



LPIC-1 TRAINING COURSE

Topic 108: Essential System Services

Contents



1. Maintain system time

2. System logging

3. Mail Transfer Agent (MTA) basics

4. Manage printers and printing

Objectives

- ❖ Properly maintain the system time and synchronize the clock via NTP
- ❖ Configure the ***syslog*** daemon
- ❖ Aware of the commonly available MTA programs and able to perform basic forward and alias configuration on a client host
- ❖ Manage print queue and user print jobs using CUPS and the LPD compability interface



1. Maintain system time

Overview

Learn to:

- ❖ Set the system date and time
- ❖ Set the hardware clock to the correct time in UTC
- ❖ Configure the correct timezone
- ❖ Basic NTP configuration
- ❖ Knowledge of using the pool.ntp.org service

Linux Time Concepts

- ❖ *Hardware clock*: maintain the time while the computer is turned off
- ❖ *Software clock (system clock)*: set on system boot to the same as hardware clock
- ❖ Linux use *system clock* for most purposes
 - While the system is running, changes to one of these doesn't effect the other
- ❖ System clock is set to *UTC* or *GMT*
 - *Hardware clock* should be set to UTC too
 - Timezone is used when time is displayed to user
- ❖ Both the *hardware* and *software clock* are notoriously unreliable on standard x86 hardware

Manually Setting the Time

❖ Setting software clock:

`date [-u] [MMDDhhmm[[CC]YY][.ss]]`

- Example: Set the time to *3:02PM, 27/10/2010*

`$ date 102715022010`

❖ Setting hardware clock (other than from BIOS):

`hwclock [--utc | --localtime] --set --date=newdate`

❖ Set software clock based on hardware clock:

`hwclock --hctosys`

❖ Set software clock based on hardware clock:

`hwclock --systohc`

Understanding NTP basics

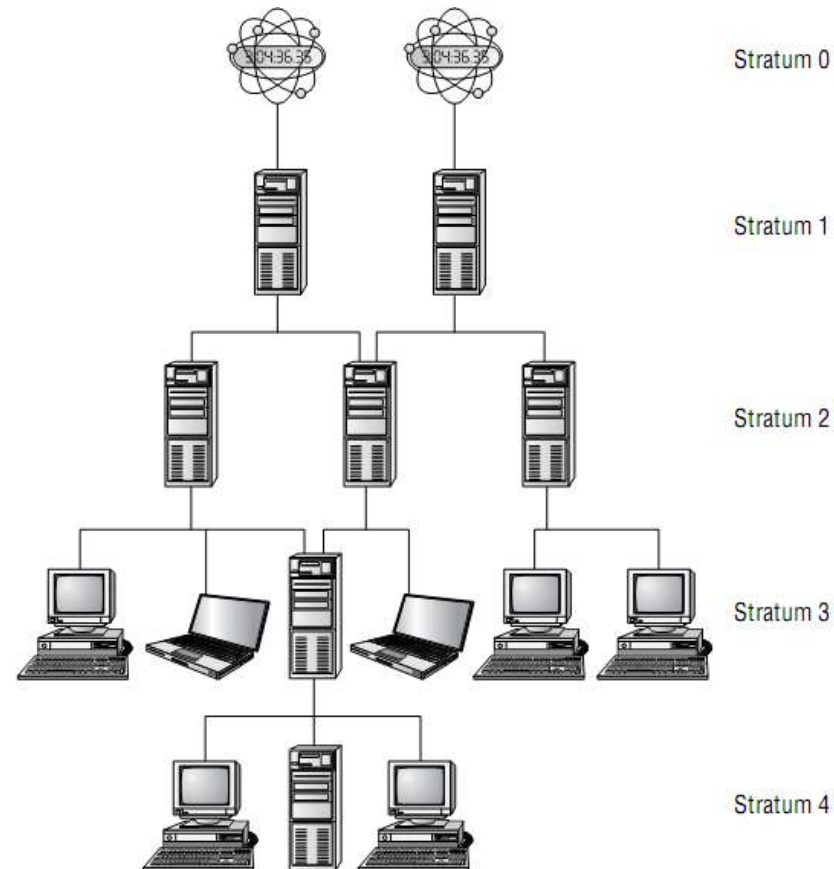
❖ NTP protocol creates a tiered hierarchy of time sources

- *Stratum 0*: highly accurate time sources (atomic clocks)
- *Stratum 1*: connect directly to Stratum 0, running NTP service
- *Stratum 2...*: get time from upper Stratum

❖ NTP works by measuring the round-trip time for packets between the server and the client

❖ NTP server program functions as both a server and a client

- the server improves the accuracy of the system clock through ***ntp.drift***
- a full NTP server periodically checks with its source systems to keep the system time set correctly and to update the ***ntp.drift*** file



Locating a Time Source

- ❖ Your ISP's NTP servers
- ❖ Your distro's NTP servers
- ❖ Public NTP server lists
(<http://support.ntp.org/bin/view/Servers/WebHome>)
- ❖ Public NTP server pool: *pool.ntp.org*
- ❖ Try using **ping** to determine the round-trip time for packets to this system.

Configuring NTP Servers

- ❖ Install NTP software (*ntp*, *xntp*, *ntpd* or *xntpd*)
- ❖ Edit NTP's configuration file (*/etc/ntp.conf*)

```
server clock.example.com  
server ntp.pangaea.edu  
server time.luna.edu
```
- ❖ Start/restart NTP daemon

```
/etc/init.d/ntpd restart
```
- ❖ Verify NTP is working:

```
ntpq  
ntpq> peers
```

Configuring NTP Clients

- ❖ Just like the NTP server configuration with some exceptions:
 - NTP Clients refer to *your* NTP server rather than to an outside NTP source
 - Ensure that your NTP clients can't be accessed as servers
 - Add this line in ***ntp.conf***:
restrict default ignore
- ❖ Simple way to set time on a client (one-time clock setting): **ntpdate <servername>**



2. System logging

Overview

Learn about:

- ❖ Syslog configuration files
- ❖ Syslog
- ❖ Standard facilities, priorities and actions

Understanding *syslogd*

- ❖ The daemon that handles messages from servers and other user-mode programs
 - Provide a unified means of handling log files
- ❖ Usually paired with a daemon called ***klogd***
- ❖ Accepts data delivered from servers and other programs
 - Most programs are servers & system tools
- ❖ Classify received data and direct it to an appropriate log file
- ❖ Configuration file: ***/etc/syslog.conf***

Form of */etc/syslog.conf*

- | | <u>facility.priority</u> | <u>action</u> |
|-------------|---|---------------|
| ❖ facility | :code word for the type of program or tool that generated the message (<code>auth</code> , <code>authpriv</code> , <code>cron</code> , <code>daemon</code> , <code>kern</code> , <code>lpr</code> , <code>mail</code> , <code>mark</code> , <code>news</code> , <code>security</code> , <code>syslog</code> , <code>user</code> , <code>uucp</code> , <code>local0-local7</code>) | |
| ❖ priority | :code word for the importance of this message (<code>debug</code> , <code>info</code> , <code>notice</code> , <code>warning</code> , <code>warn</code> , <code>error</code> , <code>err</code> , <code>crit</code> , <code>alert</code> , <code>emerg</code> , <code>panic</code>) | |
| ❖ action | :most commonly is a filename in <i>/var/log/</i> or <i>/dev/console</i> | |
| ❖ Examples: | | |

<code>mail.*</code>	<code>/var/log/mail</code>
<code>*.emerg</code>	<code>*</code>
<code>kernel.!error</code>	<code>/var/log/kernel</code>
<code>kern.=crit</code>	<code>/dev/console</code>
<code>kern.crit</code>	<code>@logger.myserver.edu</code>

Manually Logging Data

- ❖ **logger** allow you to manually create a log entry

`logger [-is] [-f file] [-p pri] [-t tag] [message...]`

- `-i` records the process ID (PID) of the logger process
- `-s` also echo data to standard error
- `-f file` log the contents of a file
- `-p pri` specified a priority
- `-t tag` change the tag name of the log entry
- `message` message to be logged

- ❖ **Example: \$logger shutting down for maintenance**

- The result will be an entry like the following in `/var/log/message`
`Jul 29 14:09:50 myhost logger: shutting down for maintenance`

Rotating Log Files

- ❖ **syslogd** provide no means to control the size of log files
- ❖ Linux employ *log file rotation* tools: **logrotate**
 - Typically called via *cron job* to
 - compress the current log files
 - delete old log files
 - force the logging system to begin using new log files
 - Configuration file: ***/etc/logrotate.conf***
 - Example of [/etc/logrotate.conf](#)

Reviewing Log File Contents

- ❖ Paging through whole log files with **less**
- ❖ Searching for keyword with **grep**
 - Example: **grep eth0 /var/log/***
- ❖ Examining the start or end of a file with **head** or **tail**
- ❖ Monitoring log file with **tail -f**
- ❖ Using advanced log analysis tools
(**Logcheck** or **Sentry Tools** package)

Exercise

1. Backup your current *syslog* configuration file
2. Create new **syslog.conf** file to:
 1. Direct all messages of *kernel* facility to ***/var/log/test1***
 2. Direct all messages of *info* priority level to ***/var/log/test2***
 3. Direct all messages of *local7* facility to ***/var/log/test3***
3. Restart **syslogd** service
4. Try logging in and out the system
5. Verify the new log files

Hints to Exercise

1. Backup your current *syslog* configuration file
Hint: `mv /etc/syslog.conf /etc/syslog.conf.bak`
2. Create new **syslog.conf** file to:
Hint: `vi /etc/syslog.conf`
 1. Direct all messages of **kernel** facility to **/var/log/test1**
Hint: `kernel.* /var/log/test1`
 2. Direct all messages of **info** priority level to **/var/log/test2**
Hint: `*.info /var/log/test2`
 3. Direct all messages of **local7** facility to **/var/log/test3**
Hint: `local7.* /var/log/test3`
3. Restart **syslogd** service
service syslog restart
4. Try logging in and out the system
5. Verify the new log files



3. Mail Transfer Agent (MTA) basics

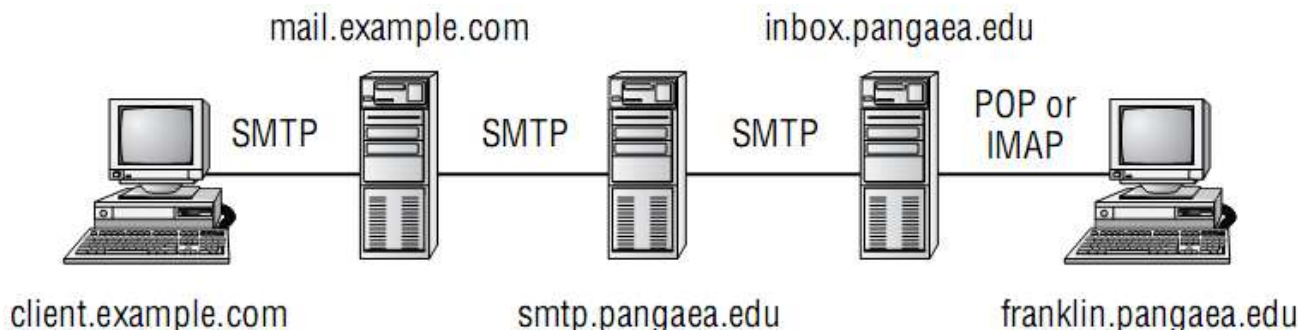
Overview

Learn to:

- ❖ Create e-mail aliases
- ❖ Configure e-mail forwarding
- ❖ Knowledge of commonly available MTA programs (**postfix**, **sendmail**, **qmail**, **exim**)
(no configuration)

Understanding Email

- ❖ Linux relies on email even in a completely non-networked environment
 - Most distro ship with email server software installed and configured for basic activities
- ❖ Protocols to manage email:
 - *SMTP*: push mail protocol, used for sending email
 - Email can be relayed through numbers of computers
 - SMTP servers are a.k.a *mail transfer agents (MTAs)*
 - *POP, IMAP*: pull mail protocol, used for receiving email



Understanding Email (cont')

- ❖ On Linux, email is tied intricately to user account
 - Incoming messages for each user is held in user's *mail spool* (`/var/spool/mail/<username>`)
- ❖ Sending email:
 - Local programs send email to the local mail server
 - Email in local mail server can be queued for later delivery
 - Local mail server send email to its outgoing mail server
- ❖ Most Linux email clients (MUAs) can directly send email to a remote SMTP server
 - Unreliable if the local network link goes down

Choosing MTA Software

- ❖ Most popular SMTP servers (MTA) on Linux
 - **Sendmail**: very popular and powerful but difficult to configure
 - **Postfix**: default email server on many distros, designed as a modular replacement for **sendmail**, easier to configure
 - **Exim**: monolithic server like **sendmail** but easier to configure
 - **qmail**: modular server with security-aware, easier to configure than **sendmail**, has strange and complicates license
- ❖ One of these servers may be installed by default
 - Find your running server:
`ps -ax | grep -e send -e post -e exim -e qmail`

Choosing MUA Software

- ❖ Pull mail servers: using POP and/or IMAP protocol. Example: Cyrus IMAP
- ❖ ***Fetchmail***: pulls mail using POP/IMAP and inject it into a local SMTP mail queue
- ❖ Mail readers: Final link in the email chain. Example: ***mail*** (installed on most distro by default), ***Evolution***, ***Kmail***, ***pine***, ***mutt***

Managing Email with *mail*

- ❖ **mail** is intended to be used on the command line or from a script
 - Some distro ship with **nail** rather than **mail**
- ❖ Sending email:
`mail [-v] [-s subj] [-c cc-addr] [-b bcc-addr] to-addr`
- ❖ Reading the *local* email queue:
`mail [-v] [-f mailspool] | -u user]`
 - Type **d** to delete email, **r** to reply email
- ❖ Example:

```
$ mail -s "Remind" -c ben@abc.com ali@abc.com
Remember the meeting at 4:00 today! [Ctrl-D]
$ mail -s "Auto alert!" < /tmp/alert.txt dany@abc.com
$ mail
0046 sally@abc.com Sun Aug 17 18:27 16/464 New job
```

Checking the Email queue

- ❖ Email queue may contain undelivered messages
- ❖ Shows the contents of the queue: `mailq` or `sendmail -bp`
- ❖ Clear the queue immediately when network connection has come up again:
`sendmail -q` *#for most SMTP server*
`postqueue` *#for Postfix*
`runq` *#for Exim*

Redirecting Email

- ❖ Email *aliases* enable one address to stand in for another one
 - Eg: **postmaster** account is the alias to the account of mail server administrator
- ❖ **aliases** file usually resides in */etc* or */etc/mail/*
 - Syntax: **name: addr1[,addr2[,...]]**
 - Example:
postmaster: mailadm
root: fred@abc.com, include:/tmp/admins.list
local: /tmp/local.email, -/opt/forwarder.sh
 - Some mail server require to compile */etc/aliases* into a binary file with **newaliases** command
- ❖ Email redirecting can be done on the user level by editing *~/.forward*

Securing Your Email Server

❖ Potentil security risks to your email server:

- **Bugs:** someone can connecting to SMTP port (25) via Telnet and typing SMTP commands to trigger the bug
- **Misconfiguration:** Most common is an open relay

❖ Securing your email server:

- Limit access to the email server
- Ensure that your email server is up to date
- Configure your email server to be not open relays
 - Check <http://www.abuse.net/relay.html> to verify your system is not open relays

Review Questions

1. Which of the following is not a popular SMTP server for Linux: ***Postfix, Sendmail, Fetchmail, Exim***
2. Your Internet connection has gone down for several hours. What happen to an email sent by your users to off-site recipients via a properly configured local SMTP server?
3. What is the effect of the following line in a script:
`mail -s "Error" -c abort < /tmp/msg root`
4. You examine your ***/etc/aliases*** file and find the following line:
`root: jody`
What can you conclude from this?

Exercise

1. Find your running email server
 - `ps -ax | grep -e send -e post -e exim -e qmail`
2. Use an account other than **root**, send an email to **root** with the subject is “*Disks usage*” and content is the output of **df -h** command
 - `echo `df -h` | mail -s “Disks usage” root`
3. Login as **root** and read your email, view and delete the email you received from step 2.



4. Manage printers and printing

Overview

Learn to:

- ❖ Basic CUPS configuration (for local and remote printers)
- ❖ Manage user print queues
- ❖ Troubleshoot general printing problems
- ❖ Add and remove jobs from configured printer queues

The Linux Printing Architecture

- ❖ *Print queue*: holding area where files wait to be printed
 - Print queue directory: typically */var/spool/cups*
 - Each printer can use different queues
- ❖ Users submit print jobs to a queue by using *lpr*
- ❖ Printing system's works:
 - Accept print jobs from *lpr* or from remote computer
 - Even local print jobs are submitted via network protocol
 - Monitor print queues
 - Direct print jobs orderly from print queues to printers
- ❖ Linux common printing system: LPD, LPRng, CUPS
 - CUPS can function as *server* or *client* for print jobs
 - Application can query CUPS about printer's capabilities

PostScript and Ghostscript

❖ **PostScript:** the de-facto Linux printer language

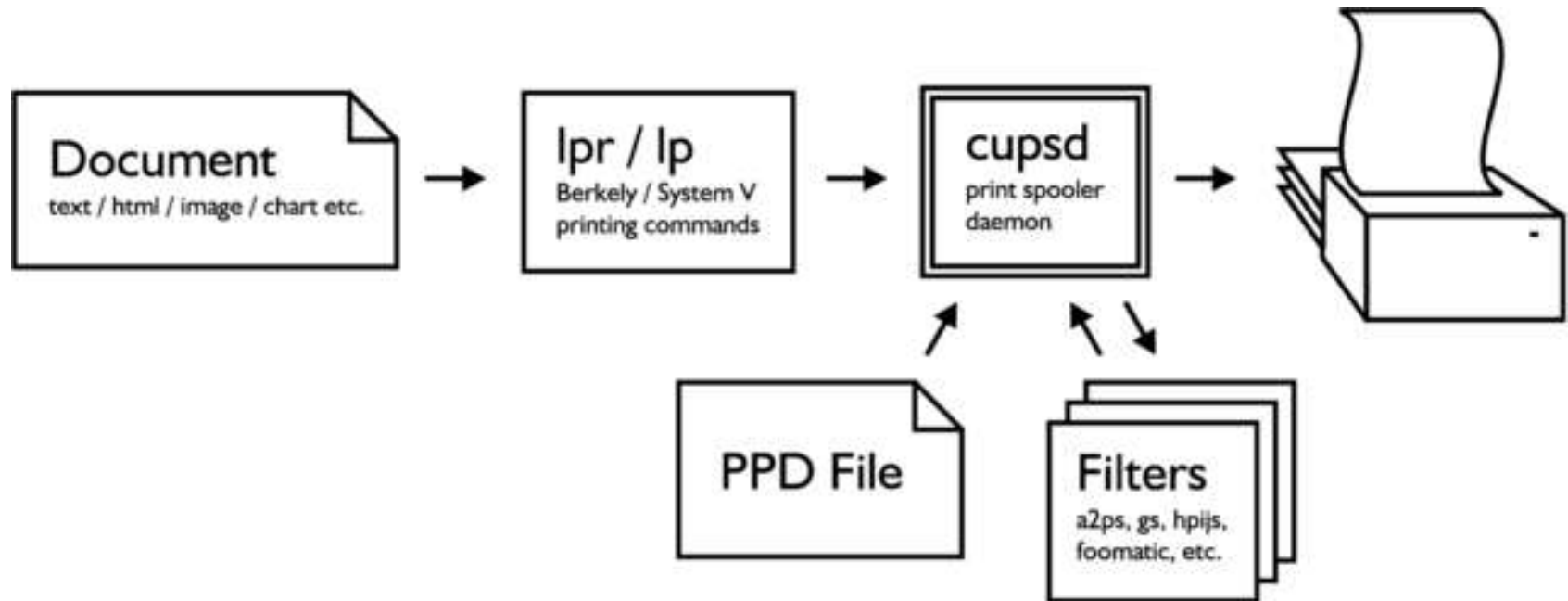
- Most programs on Linux generate PostScript and send the result to the print queue
- PostScript is uncommon on the low- and mid-priced printers

❖ **Ghostscript:** the PostScript interpreter

- Takes PostScript input then produces output for non-PostScript printers
- Printer driver is part of Ghostscript

❖ Printing System's ***Smart Filter*** pass printing files to Ghostscript (if needed) before sending to print queue

How printing works



Running a Printing System (CUPS)

- ❖ CUPS is automatically started via startup scripts
 - View running CUPS: **ps ax | grep cups**
- ❖ CUPS uses **IPP** (Internet Printing Protocol) in addition to the older **LPD** protocol
 - IPP supports *browsing* feature for network printing
- ❖ Configuration files for CUPS: in ***/etc/cups/***
 - Configure CUPS Web-based interface tool: edit ***/etc/cups/cupsd.conf***
 - Add or delete printers: edit ***/etc/cups/printers.conf***
- ❖ Obtaining CUPS Printer Definitions: Your Linux distro, **Foomatic**, **Gutenprint**, **CUPS DDK** or from your printer manufacturers

Controlling the Print Queue

- ❖ Printing files:

`lpr [-Pqueue] [-# copy] filename`

- ❖ Displaying print queue:

`lpq [-Pqueue] [username]`

- ❖ Removing print jobs:

`lprm jobID`

- ❖ Controlling the print queue: **cupsenable**, **cupsdisable**, **lpmove**

- Normally done by using CUPS Web interface

Exercise

1. Verify that ***cupsd*** daemon is running
 - `ps ax | grep cupsd`
2. View the current listening port of CUPS Web-based interface
 - `grep Listen /etc/cups/cupsd.conf`
3. Open ***firefox*** and access to the CUPS Web-based interface (using the port found in step 2)
4. View your CUPS Configuration file from Web-based interface
5. Enable remote administration to CUPS Web-based interface
 - Select ***Allow remote administration*** in ***Administration*** page
 - Verify that you can connect to CUPS Web-based interface from other computer
6. Add a new pseudo printer
 - Name: ***pseudoprt*** Device: ***Windows Printer via SAMBA***
 - Device URI: ***smb://admin:password@192.168.150.1/shareprt***
 - Make: <Any> Model: <Any>
7. Print a test page with your newly added printer
8. Remove this print job from queue
9. Remove your pseudo printer



Thank You !



BACKUP SLIDES

Sample */etc/logrotate.conf*

```
# Rotate logs weekly
weekly
# Keep 4 weeks of old logs
rotate 4
# Create new log files after rotation
create
# Compress old log files
compress
# Refer to files for individual packages
include /etc/logrotate.d
# Set miscellaneous options
notifempty
nomail
noolddir
# Rotate wtmp, which isn't handled by a specific program
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}
```