Notes

By Kiran Rao

Overview

IDT provides a royalty free, license-fee free Linux operating system environment for its Interprise™ family of integrated communications processors. In addition to the basic operating system and device drivers, IDT provides higher level applications functions common in gateway and other applications. This application note describes the File and Print Server applications software available with IDT Linux.

The Linux File Server and Print Server environments are extremely useful networking tools for anyone who has both Windows and Unix systems on their network. Running on a Unix system, the software allows Windows to share files and printers residing on the Unix host, and also allow Unix users to access resources shared by Windows systems (see Figure 1). There is no requirement that the Unix host be a computer; the File and Print Servers can also be running on a media server box within a home network, allowing users to share printers and hard disks within the home, without the restriction of having a dedicated computer for these services. This application note focuses on the Samba File Server and CUPS (Common Unix Printing System) print server.
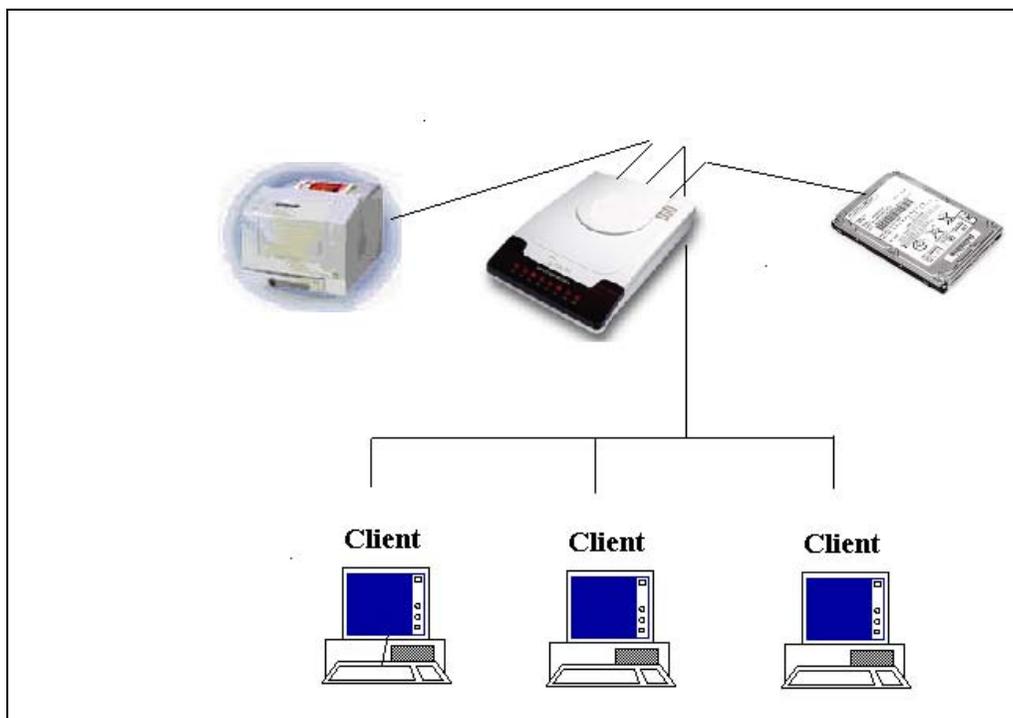

Figure 1 Linux-based Media Gateway, with Integrated File Server and Print Server

Samba As a File Server

Samba is an Open Source/Free Software suite that provides seamless file and print services to SMB/CIFS clients. Samba is freely available under the GNU General Public License.

IDT has integrated the Samba File Server Application into its Linux release. This allows several Windows-based machines to share a common hard disk which can be present on an embedded platform (e.g. Media Server) running IDT Linux. Users can connect to the server and share files with other users or

DSC 6423

Notes

have their own personal disk space. On a Windows-based machine, it is possible to use an SMB (Server Message Block) file share as if it were a hard disk on a user's own machine (see Figure 2). The N: drive can appear fully as local disk space when, in reality, it is disk space on a network server. This mapping can be achieved by the Map Network Drive feature provided by Windows.
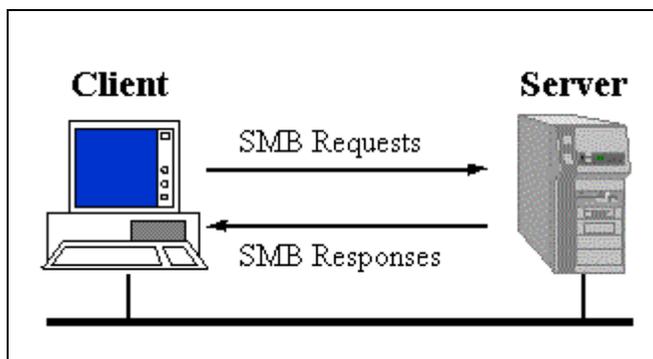

Figure 2 Samba File Server

SMB Protocol

The SMB protocol allows file sharing to occur. SMB is a client server, request-response protocol. Figure 2 illustrates the way in which SMB works. The only exception to the request-response nature of SMB (that is, where the client makes requests and the server sends back responses) is when the client has requested opportunistic locks (oplocks) and the server subsequently has to break an already granted oplock because another client has requested a file open with a mode that is incompatible with the granted oplock. In this case, the server sends an unsolicited message to the client, signaling the oplock break.

Opportunistic locking allows a client to notify the Samba server that it will not only be the exclusive writer of a file, but that it will also cache its changes to that file on its own machine (and not on the Samba server) in order to speed up file access for that client. When the Samba server knows that a file has been opportunistically locked by a client, it marks its version as having an opportunistic lock and waits for the client to complete work on the file, at which point it expects the client to send the final changes back to the Samba server for synchronization.

If a second client requests access to that file before the first client has finished, Samba can send an oplock break request to the first client. This tells the client to stop caching its changes and return the current state of the file to the server so that the interrupting client can use it as it sees fit.

At a high level, the SMB protocol suite is relatively simple. It includes commands for all the file and print operations that might be performed on a local disk or printer, such as:

- – *Opening and closing files*
- – *Creating and deleting files and directories*
- – *Reading and writing files*
- – *Searching for files*
- – *Queueing and dequeueing files in a print spool.*

Samba Configuration

The Samba configuration file, called smb.conf by default, uses the same format as Windows .ini files, and thus is easy to create or modify by individuals familiar with .ini files. The following is an example of a Samba configuration file:

Global parameters

     workgroup = idt

     server string = Samba SMB Server

     interfaces = 192.168.1.1/24

Notes

```
        bind interfaces only = Yes

        security = SHARE

        max log size = 50

        read bmpx = No

        time server = Yes

        socket options = TCP_NODELAY

        os level = 65

        preferred master = Yes

        dns proxy = No

        wins support = Yes

        guest account = Guest

        hide dot files = No

    encrypt passwords = yes


    [home]

        comment = home

        valid users = root

        path = /

        read only = No

        create mask = 0660

        directory mask = 0770

        guest ok = No
```

This default configuration file sets up a workgroup in which Samba authenticates users using encrypted passwords and the default user-level security method. The [home] share has been added to allow Samba to create a disk share for the home directory of each user who has a standard Linux account on the server.

Security in Samba

The SMB model defines two levels of security:

– *Share level. Protection is applied at the share level on a server. Each share can have a password, and a client only needs that password to access all files under that share. This was the first security model that SMB had and is the only security model available in the Core and CorePlus protocols. Windows for Workgroups' vserver.exe implements share level security by default, as does Windows 95.*

– *User Level. Protection is applied to individual files in each share and is based on user access rights. Each user (client) must log in to the server and be authenticated by the server. When it is authenticated, the client is given a UID which it must present on all subsequent accesses to the server.*

Disconnecting from File Server

To disconnect the network drive, Windows provides a Disconnect Network Drive feature analogous to the Map Network Drive feature. This ensures that the drive letter can be reused on the Windows machine.

Samba Summary

The Samba File Server provides a convenient method of storing and sharing files and avoiding duplication. This networking tool can be used in homes running Media Server Gateways or Home NAS appliances and also in corporations to share hard disks on a server.

Notes

CUPS Print Server

The Common UNIX Printing System (CUPS) is a cross-platform printing solution for all UNIX environments. It is based on the Internet Printing Protocol and provides complete printing services to most Post-Script and raster printers. CUPS provides a portable printing layer for UNIX®-based operating systems. It has been developed by Easy Software Products to promote a standard printing solution for all UNIX vendors and users. CUPS provides the System V and Berkeley command-line interfaces.

CUPS uses the Internet Printing Protocol (IPP) as the basis for managing print jobs and queues. The Line Printer Daemon (LPD) Server Message Block (SMB), and AppSocket (a.k.a. JetDirect) protocols are also supported with reduced functionality. CUPS adds network printer browsing and PostScript Printer Description (PPD) based printing options to support real-world printing under UNIX.

Design Overview

Like most printing systems, CUPS is designed around a central print scheduling process that dispatches print jobs, processes administrative commands, provides printer status information to local and remote programs and to users, as needed. Figure 3 shows the basic organization of CUPS.
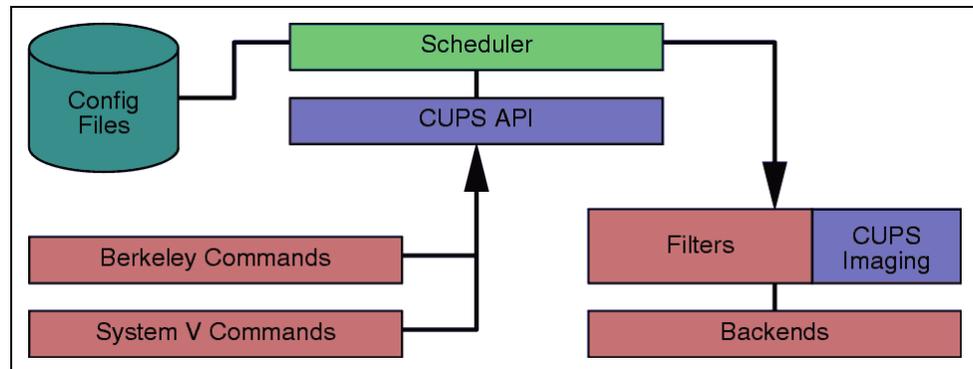

Figure 3 CUPS Block Diagram

Scheduler

The scheduler is a HTTP/1.1 server application that handles HTTP requests. Besides handling printer requests via IPP POST requests, the scheduler also acts as a full-featured web server for documentation, status monitoring, and administration. The scheduler also manages a list of available printers on the LAN and dispatches print jobs as needed using the appropriate Filters and Backends.

Configuration Files

The configuration files consist of:

– *The HTTP server configuration file*
– *Printer and class definition files*
– *MIME type and conversion rule files*
– *PostScript Printer Description (PPD) files.*

The HTTP server configuration file is purposely similar to the Apache server configuration file and defines all of the access control properties for the server. The printer and class definition files list the available printer queues and classes. Printer classes are collections of printers. Jobs sent to a class are forwarded to the first available printer in the class, round-robin fashion.

The MIME type files list the supported MIME types (text/plain, application/postscript, etc.) and "magic" rules for automatically detecting the format of a file. These are used by the HTTP server to determine the Content-Type field for GET and HEAD requests and by the IPP request handler to determine the file type when a Print-Job or Send-File request is received with a document-format of application/octet-stream.

Notes

The MIME conversion rule files list the available filters. The filters are used when a job is dispatched, so that an application can send a convenient file format to the printing system which then converts the document into a printable format. Each filter has a relative cost associated with it, and the filtering algorithm chooses the set of filters that will convert the file to the needed format with the lowest total "cost".

The PPD files describe the capabilities of all printers, not just PostScript printers. There is one PPD file for each printer. PPD files for non-PostScript printers define additional filters through cupsFilter attributes to support printer drivers.

CUPS API

The CUPS API contains CUPS-specific convenience functions for queuing print jobs, getting printer information, accessing resources via HTTP and IPP, and manipulating PPD files. Unlike the rest of CUPS, the CUPS API is provided under the terms of the GNU LGPL, so it may be used by non-GPL applications.

Berkeley and System V Commands

CUPS provides the System V and Berkeley command-line interfaces for submitting jobs and checking the printer status. The lpstat and lpc status commands also show network printers (printer@server) when printer browsing is enabled.

The System V administration commands are supplied for managing printers and classes. The Berkeley printer administration tool (lpc) is only supported in a read-only mode to check the current status of the printer queues and scheduler.

Filters

A filter program reads from the standard input or from a file if a filename is supplied. All filters must support a common set of options, including printer name, job ID, username, job title, number of copies, and job options. All output is sent to the standard output. Filters are provided for many file formats and include image file and PostScript raster filters that support non-PostScript printers. Multiple filters are run in parallel to produce the required output format.

The PostScript raster filter is based on the GNU Ghostscript 5.50 core. Instead of using the Ghostscript printer drivers and front-end, the CUPS filter uses a generic raster printer driver and CUPS-compliant front-end to support any kind of raster printer. This allows the same printer driver filter to be used for printing raster data from any filter.

CUPS Imaging

The CUPS Imaging library provides functions for managing large images, doing colorspace conversion and color management, scaling images for printing, and managing raster page streams. It is used by the CUPS image file filters, the PostScript RIP, and all raster printers drivers.

Backends

A backend program is a special filter that sends print data to a device or network connection. Backends for parallel, serial, USB, LPD, IPP, and AppSocket (JetDirect) connections are provided in CUPS 1.1.

SAMBA version 2.0.6 and higher includes a SMB backend (smbspool(1)) that can be used with CUPS 1.0 or 1.1 for printing to Windows.

Printers Supported by CUPS

CUPS supports printers connected to the host using Serial port, Parallel port, and Network printers.   To determine if a specific printer is supported in CUPS and to obtain drivers, refer to the printer database at http://www.linuxprinting.org.

Notes

CUPS Summary

The Common UNIX Printing System provides a modern printing interface for UNIX applications that is both flexible and user-friendly. The software provides System V and Berkeley compatible command-line interfaces to ensure compatibility with existing applications. CUPS 1.1 adds many new features that make it an even better choice for printing under UNIX.

Summary

IDT continues to enhance its Linux offering, allowing vendors of gateway and other equipment to offer compelling capabilities within the home network. Additional information on the capabilities of IDT Linux and on the IDT™ Interprise™ family of processors can be found on the IDT web site: www.idt.com.

Reference

http://www.samba.org

http://www.cups.org

http://www.linuxprinting.org

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES ("RENESAS") PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/