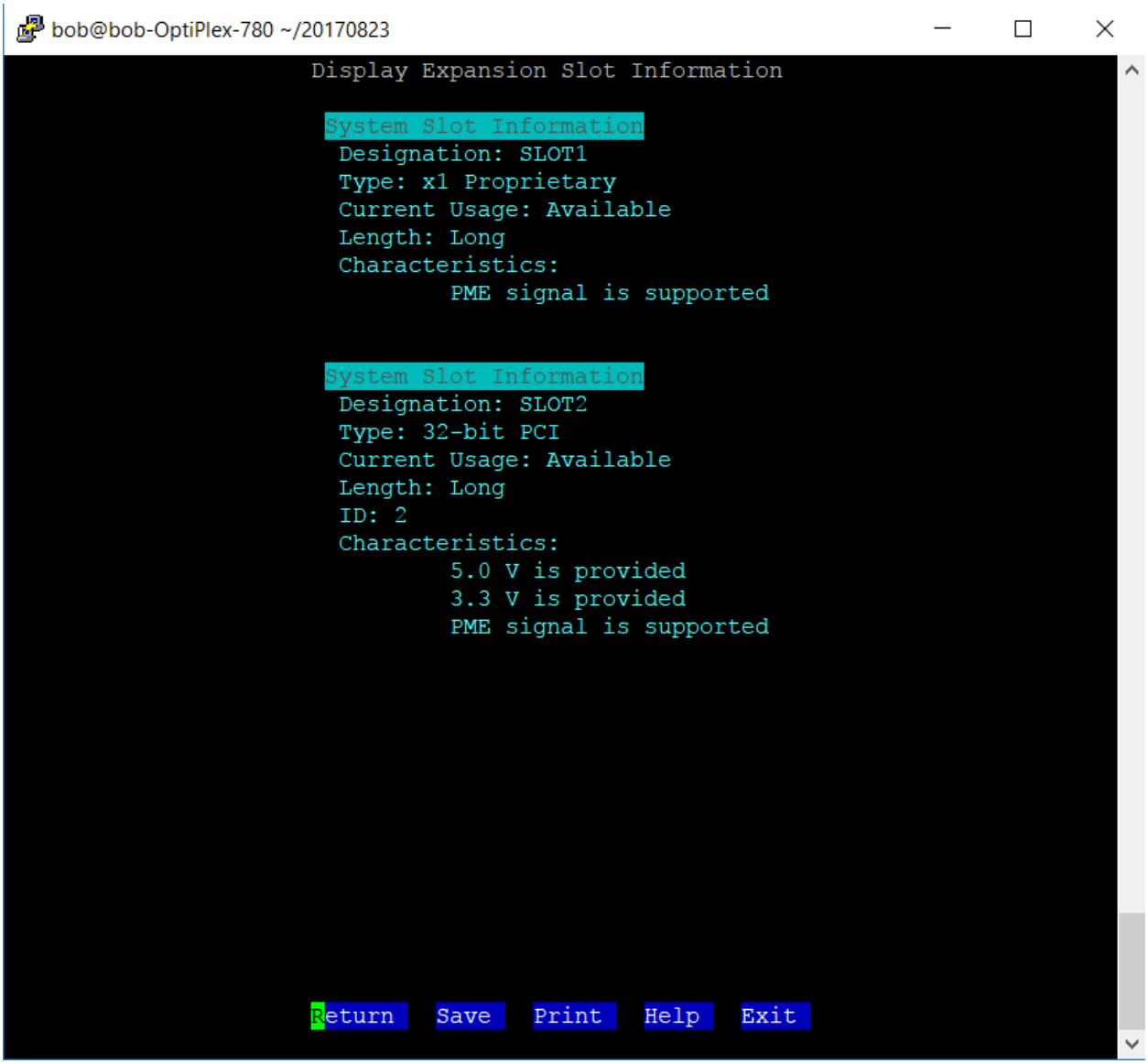


Figure 24: Display Expansion Slot Information.

An **Expansion slot** is a connection or port located inside a computer on the **motherboard** or **riser board** that allows a computer hardware **expansion card** to be connected.

An **expansion card**, **expansion board, adapter card** or **accessory card**, is a **printed circuit board** that can be inserted into an **electrical connector**, **or expansion slot** on a computer **motherboard**, **backplane,** or **riser card** to add functionality to a computer system using the **expansion bus**.

Some examples of **Expansion cards** include **Network cards**, **Sound cards**, **Video cards**, and **Interface cards** for **ATA**, **Bluetooth**, **EIDE**, **Firewire**, **IDE**, **Parallel**, **RAID**, **SCSI**, **Serial**, **USB**, **Fibre Channel**, and **FICON**.

Display System Input Information

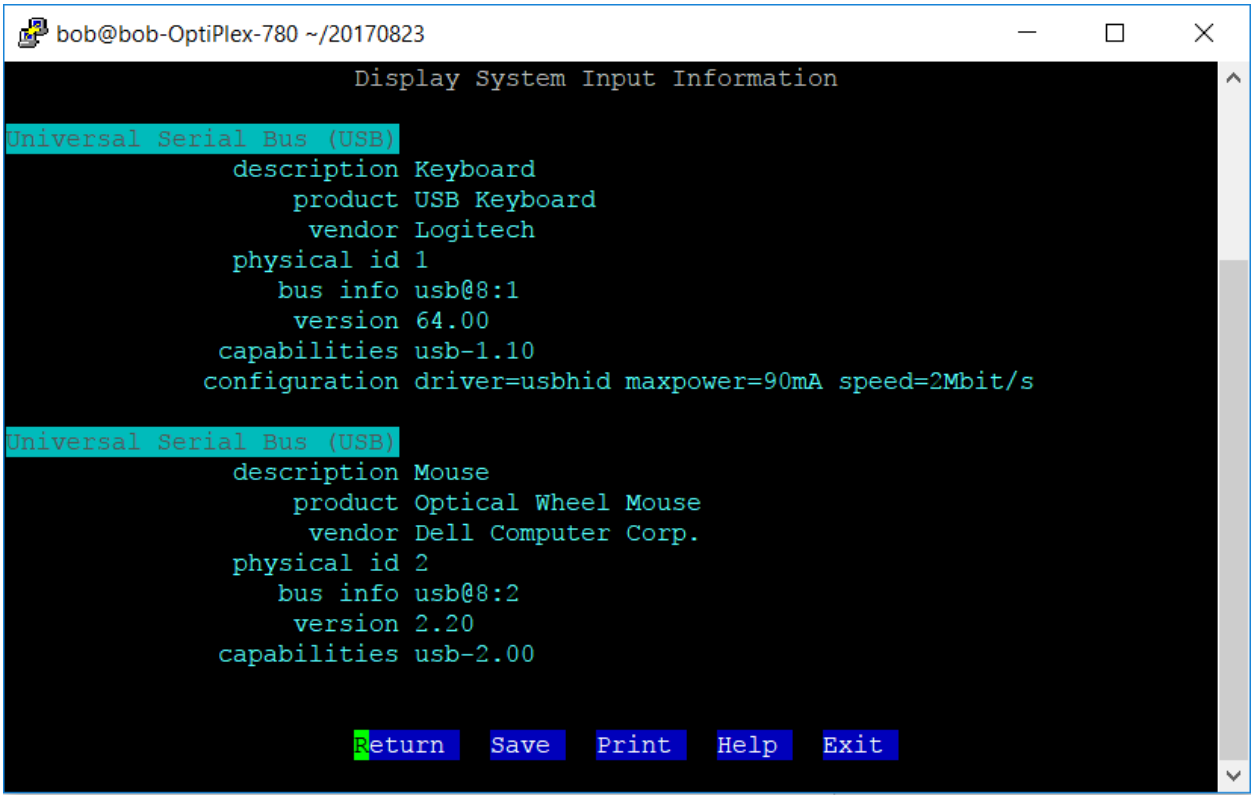


Figure 25: Display System Input Information.

In computing, an **input device** is a piece of computer hardware equipment used to provide data and control signals to an information processing system such as a computer or information appliance.

Examples of input devices include a **Keyboard**, **Mouse**, **Scanner**, **Digital Camera**, and **Joystick**.

Display System Motherboard Information

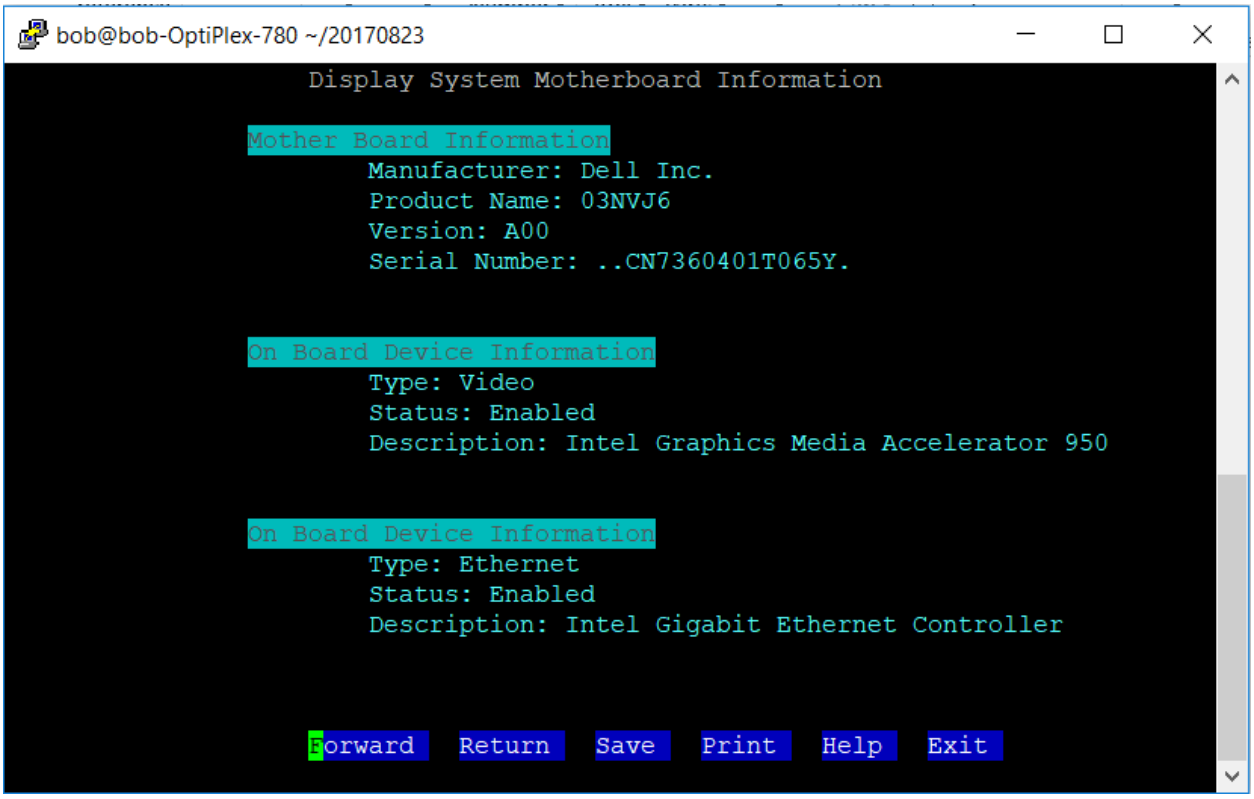


Figure 26: Display System Motherboard Information.

A **Motherboard** is the main **Printed Circuit Board** (**PCB**) found in general purpose **Microcomputers** and other expandable systems. It holds and allows communication between many of the crucial electronic components of a system, such as the **Central Processing Unit** (**CPU**) and **Memory**, and provides connectors for other peripherals.

A **Motherboard** usually contains significant Subsystems such as the **Central Processor**, the **Chipset's Input/Output** and **Memory Controllers**, **Interface Connectors**, and other components integrated for general purpose use.

An important component of a **motherboard** is the **Microprocessor's** supporting **Chipset**, which provides the supporting interfaces between the **CPU** and the various **Buses** and external components. This **Chipset** determines the features and capabilities of the **Motherboard**.

Display Multimedia Information

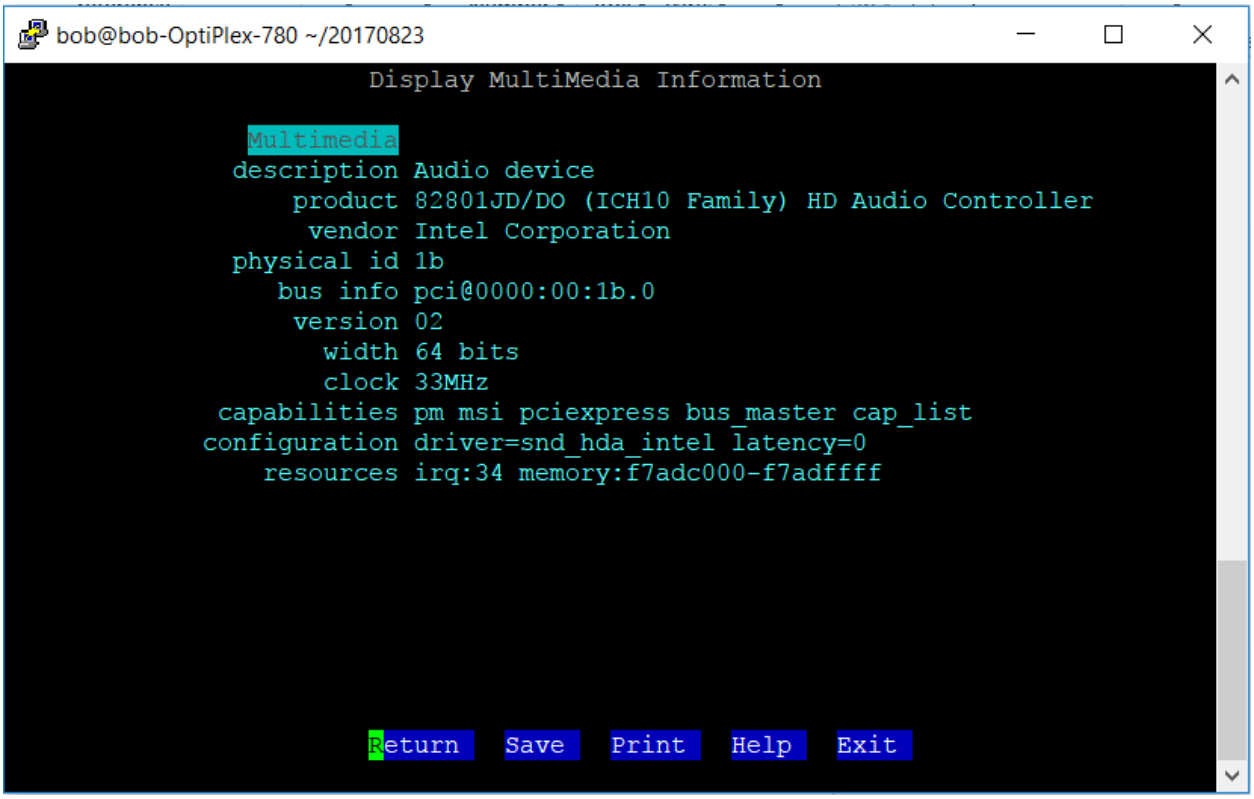


Figure 27: Display Multimedia Information.

Multimedia Devices are electronic media devices used to store and experience multimedia content.

Multimedia is content that uses a combination of different content forms such as text, audio, images, animations, video and interactive content.

Multimedia contrasts with media that use only rudimentary computer displays such as text-only or traditional forms of printed or hand-produced material.

Types of **Multimedia Devices** include **Microphones**, **Speakers**, **Cameras**, and **Headphones**.

Display Network Interfaces

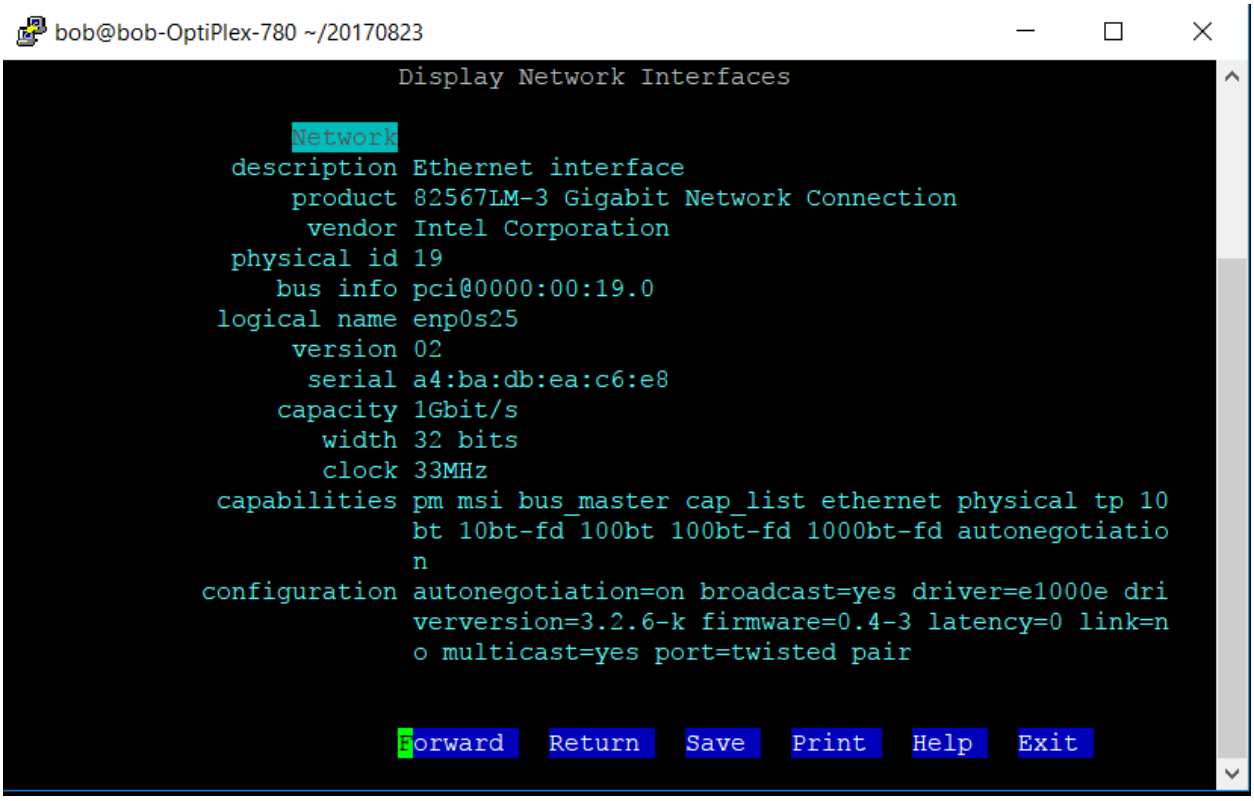


Figure 28: Display Network Interfaces.

A **Network Interface Controller** is a computer hardware component that connects a computer to a computer network. Modern **Network Interface Controllers** offer advanced features such as interrupt and **DMA** interfaces to the host **Processors**, support for multiple receive and transmit queues, partitioning into multiple logical interfaces, and on-controller network traffic processing such as the **TCP Offload Engine** (**TOE**).

The **Network Controller** implements the electronic circuitry required to communicate using a specific physical layer and data link layer standard such as **Ethernet**, **Fibre Channel**, or **Wi-Fi**. This provides a base for a full network protocol stack, allowing communication among small groups of computers on the same local area network and large-scale network communications through routable protocols, such as **Internet Protocol** (**IP**).

Display System Port Connectors Information

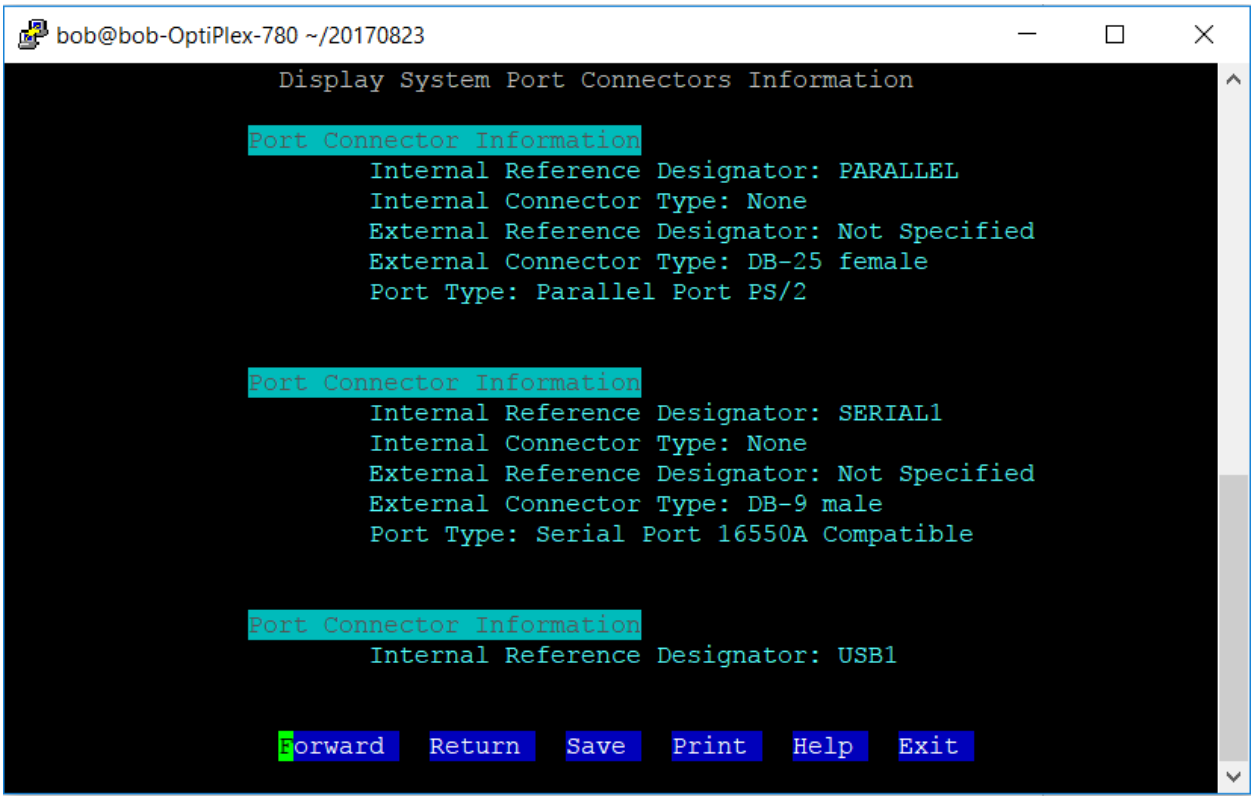


Figure 29: Display System Port Connectors Information.

A **Port Connector** serves as an interface between the computer and other computers or peripheral devices. In computer terms, a **Port** generally refers to the female part of connection.

Computer Ports have many uses, to connect a **Monitor**, **Webcam**, **Speakers**, or other peripheral devices.

On the physical layer, a computer **Port** is a specialized outlet on a piece of equipment to which a plug or cable connects.

Electronically, the several conductors where the **Port** and cable contacts connect, provide a method to transfer signals between devices.

Display Storage Interfaces

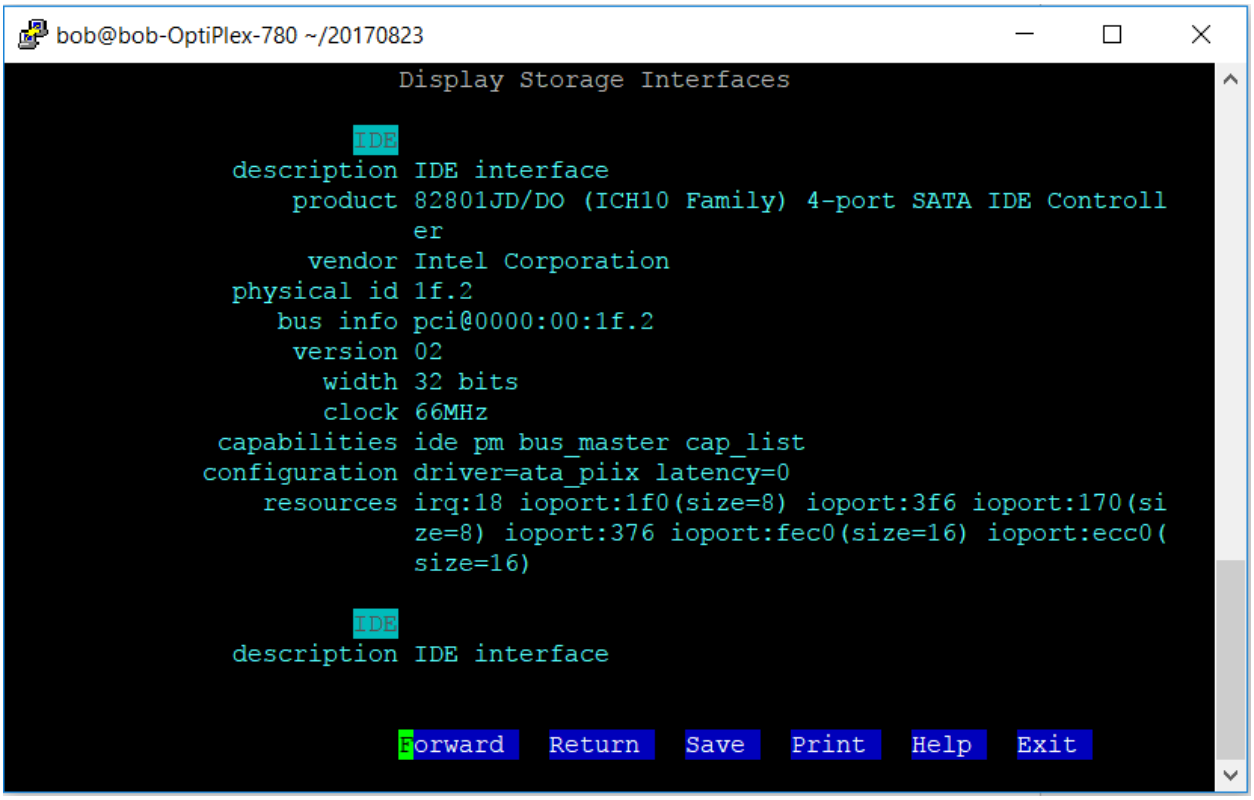


Figure 30: Display Storage Interfaces.

Hard Disk Drives are accessed over one of a number of **Storage Interfaces**, including parallel **ATA** (**PATA**, also called **IDE** or **EIDE**; described before the introduction of **SATA** as **ATA**), **Serial ATA** (**SATA**), **SCSI**, **Serial Attached SCSI** (**SAS**), and **Fibre Channel**.

Bridge circuitry is sometimes used to connect **Hard Disk Drives** to **Buses** with which they cannot communicate natively, such as **IEEE 1394**, **USB**, and **SCSI**.

Solid State Drives typically use storage interfaces such as **SATA**, **SAS**, or **Fibre Channel**. **Solid State Card** (**SSC**) **Solid State Storage** that resides on a printed circuit board may utilize a standard card form factor such as a **PCI** card and use an interface such as **PCIe**.

Display Attached Tape Drives

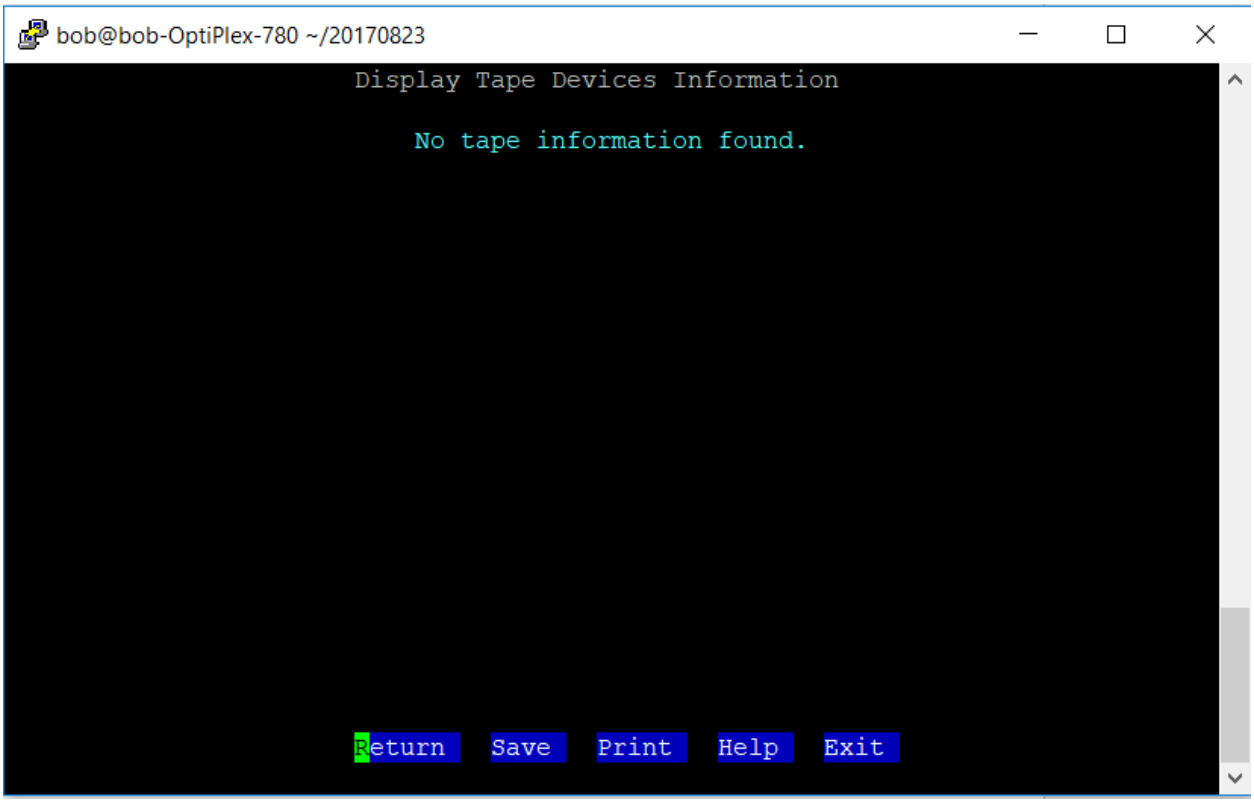


Figure 31: Display attached Tape Drives.

A **Tape Drive** is a data storage device that reads and writes data on a magnetic tape. Magnetic tape data storage is typically used for offline, archival data storage. Tape media generally has a favorable unit cost and a long archival stability.

A **Tape Drive** provides sequential access storage, unlike a **Hard Disk Drive**, which provides direct access storage. A **Disk Drive** can move to any position on the disk in a few milliseconds, but a Tape Drive must rewind to the start of data to read at random positions. As a result, **Tape Drives** have very slow average seek times but can stream data quickly when the required position has been reached.

SCSI Tape Device names begin with **/dev/st** or **/dev/nst**, while **IDE Tape Device** names begin with **/dev/ht** or **/dev/nht**, depending upon whether **auto rewind** is supported. First device has a suffix of 1 with subsequent devices numbered accordingly.

Display System Volume Information

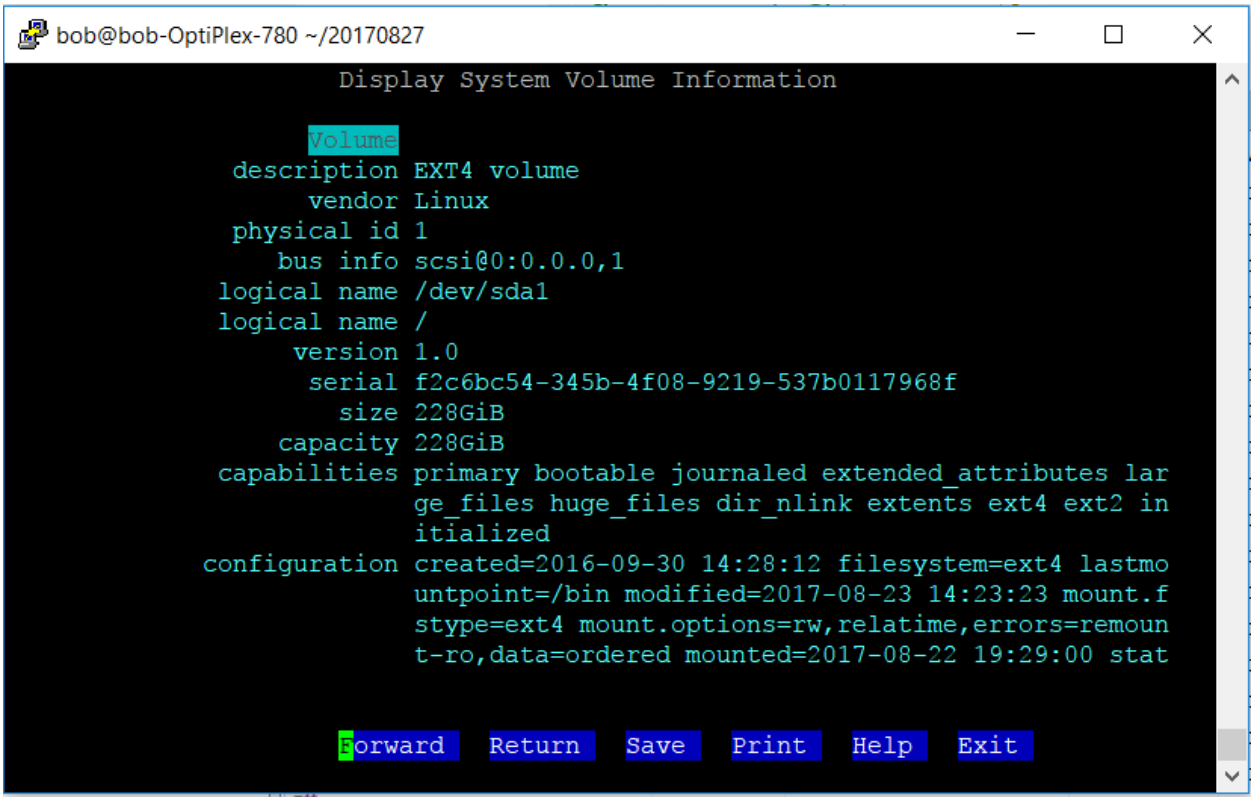


Figure 32: Display System Volume Information.

A **System Volume** or logical drive is a single accessible storage area with a single file system, typically (though not necessarily) resident on a single partition of a hard disk.

Although a **System Volume** might be different from a physical disk drive, it can still be accessed with an operating system's logical interface.

Volumes other than the boot volume have a mount-point somewhere within the **filesystem**, represented by a path. Logically, the directory tree stored on the volume is grafted in at the **mountpoint**.

By convention, mount-points will often be placed in a directory called **'/mnt'**, though **'/media'** and other terms are sometimes used.

To use a given path as a mount-point for another volume, an empty directory (sometimes called a folder) must exist there.

Linux operating systems use the **mount** command to manipulate mount points for volumes.

Manage Preferences

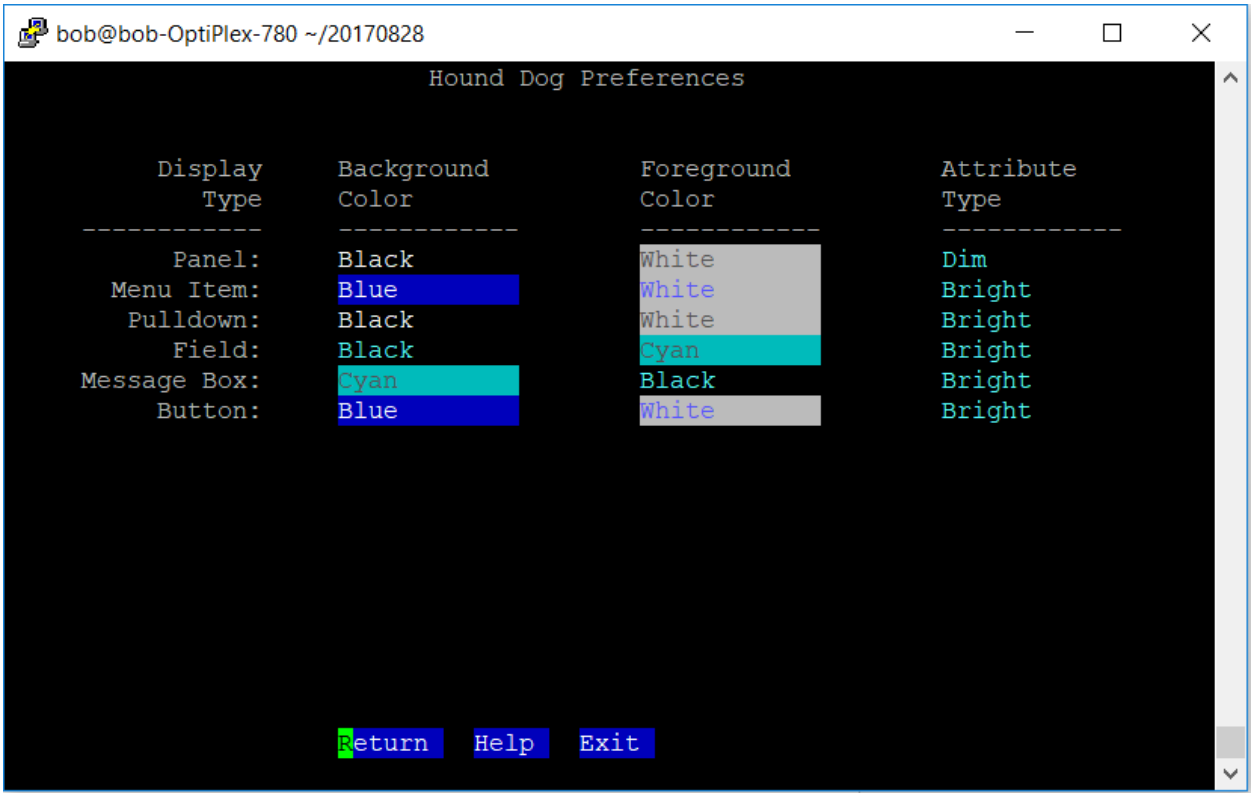
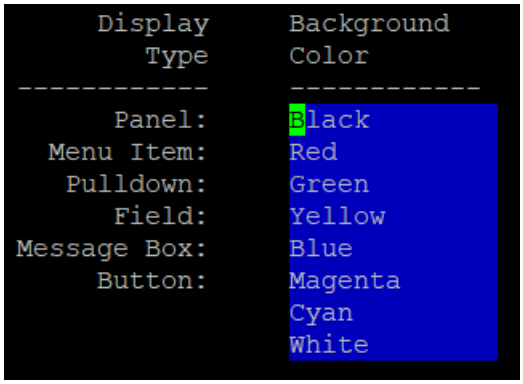


Figure 33: Manage Preferences.

This panel allows for customization of settings. This is currently limited to default colors and attributes.

To Change a preference, **Click** or **Tab** and Press **Enter** to select the Preference value you which to change.
Select a choice and **Click** or press **Enter** to complete.

Here is a snapshot of the pulldown presented when you select the Background color for the Display Type **Panel**. →



Colors and Attributes can be set for:

Panel:	Panel Background and literals.
Menu Item:	Items that appear on menu.
Pulldown:	Choices that appear in Pulldowns.
Field:	Items that have informational content.
Button:	Push Buttons.
Message Box:	Message notifications.

Colors:	White, Red, Blue, Cyan, Green, Black, Magenta, Yellow.
Attributes:	Bright, Dim, Underline, Reverse Video.

Display Product Information

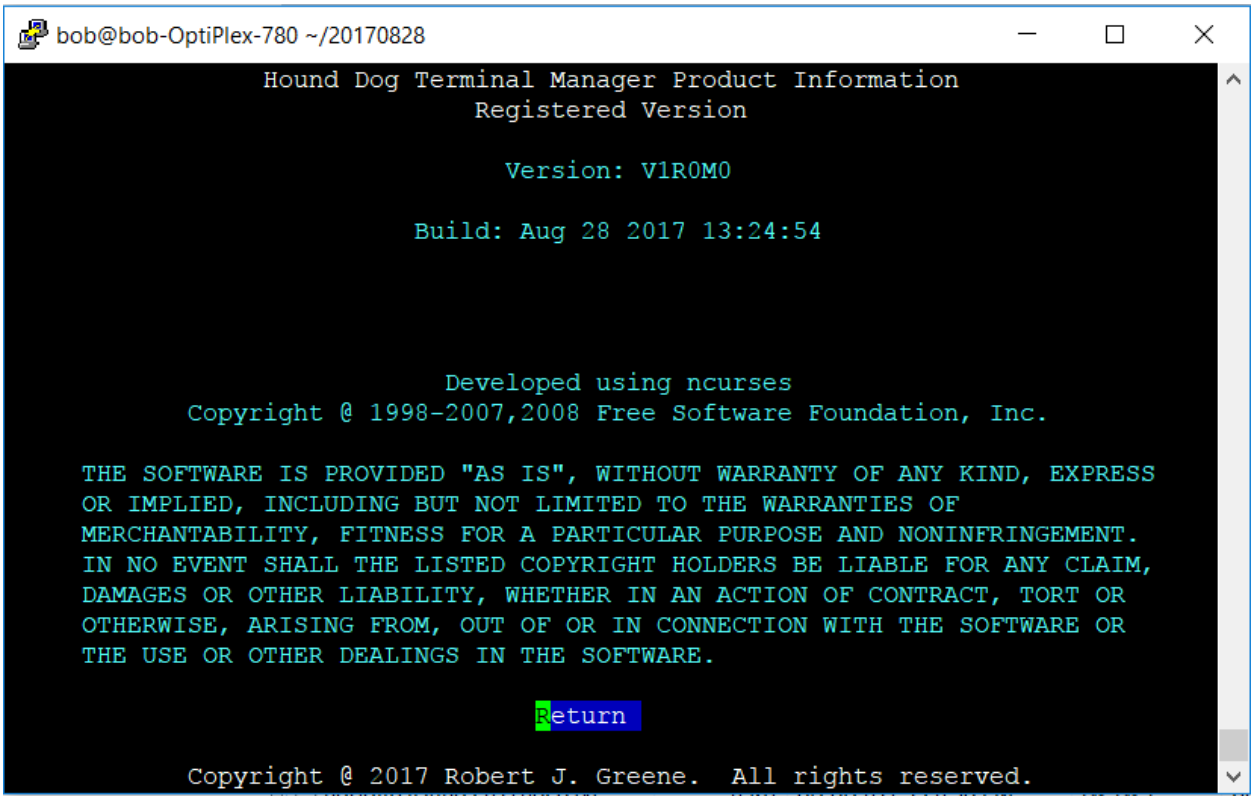


Figure 34: Display Product Information.

The HDog Terminal Manager is being initially released as Shareware, providing a 30 day trial period. Registration must be completed prior to the end of the 30 day trial to continue use.

The first release id will be V1R0M1 and is currently available. Installation packages can be downloaded from the www.hounddogtech.us website.

The HDog Terminal Manager is written in GNU C and was developed using ncurses and other GNU libraries. ncurses is copyrighted software of the Free Software Foundation, Inc.

This software is provided “As Is”, without warranty of any kind, expressed or implied including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the listed copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort, or otherwise, arising from, out or in connection with the software or the use of other dealings in the software.

Command Shell

Figure 35: Command Shell.

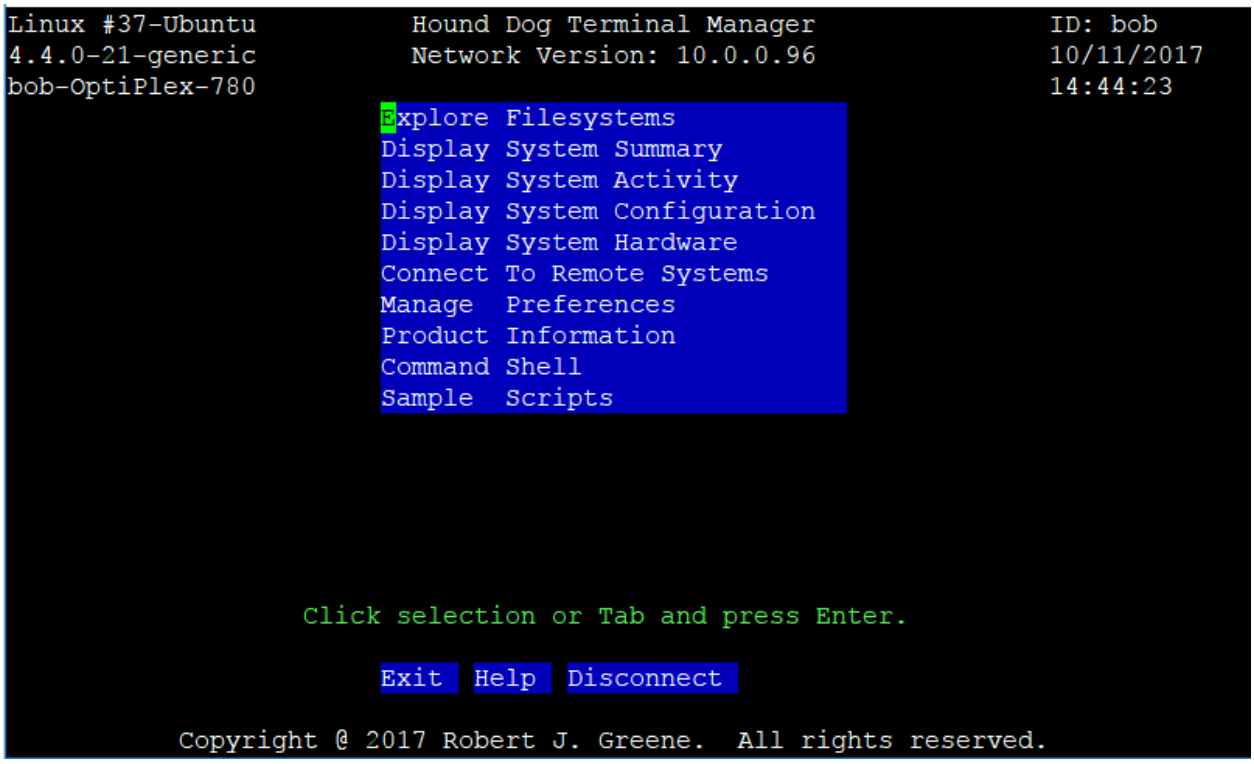
When the command shell is selected, a panel like the one above will be presented. You are free to perform any shell commands you wish. When you type **exit**, you be returned to the Hound Dog Terminal Manager.

Network Version Feature

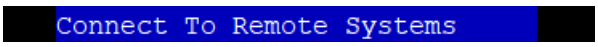
The Hound Dog Terminal Manager Network Version can be installed on one Linux System and have its actions performed on up to 100 Remote Linux Systems. You will setup each Remote System with an IP address, Port, and User ID to use. You may also assign a Nickname. The Hound Dog Terminal Manager will use Secure Sockets to connect and execute the actions you specify and return the results back. Here are the details of this feature:

Main Menu

Here is an example of the Hound Dog Terminal Manager Network Version Main Menu panel:

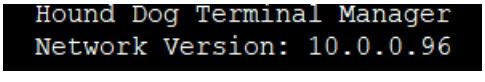


The Main Menu panel has the following differences from the Local Version:

- Connect to Remote Systems Menu Item -> 

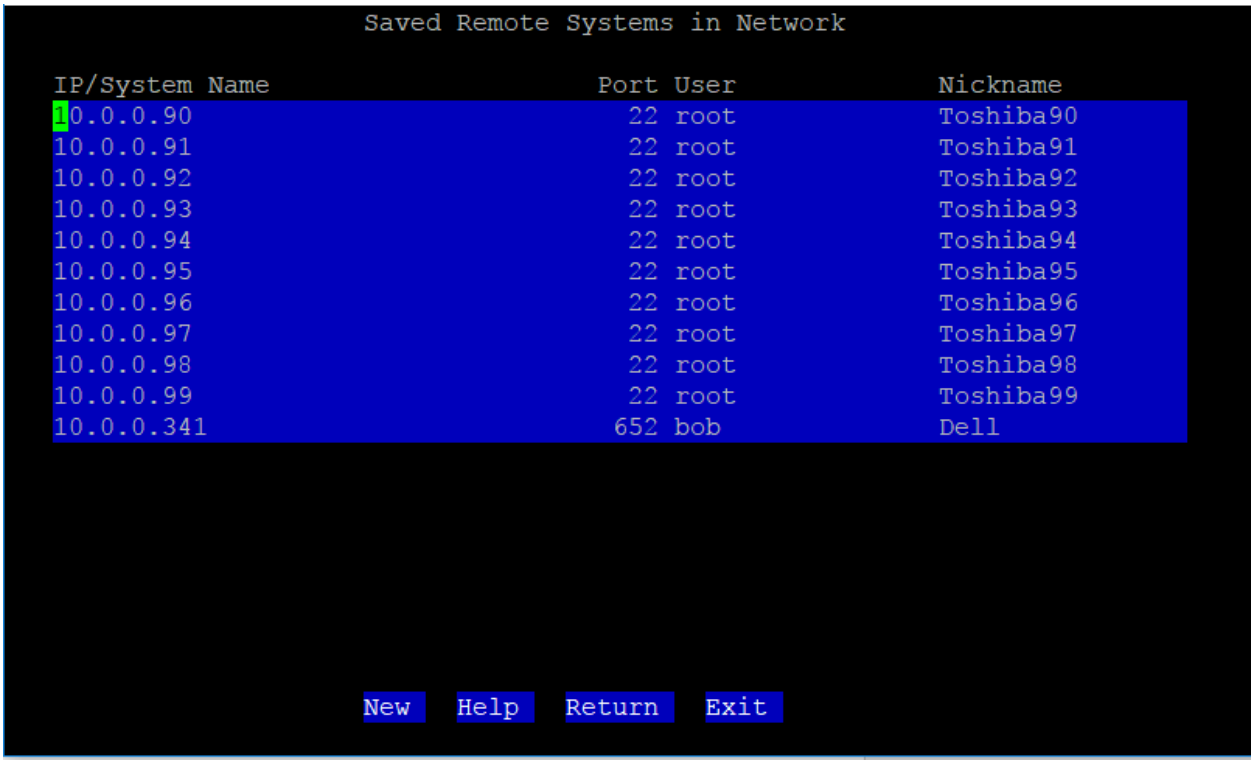
  Clicking on this Menu Item will result in a list of predefined Remote Systems to select from.

- Remote System IP in Title when a Remote System has been connected. In the example a Remote System with an IP of 10.0.0.96 is currently connected.



- Disconnect Push Button: When a Remote System is currently connected, a Disconnect Push Button is displayed.

Saved Remote Systems in Network Panel



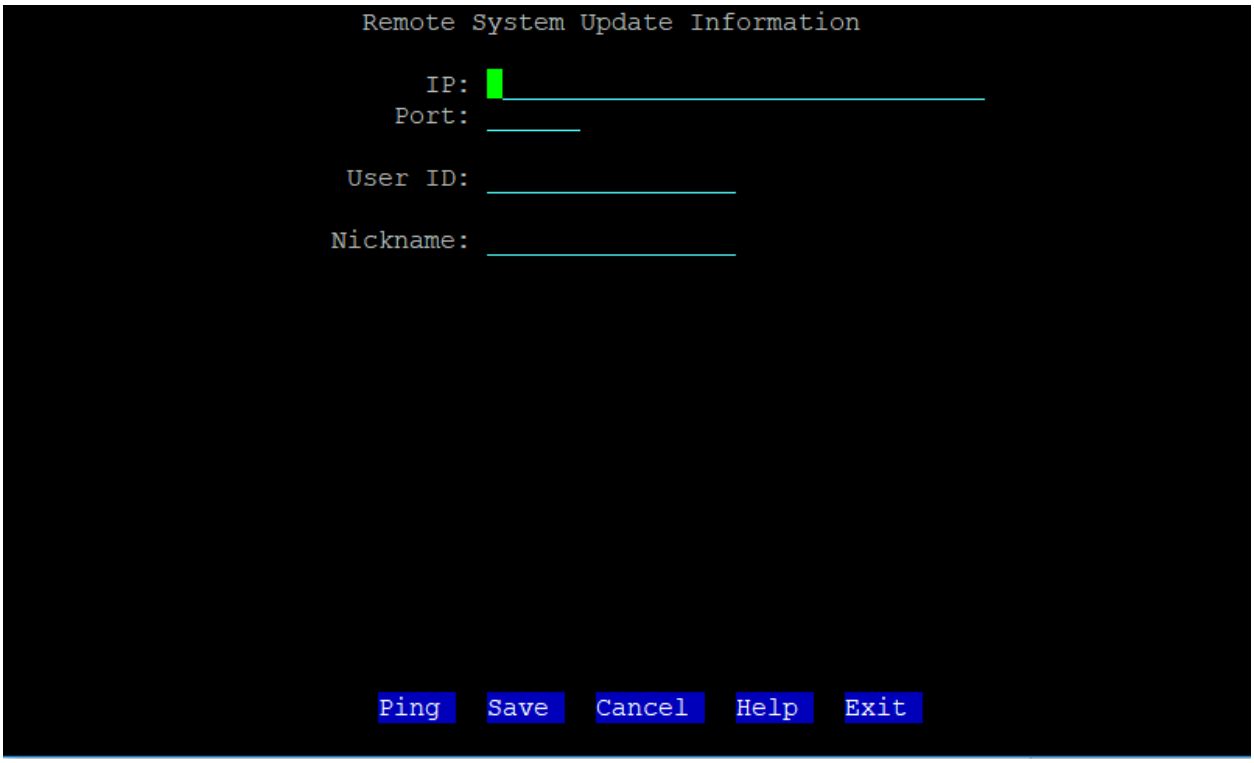
When the **Connect to Remote Systems** Menu Item is selected from the **Main Menu**, A list of Remote Systems will be displayed. Up to 100 Remote Systems can be defined. The list will include the IP or Host Name, Port and User ID to use, and an optional Nickname. The Hound Dog Terminal Manager will use Secure Sockets to execute actions on the Remote System.

Click on the **New** Push Button to add a new Remote System. If no Remote Systems has been defined yet, you will be presented with an entry panel to add a new Remote System.

Clicking on a Remote System list will result in a Remote System Detail panel like the example on the next page.

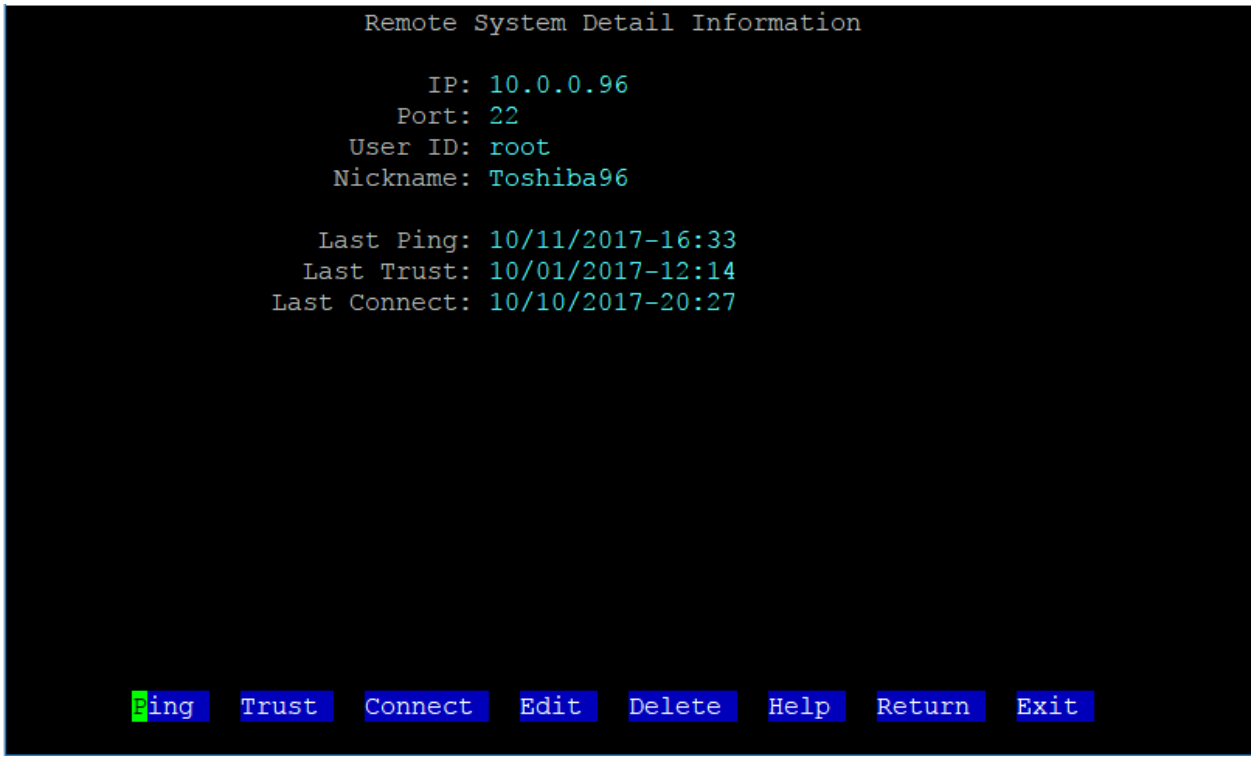
The Remote System list can be sorted ascending or descending by IP/System Name, Port, User, or Nickname. Simply click on the respective heading to cause the sort to occur. Clicking the first time on a particular heading will cause the list to be sorted ascending. Clicking on the sorted-ascending list's heading a second time will cause the list to be presented in descending order.

Adding a New Remote System



Above is an example of the response you will see when you select the **New** Push Button on the **Saved Remote Systems in Network Panel**. After entering the above information, select **Save** to save your new Remote System entry or **Cancel** to cancel your changes. You may also select **Ping** to ensure that the Remote System is reachable.

Remote System Detail



Clicking on a Remote System on the **Saved Remote Systems in Network Panel** will return a panel like above. The **Remote System Detail** panel provides for managing a Remote System. Here is a list of the actions that you can do:

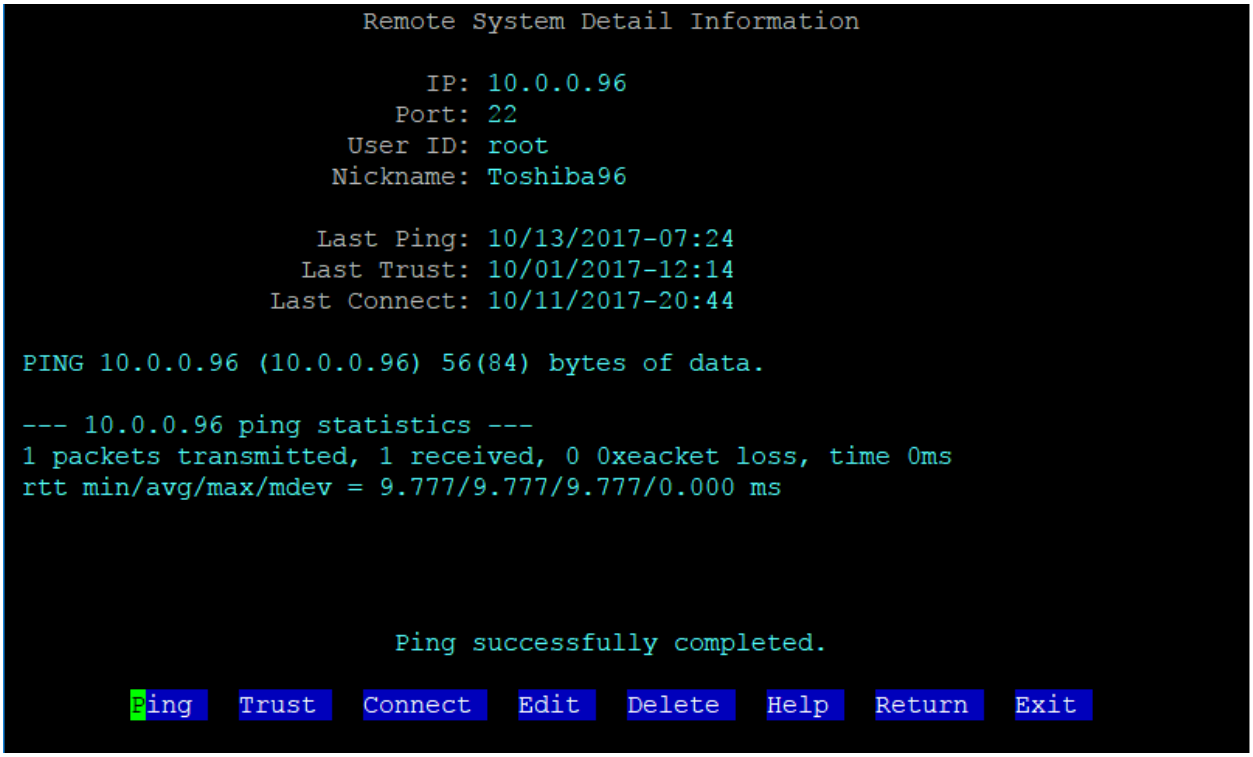
- Edit: Edit the **IP/Name**, **Port**, **User ID**, and **Nickname** for the Remote System.
- Delete: Delete the Remote System.
- Ping: Ping the IP/Name of the Remote System to ensure that it is reachable.
- Trust: Establish Secure Shell (SSH) Trust between your signed-on Linux System and User ID with the Remote Linux System and User ID. This will eliminate having to reenter user id passwords each time that the Hound Dog Terminal Manager executes an action on the Remote System.

  Note: On Ubuntu systems, many actions must be executed with the **sudo** command, (Super User Do). Use the **visudo** command on the Remote System to add an entry for the User ID to avoid having to enter a password each time the **sudo** command is executed on the Remote System.

- Connect: Connect to the Remote System. This will result in actions being performed on the Remote System until a Disconnect is selected.

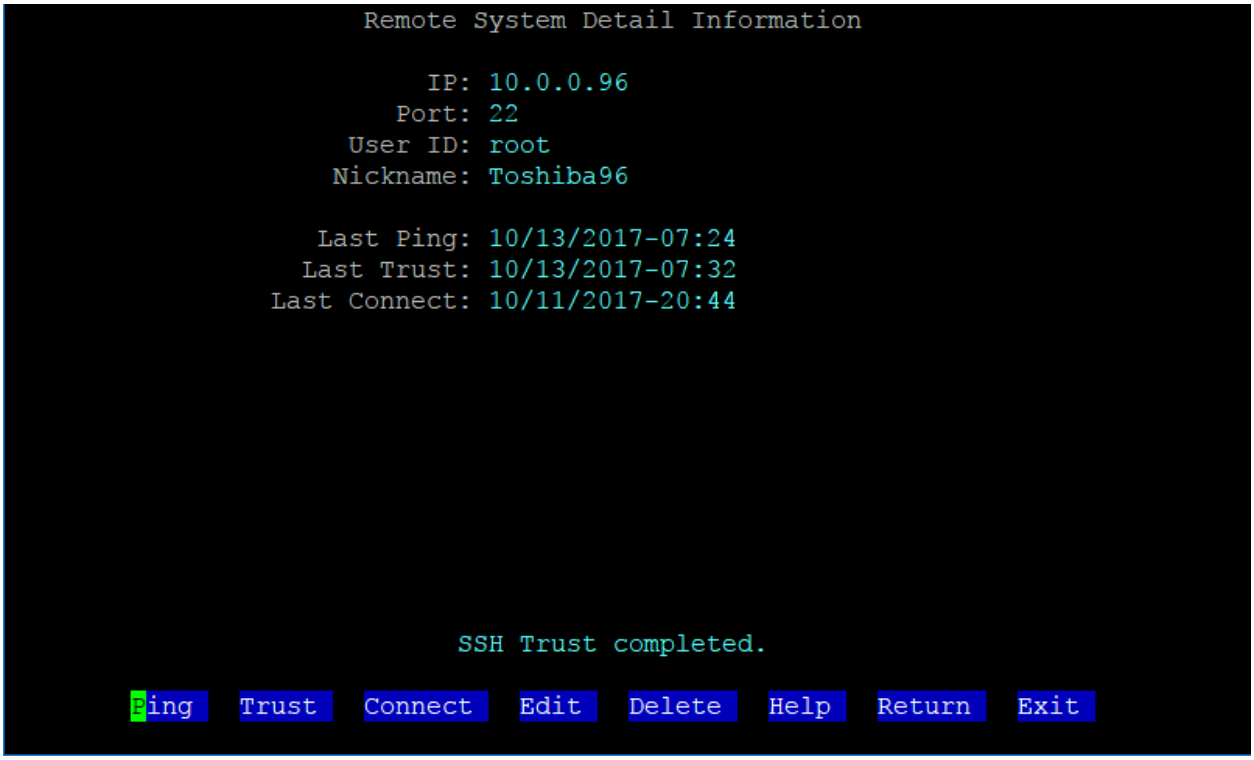
The following pages display examples of executing the above actions.

Remote System Detail - Ping



The **Remote System Detail** panel displayed is an example of the result when you select the **Ping** Push Button. In the above example the Ping was successfully completed, showing that the IP/Name is reachable.

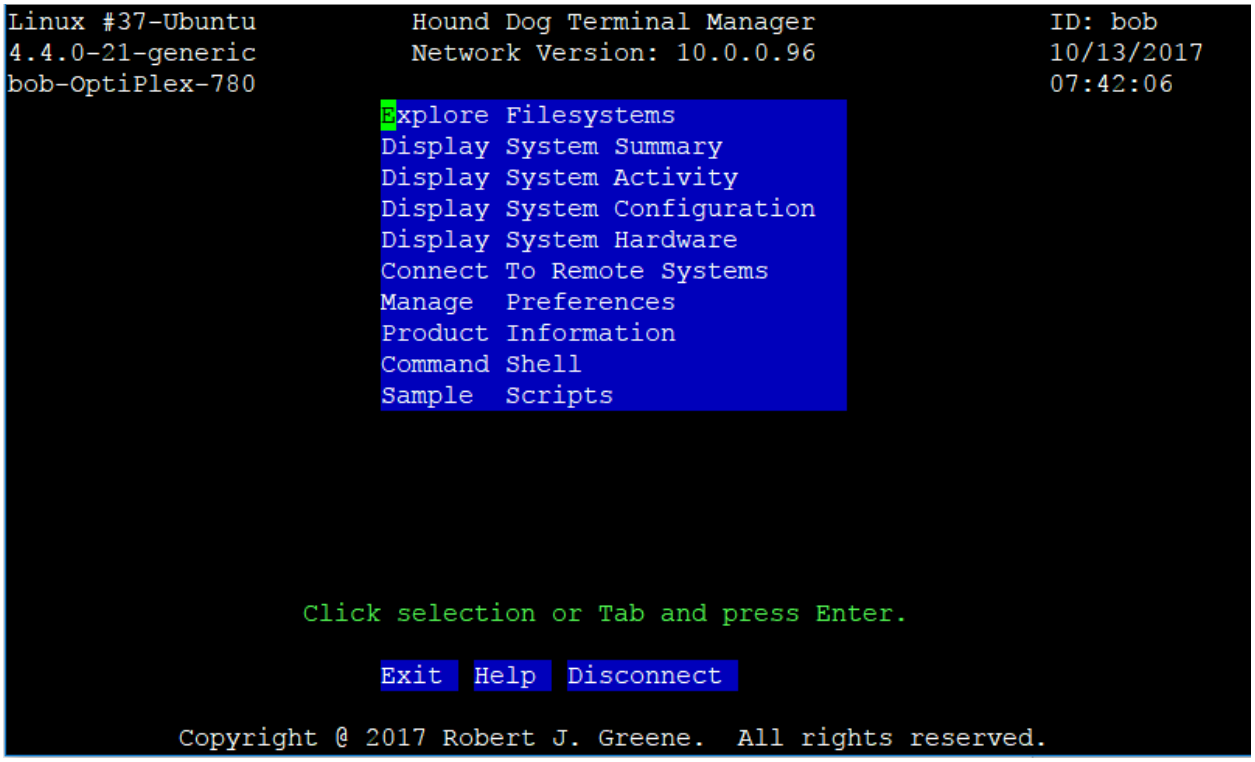
Remote System Detail- Trust



The **Remote System Detail** panel displayed is an example of the result when you select the **Trust** Push Button. In the above example Secure Sockets Trust was established between your signon Linux System and UserID with the IP/Name-UserID that you set up for this Remote System.

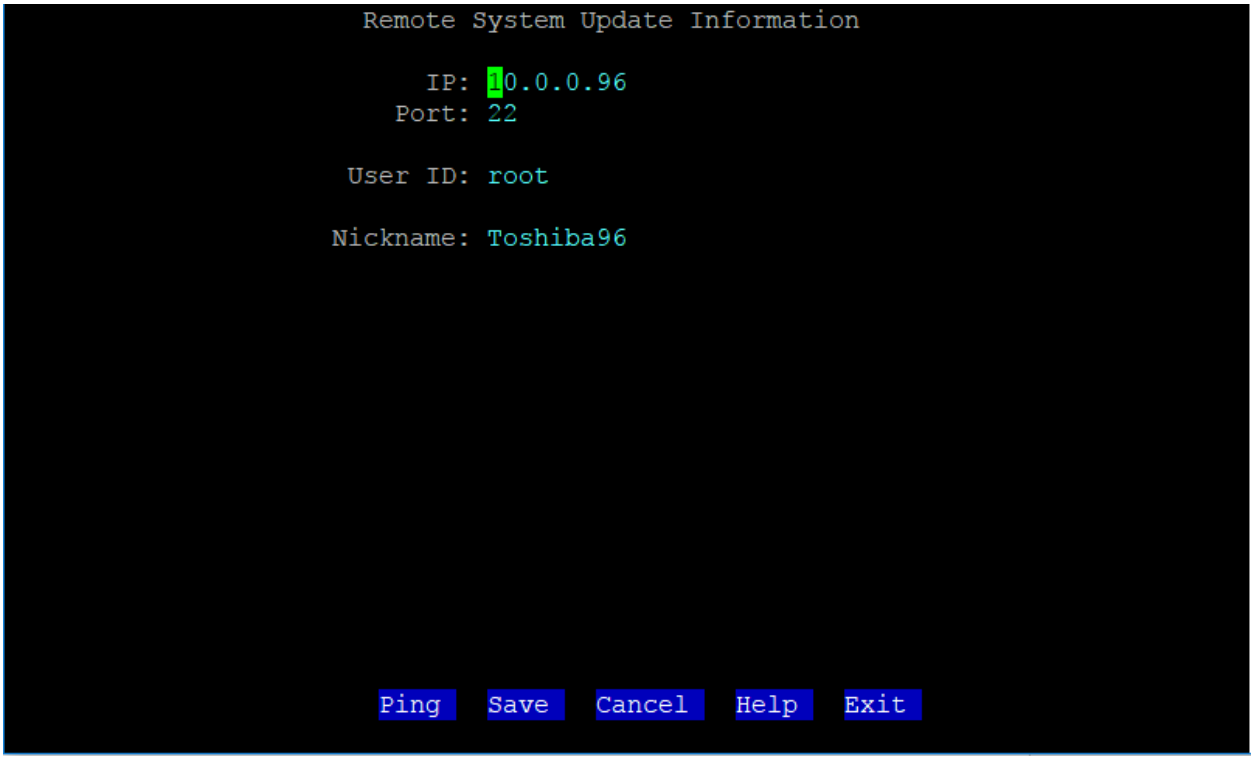
Note: When establishing Trust, you may be prompted to enter the Remote System's UserID password multiple times. Below is an example of the prompt that you will see:

Remote System Detail- Connect



Above is an example of selecting the Connect Push Button on the Remote System Detail panel for IP/Name 10.0.0.96. A ping will be performed on the IP/Name value to ensure the Remote System is reachable. If successful, you will be returned to the Hound Dog Terminal Manager Main Menu with the Remote System set as the Connected Remote System.

Remote System Detail- Update



When you select the **Edit** Push Button on the Remote System Detail Panel, you will be presented with a Remote System Update panel like above. Make any changes and use the **Save** Push Button to save your changes and the **Cancel** Push Button to cancel any unsaved updates.

Shareware Evaluation Mode

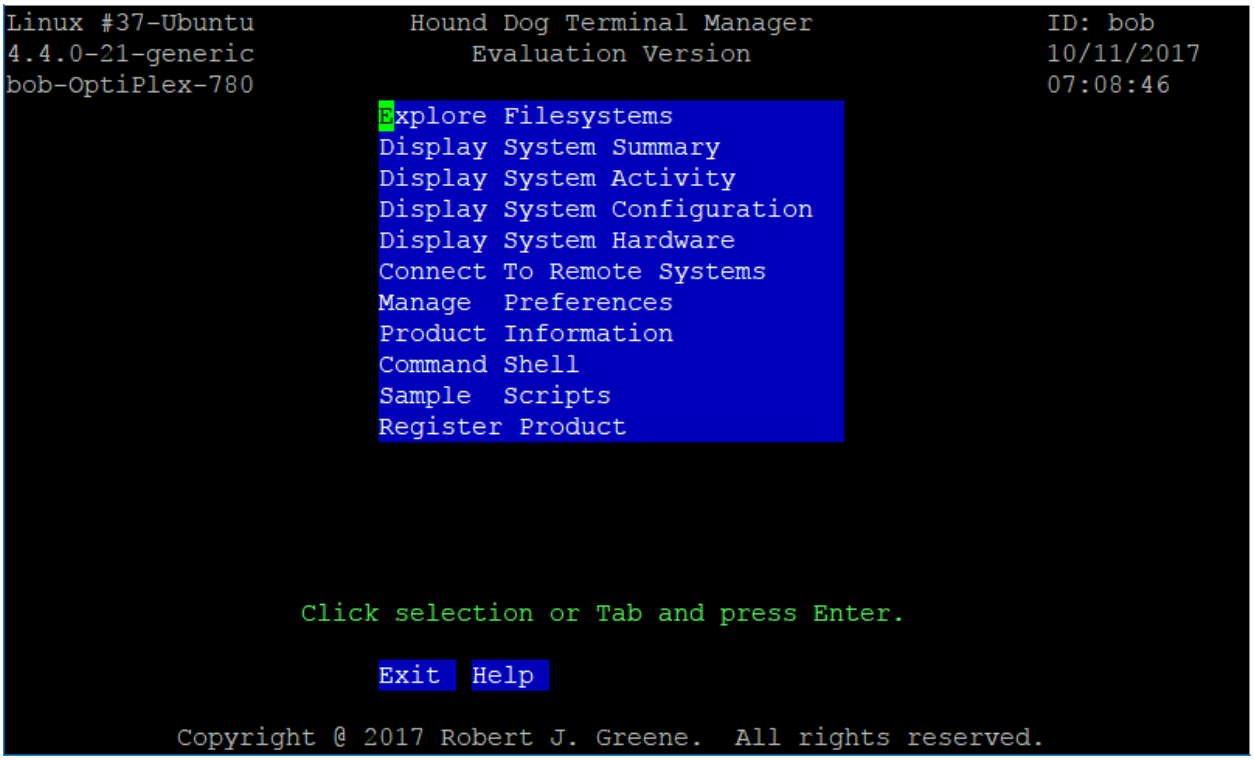


Figure 36: Shareware Evaluation Mode.

The HDog Terminal Manager is being initially released as Shareware, providing a 30 day trial period. Registration must be completed prior to the end of the 30 day trial to continue use.

The first release id isV1R0M1 and available in October, 2017. Single user licenses, (Registration), can be purchased on the www.hounddogtech.us website. You can also download the installation package and User Guide on that website.

After completing registration at the www.hounddogtech.us website, you will be given a registration key. Select the **Register Product** menu item above to complete registration.

Product Registration

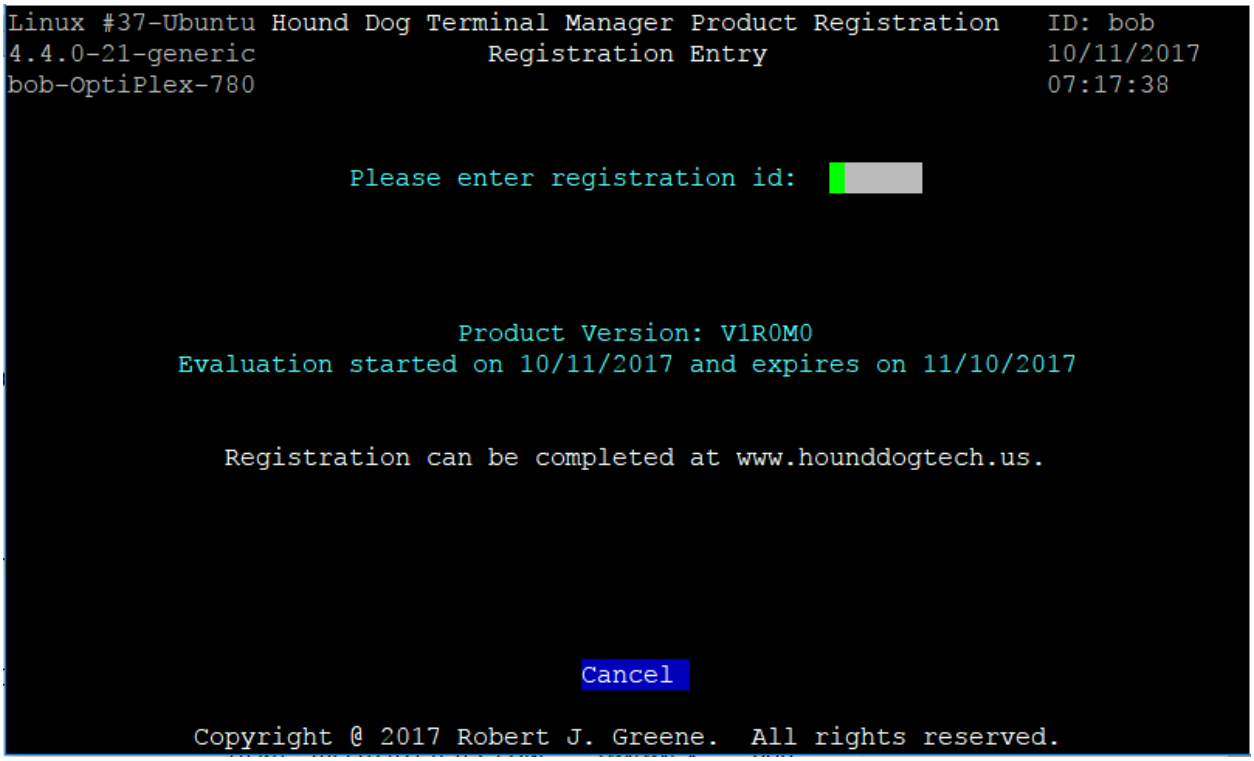


Figure 37: Product Registration.

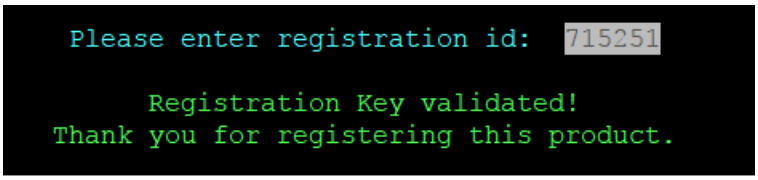
The HDog Terminal Manager is being initially released as Shareware, providing a 30 day trial period. Registration must be completed prior to the end of the 30 day trial to continue use.

The first release id will be **V1R0M1** and is currently available. An installation package is available for download on the www.hounddogtech.us website.

In the evaluation period you will see an expiration message in the heading stating the expiration date. After completing registration at the www.hounddogtech.us website, you will be given a registration key.

After selecting the **Register Product** menu item from the Main Menu you will be presented with the above Product Registration Entry panel. Input the product key in the **registration id** prompt to complete registration.

Successful registration will result in a response such as this →



Setup and Customization Considerations

HDog RunTime Parameters

The HDog Terminal Manager accepts runtime parameters at startup. These parameters include:

-f xmlfilename: The HDog XML file default location and name is in the same folder as the hdog executable. Specifying the **–f** runtime parameter allows you to specify a new location and/or name for this file.

-l logfilename: The HDog Terminal Manager writes log records to a log file when significant events occur. The default location name for this file is **/tmp/HDogLog.txt**. It will append new messages to this file over multiple sessions. Use the **–l** runtime option to specify a different log file location and name.

-**d directoryname**: The HDog Terminal Manager uses a Configuration File to maintain options over multiple sessions. The location of this Configuration file defaults to the same folder as the hdog executable. Use –d parameter to specify a different location.

debug: The HDog Terminal Manager provides debugging information in its log file upon request. Specify debug to turn this feature on.

Menu Customizations

The Hound Dog Terminal Manager's menu structure is constructed using the hdog.xml file. This XML file is composed of MenuItem sections which define submenus and menu items presented to the user.

The figure below is an example of a MenuItem in the hdog.xml file that invokes the Explore Filesystem Menu Item in the Main Menu.

```
  <MenuItem>
    <Title>Explore Filesystem</Title>
    <Invoke>./hdogfsys</Invoke>
    <Help>                          Explore Filesystem Help

      The term filesystem used here refers to the entire hierarchy
      of directories (also referred to as the directory tree) that
      is used to organize files on this system.
      The directories start with the root directory, (/), which
      contains a series of subdirectories, each of which, in turn,
      contains further subdirectories, etc.  Subdirectories are
      displayed with a trailing '/' in their name.

      Clicking on a Subdirectory will result in a display of that
      Subdirectory's contents.

      Clicking on a File will return detail information about that
      File along with the option to browse or edit it if you are
      authorized to do so.

      Clicking on the '../' Subdirectory name will result in a
      display of the most recent 'upper' directory.
    </Help>
  </MenuItem>
```

Figure 38: Sample XML entry from hdog.xml.

The MenuItem Title section specifies the displayed title when included in a menu. When this menu item is selected by the user, the Invoke value is executed. Specifying a value of **SubMenu** for the MenuItem Title will result in a Submenu being presented to the user when the MenuItem is selected.

The User Interface can be extended by simply adding new MenuItems or modifying existing MenuItems in the hdog.xml file.

Help documentation inserted into the MenuItem Help section will be displayed when a user selects **Help** on the panel associated with MenuItem.

Below are a few more examples of MenuItems that you can create for your use.

```
    <MenuItem>
      <Title>List Mounted Filesystems</Title>
      <Invoke>Shell(clear;uname -a;echo "Mounted Filesystems follow:";cat
/etc/mtab;echo "Press any key to continue.";read -p " " cInputKey)</Invoke>
      <Help>                    Hound Dog Command Shell Help
      </Help>
    </MenuItem>
```

Figure 39: Sample Shell Command Script that lists all Mounted Filesystems in a Linux System.

In the above example, a shell is created with the above commands being executed. The result is that the screen will be cleared and a title of "Mounted Filesystems follow:" will be printed. The next lines will be a list of the mounted filesystems. Note that when a Remote System has been connected, these commands will be executed on that system and displayed on your console.

Note: In the above Figure the <Invoke> command did not fit on the page. The commands are on a single line and included within the **Shell()** value.

Here is another example which lists all active Processes on a Linux System.

```
    <MenuItem>
      <Title>List Active Processes</Title>
      <Invoke>Shell(clear;uname -a;ps -ef;echo "Press any key to
continue.";read -p " " cInputKey)</Invoke>
      <Help>                    Hound Dog Command Shell Help
      </Help>
    </MenuItem>
```

Figure 40: Sample Shell CommandScript that lists all Active Processes in a Linux System.

The above examples are shown to help you create MenuItems that you wish to include in your menus for your own purposes.

