LPI 108.2 - System logging

Curs 2021 - 2022

ASIX M01-ISO 108 Essential System Services

System loggings

Description

Key concepts:

- ❏ Basic configuration of rsyslog.
- ❏ Understanding of standard facilities, priorities and actions.
- ❏ Query the systemd journal.
- ❏ Filter systemd journal data by criteria such as date, service or priority.
- ❏ Configure persistent systemd journal storage and journal size.
- ❏ Delete old systemd journal data.
- ❏ Retrieve systemd journal data from a rescue system or file system copy.
- ❏ Understand interaction of rsyslog with systemd-journald.
- ❏ Configuration of logrotate.
- ❏ Awareness of syslog and syslog-ng.

Commands and files:

- ❏ /etc/rsyslog.conf
- ❏ /var/log/
- ❏ logger
- ❏ logrotate
- ❏ /etc/logrotate.conf
- ❏ /etc/logrotate.d/
- ❏ journalctl
- ❏ systemd-cat
- ❏ /etc/systemd/journald.conf
- ❏ /var/log/journal/

System Logging

System logging is the process of capturing most everything that happens on and to the Linux system and sending the information to log files to be viewed later. Many Linux distributions have replaced the combination of the syslogd and klogd daemons with the more recently developed rsyslogd daemon. The rsyslogd daemon configuration settings are stored in the */etc/rsyslog.conf* file. The daemon name is rsyslogd, the service name is rsyslog. rsyslog uses a client-server model. The client and the server can live on the same host or in different machines.

Usual files and directories:

/var/log

Main log directory.

/etc/rsyslog.conf

rsyslogd daemon file configuration

/var/log/messages

/var/log/rsyslog

General message and system-related information

/var/log/secure

var/log/auth.log

Authentication log

/var/log/maillog

Mail server logs

/var/log/kern.log

Kernel logs

/var/log/boot.log

System boot log

/var/log/cron.log

crond logs

/var/log/Xorg.0.log

Information related to the graphics card.

/var/run/utmp and /var/log/wtmp

Successful logins.

/var/log/btmp

Failed login attempts, e.g. brute force attack via ssh.

/var/log/faillog

Failed authentication attempts.

/var/log/lastlog

Date and time of recent user logins.

```
pue@debian:~$ sudo tail /var/log/messages
Nov  8 19:16:28 debian gnome-software[1759]: hiding category productivity featured applications: found only 0 to show, need at least 9
Nov  8 19:16:28 debian gnome-software[1759]: hiding category games featured applications: found only 0 to show, need at least 9
Nov  8 19:16:29 debian gnome-software[1759]: Only 8 apps for popular list, hiding
Nov  8 19:16:32 debian org.gnome.Terminal.desktop[1567]: # watch_fast: "/org/gnome/terminal/legacy/" (establishing: 0, active: 0)
Nov  8 19:16:32 debian org.gnome.Terminal.desktop[1567]: # unwatch_fast: "/org/gnome/terminal/legacy/" (active: 0, establishing: 1)
Nov  8 19:16:32 debian org.gnome.Terminal.desktop[1567]: # watch_established: "/org/gnome/terminal/legacy/" (establishing: 0)
Nov  8 19:17:18 debian geoclue[1308]: Service not used for 60 seconds. Shutting down..
Nov  8 19:17:25 debian gnome-shell[1567]: pushModal: invocation of begin_modal failed
Nov  8 19:17:25 debian gnome-shell[1567]: polkitAuthenticationAgent: Failed to show modal dialog. Dismissing authentication request for
action-id org.freedesktop.packagekit.system-sources-refresh cookie 2-03ba2e3c705fd3914ea05a42ebe7ea63-2-8758b78794b903e19a1a874195783d91
Nov  8 19:17:26 debian gnome-software[1759]: shell-extensions did not set error for gs_plugin_refresh

pue@debian:~$ sudo tail -f /var/log/auth.log
Nov  8 19:26:09 debian sudo:      pue : TTY=pts/0 ; PWD=/home/pue ; USER=root ; COMMAND=/usr/bin/less /var/log/messages
Nov  8 19:26:09 debian sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Nov  8 19:26:48 debian su: pam_unix(su-l:auth): authentication failure; logname= uid=1000 euid=0 tty=pts/1 ruser=pue
rhost=  user=root
Nov  8 19:26:50 debian su: FAILED SU (to root) pue on pts/1
Nov  8 19:26:55 debian sudo: pam_unix(sudo:session): session closed for user root
Nov  8 19:27:03 debian sudo:      pue : TTY=pts/0 ; PWD=/home/pue ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/messages
Nov  8 19:27:03 debian sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Nov  8 19:27:11 debian su: pam_unix(su-l:auth): authentication failure; logname= uid=1000 euid=0 tty=pts/1 ruser=pue
rhost=  user=root
Nov  8 19:27:13 debian su: FAILED SU (to root) pue on pts/1
Nov  8 19:27:24 debian sudo: pam_unix(sudo:session): session closed for user root
Nov  8 19:28:55 debian sudo:      pue : TTY=pts/0 ; PWD=/home/pue ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/auth.log
Nov  8 19:28:55 debian sudo: pam_unix(sudo:session): session opened for user root by (uid=0)

pue@debian:~$ sudo head -n20 /var/log/boot.log
/dev/vda1: clean, 141081/907536 files, 992790/3629312 blocks
         Starting Create Volatile Files and Directories...
[  OK  ] Started Create Volatile Files and Directories.
         Starting Update UTMP about System Boot/Shutdown...
         Starting Network Time Synchronization...
[  OK  ] Started Update UTMP about System Boot/Shutdown.
```

```
         Starting Show Plymouth Boot Screen...
         Starting Tell Plymouth To Write Out Runtime Data...
[  OK  ] Started Tell Plymouth To Write Out Runtime Data.
[  OK  ] Started Show Plymouth Boot Screen.
[  OK  ] Started Network Time Synchronization.
[  OK  ] Reached target System Time Synchronized.
[  OK  ] Started Load AppArmor profiles.
[  OK  ] Reached target System Initialization.
[  OK  ] Started Daily man-db regeneration.
[  OK  ] Started Trigger anacron every hour.
[  OK  ] Started Daily apt download activities.
[  OK  ] Started Daily apt upgrade and clean activities.
[  OK  ] Started Daily rotation of log files.
[  OK  ] Listening on CUPS Scheduler.
```

```
pue@debian:~$ ls /var/log/
alternatives.log    boot.log         daemon.log.3.gz  exim4           kern.log.1     messages.2.gz      syslog.3.gz          user.log.1
alternatives.log.1  btmp             debug            faillog         kern.log.2.gz  messages.3.gz      syslog.4.gz          user.log.2.gz
apt                 btmp.1           debug.1          fontconfig.log  kern.log.3.gz  private            syslog.5.gz          user.log.3.gz
auth.log            cups             debug.2.gz       gdm3            lastlog        speech-dispatcher  syslog.6.gz          wtmp
auth.log.1          daemon.log       debug.3.gz       hp              libvirt        syslog             syslog.7.gz          xrdp.log
auth.log.2.gz       daemon.log.1     dpkg.log         installer       messages       syslog.1           unattended-upgrades
xrdp-sesman.log
auth.log.3.gz       daemon.log.2.gz  dpkg.log.1       kern.log        messages.1     syslog.2.gz        user.log
```

Examples of service logs:

/var/log/cups/

Directory for logs of the Common Unix Printing System. It commonly includes the following default log files: error_log, page_log and access_log.

/var/log/apache2/ or /var/log/httpd

Directory for logs of the Apache Web Server. It commonly includes the following default log files: access.log, error_log, and other_vhosts_access.log.

/var/log/mysql

Directory for logs of the MySQL Relational Database Management System. It commonly includes the following default log files: error_log, mysql.log and mysql-slow.log.

/var/log/samba/

Directory for logs of the Session Message Block (SMB) protocol. It commonly includes the following default log files: log., log.nmbd and log.smbd.

Utilities to read logs:

all text linux text comamns: head, tail, cat, grep, zcat...

As you may have noticed, the output is printed in the following format:

- Timestamp
- Hostname from which the message originated
- Name of program/service that generated the message
- The PID of the program that generated the message
- Description of the action that took place

```
Nov  5 16:45:01 localhost rsyslogd[1151]: [origin software="rsyslogd"
```

```
[root@localhost pue]# last | tail
reboot   system boot  4.18.0-147.el8.x Tue Sep 29 12:32 - 12:34  (00:01)
reboot   system boot  4.18.0-147.el8.x Tue Sep 29 12:29 - 12:30  (00:01)
pue      :1           :1               Tue Sep 29 12:27 - down   (00:01)
reboot   system boot  4.18.0-147.el8.x Tue Sep 29 12:27 - 12:28  (00:01)
pue      :1           :1               Tue Sep 29 12:22 - down   (00:04)
reboot   system boot  4.18.0-147.el8.x Tue Sep 29 12:22 - 12:27  (00:04)
pue      :1           :1               Tue Sep 29 12:18 - down   (00:03)
reboot   system boot  4.18.0-147.el8.x Tue Sep 29 12:16 - 12:22  (00:06)
```

```
[root@localhost ~]# lastlog  | head
Username         Port     From             Latest
root             pts/0                     Mon Nov  8 21:01:06 +0100 2021
bin                                        **Never logged in**
daemon                                     **Never logged in**
adm                                        **Never logged in**

[root@localhost ~]# utmpdump /var/log/wtmp  | head
Utmp dump of /var/log/wtmp
[2] [00000] [~~  ] [reboot  ] [~          ] [4.18.19-100.fc27.x86_64] [0.0.0.0       ] [2021-04-07T15:02:51,997582+00:00]
[1] [00053] [~~  ] [runlevel] [~          ] [4.18.19-100.fc27.x86_64] [0.0.0.0       ] [2021-04-07T15:02:54,260088+00:00]
[7] [01509] [    ] [guest   ] [tty2       ] [:1                    ] [0.0.0.0       ] [2021-04-07T15:03:03,161375+00:00]
[1] [00000] [~~  ] [shutdown] [~          ] [4.18.19-100.fc27.x86_64] [0.0.0.0       ] [2021-04-07T15:09:34,898565+00:00]
[2] [00000] [~~  ] [reboot  ] [~          ] [4.18.19-100.fc27.x86_64] [0.0.0.0       ] [2021-04-07T15:17:23,428529+00:00]
```

rsyslogd configuration

For describing what will be logged, the configuration file uses a *selector*. The selector is made up of two parts: a *facility* and a *priority*, separated by a period . character. An *action* is used to describe where to send the log information. Each line of the configuration file will specify both a selector and an action.

By default, the rsyslogd daemon only accepts the logs which are generated by the localhost. If an administrator wants to enable a host to be a central log server, which is a server that is designated as a logging host for clients, they can edit the /etc/rsyslogd.conf file. the UDP port 514 is used by the log server to receive messages from remote systems and should be open only to legitimate systems.

After setting the rsyslogd options, the logging service would need to be restarted or the system rebooted for the change to take effect.

```
pue@debian:~$ systemctl  status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-11-08 19:16:13 CET; 1min 2s ago
     Docs: man:rsyslogd(8)
           https://www.rsyslog.com/doc/
 Main PID: 508 (rsyslogd)
    Tasks: 4 (limit: 4683)
   Memory: 2.8M
   CGroup: /system.slice/rsyslog.service
           └─508 /usr/sbin/rsyslogd -n -iNONE
```

```
pue@debian:~$ tree /etc/rsyslog.d/
/etc/rsyslog.d/
0 directories, 0 files
```

```
pue@debian:~$ cat /etc/rsyslog.conf
...
# Rules
auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log
mail.info                       -/var/log/mail.info
```

```
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err
# Emergencies are sent to everybody logged in.
*.emerg                         :omusrmsg:*
```

Syslog rules

```
facility.priority       action
```

Facility

The facility identifies the part of the system that produced some kind of message.

auth
: Security and authorization-related commands

authpriv
: Private authorization messages

cron
: The cron daemon

daemon
: System daemons

ftp
: The ftp daemon

kern
: The kernel

lpr
: The BSD printer spooling system

mail
: sendmail and other mail-related software

mark
: Timestamps generated at regular intervals

news
: The Usenet news system

security
: Same as auth

rsyslog
: rsyslogd internal messages

user
: User processes

uucp
: Reserved for UUCP

local0 to local7
: Eight flavors of local message

Example facilities

```
auth,authpriv.*                         /var/log/auth.log
cron.*                          /var/log/cron.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
```

Priority

Defines the severity of the message. Priority is ordered from lowest to highest in this order: `debug > info > notice > warning > err > crit > alert > emerg`. Priorities that are specified mean not only the level specified but anything of higher priority, as well. For example, specifying a priority of err would not only log err level messages, but also crit, alert, and emerg level messages.

none
: do not log from that facility.

debug
: For debugging only

info
: Informational messages

notice
: Things that might merit investigation

warning (or warn)
: Warning messages

err
: Other error conditions

crit
: Critical conditions

alert
: Urgent situations

emerg (or panic)
: Panic situations

Example priorities

```
auth,authpriv.*                    /var/log/auth.log
cron.err                   /var/log/cron.log
lpr.waring                 -/var/log/lpr.log
mail.info                  -/var/log/mail.log
```

Selector

The selector is comprised of both the facility and the priority separated by a period . character. An asterisk * wildcard character can be used to represent either all facilities or all priorities in a selector:

Selector examples

```
*.*            All facilities and priorities
*.info         All facilities at info priority or higher
kern.*         Select all kernel messages
mail.warning   Messages from the mail facility at a warning priority or higher
cron,lpr.err   Messages from the cron or lpr facility at an err priority or higher
cron.err;cron.!alert
               From the cron at an err priority or higher, but not at alert
```

```
priority
mail.=err      Only err messages from the mail facility
*.info;mail.none;lpr.none
               Select messages from all facilities except mail and lpr
```

Action

Combining a selector with an action results in a complete rule line in the /etc/rsyslog.conf file. The most common action is to specify the absolute path, the file that will store the information that is selected. The following table demonstrates the available actions:

/path/to/file
: Specify the full absolute path for the log file

-/path/to/file
: The - before the path means to not sync after writing each log message (better for system performance for log files that are often written to, such as mail log files on a mail server)

|/path/to/named/pipe
: Specify a pipe symbol and a path to a named pipe file created with mkfifo

/dev/tty10
: Specify a terminal or console, such as /dev/console

@10.0.0.1
: Specify an @ symbol with the IP address or resolvable hostname or a remote host

student,maya,joe
: Specify a list of users whose terminals will have the message displayed if the users are currently logged into the system

*
: Send to the terminal of everyone who is logged in

Rules examples

```
# Rules
auth,authpriv.*                     /var/log/auth.log
*.*;auth,authpriv.none              -/var/log/syslog
#cron.*                        /var/log/cron.log
daemon.*                       -/var/log/daemon.log
kern.*                         -/var/log/kern.log
lpr.*                          -/var/log/lpr.log
mail.*                         -/var/log/mail.log
user.*                         -/var/log/user.log
mail.info                      -/var/log/mail.info
mail.warn                      -/var/log/mail.warn
mail.err                       /var/log/mail.err
*.emerg                        :omusrmsg:*
```

Sending logs to a remote log server

To send the logs to a remote central server (one or more) an example configuration /etc/rsyslog.conf file can be the next one. All the facilities and from warning priority and higher are sent to 192.168.254 port 514.

```
*.warning              @192.168.0.254:514
```

The logger command

The logger command is used to send messages to the system logging facility. The following options can be used with logger:

-i	Log the process id of the logger process
-s	Log the message to standard error and the system log
-f file	Use the message found in the specified file
-p selector	Send the message as the selector like mail.info
-t tag	Mark the message line in the log with a tag

One of the main uses of the logger command is to verify that the entries that have been made in the rsyslog.conf file are working as expected.

```
(2)# logger -t TEST -p mail.err 'Testing mail.err entry'

root@debian:/home/pue# tail -f /var/log/mail.log
Nov  8 20:14:10 debian TEST: testing mail
Nov  8 20:14:26 debian TEST: testing mail


(2)# logger -t TEST -p mail.err "testing mail"
```

Rsyslog management

Most of the information logged on a Linux system is useful for a limited amount of time. Since log files grow over time, an administrator should institute logging policies that determine what to do with the log files and how often to take action.

The *logrotate* tool is used to allow a system administrator to automate the rotation of log files with different settings for different services.

- /etc/logrotate.conf
- /etc/logrotate.d

Logs are rotated on a regular basis, which serves two main purposes:

- Prevent older log files from using more disk space than necessary.
- Keep logs to a manageable length for ease of consultation.

```
# tree /etc/logrotate.d/
/etc/logrotate.d/
├── alternatives
├── apt
├── btmp
```

```
├── cups-daemon
├── dpkg
├── exim4-base
├── exim4-paniclog
├── libvirtd
├── libvirtd.libxl
├── libvirtd.lxc
├── libvirtd.qemu
├── ppp
├── rsyslog
├── speech-dispatcher
├── unattended-upgrades
└── wtmp
```

```
# cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
#dateext

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# system-specific logs may be also be configured here.
```

```
[root@localhost pue]# ls /var/log/messages*
/var/log/messages           /var/log/messages-20211011  /var/log/messages-20211105
/var/log/messages-20210929  /var/log/messages-20211017
```

Settings description:

weekly/daily/monthly/yearly

Rotates the logs at the specified time interval

rotate 4

Determines how many rotated logs are kept before logrotate deletes older logs

compress

Specifies logrotate to compress rotated logs

missingok

Tells logrotate not to return an error if the log file is not found

notifempty

Do not rotate the log if it is empty.

postrotate

Indicate the beginning of a postrotate script.

include /etc/logrotate.d

Files in the /etc/logrotate.d directory are loaded and allow the system administrator to have different configurations for the logs of different services.

```
# cat /etc/logrotate.d/apt
/var/log/apt/term.log {
  rotate 12
  monthly
  compress
  missingok
  notifempty
}

/var/log/apt/history.log {
  rotate 12
  monthly
  compress
  missingok
  notifempty
}
```

```
# cat /etc/logrotate.d/alternatives
/var/log/alternatives.log {
        monthly
        rotate 12
        compress
        delaycompress
        missingok
        notifempty
        create 644 root root
}
```

Systemd journal

On systems using systemd as their init system, rsyslog has been replaced by the *systemd-journal*, it handles logs from other systemd units. Using the systemd-journal, the log information is written to binary databases in the /var/log/journal/ directory instead of writing to text files like rsyslog. To interpret and present the logs in a human-readable format the command *journalctl* is used.

- /etc/systemd/journald.conf
- /var/log/journal
- persistent storage (10% < 4GB)
- volatile memory

The /etc/systemd/journald.conf file controls the systemd-journal, but the most used directive controls the amount of space that is used for storing persistent logs found in /var/log/journal/ if the directory exists. Otherwise, the systemd-journal stores logs in volatile memory (RAM) located at /run/log/journal.

Persistent storage is a type of storage used to ensure that data is not modified after it is stored and is available even if updates are made to the storage software. Files stored in volatile memory disappear when a computer is reset.

journald advantages:

- It centralizes all logs in one place.
- It does not require log rotation.
- Logs can be disabled, loaded in RAM or made persistent.

journal.conf configuration directives:

Storage

Determines how the journal will be stored. The volatile option keeps the journal only in memory. The persistent option stores the log data on the disk. The auto option stores to the disk also, but will not create a log file if it doesn't already exist. The none option does not store the journal data, but only displays it on the console.

Compress

Specifies if the journal logs should be compressed or not.

SystemMaxUse

Limits the amount of space a log can use on the disk. By default, the limit is set to 10% of the total disk space with a cap of 4GB.

SystemMaxFileSize

Specifies the maximum size that an individual journal file can be before the file is rotated.

```
# cat /etc/systemd/journald.conf
[Journal]
#Storage=auto
#Compress=yes
#Seal=yes
#SplitMode=uid
#SyncIntervalSec=5m
#RateLimitIntervalSec=30s
#RateLimitBurst=10000
#SystemMaxUse=
#SystemKeepFree=
#SystemMaxFileSize=
#SystemMaxFiles=100
#RuntimeMaxUse=
#RuntimeKeepFree=
#RuntimeMaxFileSize=
#RuntimeMaxFiles=100
#MaxRetentionSec=
#MaxFileSec=1month
#ForwardToSyslog=no
#ForwardToKMsg=no
#ForwardToConsole=no
#ForwardToWall=yes
#TTYPath=/dev/console
#MaxLevelStore=debug
#MaxLevelSyslog=debug
#MaxLevelKMsg=notice
#MaxLevelConsole=info
#MaxLevelWall=emerg
#LineMax=48K
```

```
[root@localhost pue]# systemctl status systemd-journald.service
● systemd-journald.service - Journal Service
   Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static; vendor
preset: disabled)
   Active: active (running) since Mon 2021-11-08 20:41:03 CET; 13min ago
     Docs: man:systemd-journald.service(8)
           man:journald.conf(5)
 Main PID: 733 (systemd-journal)
   Status: "Processing requests..."
    Tasks: 1 (limit: 23548)
   Memory: 3.8M
   CGroup: /system.slice/systemd-journald.service
           └─733 /usr/lib/systemd/systemd-journald
```

```
[root@localhost ~]# ls -lh /var/log/journal/f8bff953853f49b4963ed3aa06461c80/
-rw-r-----+ 1 root systemd-journal  56M Jun  9 22:58 system@0005c45b89917386-76f8379964536930.journal~
-rw-r-----+ 1 root systemd-journal  16M Jul  3 13:20 system@0005c63641fbbdcf-a4e5880ec760faaf.journal~
-rw-r-----+ 1 root systemd-journal  16M Aug 14 19:34 system@0005c98860b34b71-947731060d70fd2a.journal~
-rw-r-----+ 1 root systemd-journal  64M Aug  3 21:16
system@303a26ba4b2f4be88d7f4f4c7f710f4a-0000000000000001-0005c63641cc8043.journal
```

```
-rw-r-----+ 1 root systemd-journal  56M Jun 24 14:50
system@92bfbab595744f528c91bc1f51e4ed48-0000000000000001-0005c45b8961116f.journal
-rw-r-----+ 1 root systemd-journal  40M Sep  3 21:13
system@cdaf58ca5ec946b0800a1011aa98c510-0000000000000001-0005c98860805dff.journal
-rw-r-----+ 1 root systemd-journal  88M Oct  4 16:04
system@cdaf58ca5ec946b0800a1011aa98c510-0000000000023eaa-0005cb1c19c60bb8.journal
-rw-r-----+ 1 root systemd-journal 112M Oct 24 14:09
system@cdaf58ca5ec946b0800a1011aa98c510-000000000004008f-0005cd8763ef7aa6.journal
-rw-r-----+ 1 root systemd-journal 104M May 24 22:57
system@fe7ff0ca35264a4d8afa9392dacb576a-00000000000332c0-0005c0e1778f1b38.journal
-rw-r-----+ 1 root systemd-journal  56M Nov  8 21:07 system.journal
...
```

journalctl management

To interact with the systemd-journald, the journalctl command is used. The output from the *journalctl* command uses a pager by default, unless --no-pager option is used.

Journalctl general options:

-b --boot
: Limits output to only journal data since the last time the system booted. To see log messages from previous boots, just add an offset (0 refers to the current boot, -1 is the previous one, -2 the one prior to the previous one and so on).

-u <systemd unit>
: Limits output to only contain output from the specified systemd unit. An example would be “journalctl -u postfix”.

-n <number>
: Shows only the last of lines specified.

-r
: Reverses chronology. Shows logs with the newest first and then each older entry in order.

-f
: it will print the most recent journal messages and keep printing new entries as they are appended to the journal — much like tail -f:

-e
: It will jump to the end of the journal so that the latest entries are visible within the pager:

--no-pager
: turn off pager

-k, --dmesg
: Equivalent to using the dmesg command

--list-boots
: It lists all available boots.

-p
: you can also filter by severity/priority with the -p option.

--since
--until
: Print only the messages logged within a specific time frame. The date specification should follow the format YYYY-MM-DD HH:MM:SS. Midnight will

be assumed if we omit the time component. By the same token, if the date is omitted, the current day is assumed.

/path/to/executable

To see journal messages related to a specific executable.

<field-name>=<value>

<field-name>=<value>

__<field-name>=<value>

The journal can also be filtered by specific fields: PRIORITY, SYSLOG_FACILITY, _PID, _BOOT_ID, _TRANSPORT (Possible values are: audit (kernel audit subsystem), driver (generated internally), syslog (syslog socket), journal (native journal protocol), stdout (services' standard output or standard error), kernel (kernel ring buffer — the same as dmesg, journalctl -k or journalctl --dmesg)

Examples:

```
# journalctl -n 12

# journalctl -b -r -u httpd

# journalctl -b -1 --no-pager

# journalctl -p err

# journalctl --since "today" --until "21:00:00"

# journalctl /usr/sbin/sshd

# journalctl PRIORITY=3

# journalctl SYSLOG_FACILITY=1

# journalctl _PID=1

# journalctl _TRANSPORT=audit
```

```
[root@localhost pue]# journalctl -b -1 --no-pager
Specifying boot ID or boot offset has no effect, no persistent journal was found.
```

```
[root@localhost ~]# journalctl --no-pager -b -1 | head
-- Logs begin at Wed 2021-04-07 17:02:51 CEST, end at Tue 2022-01-25 12:07:28 CET. --
Oct 30 17:40:21 localhost.localdomain kernel: Linux version 5.11.22-100.fc32.x86_64
(mockbuild@bkernel01.iad2.fedoraproject.org) (gcc (GCC) 10.3.1 20210422 (Red Hat 10.3.1-1), GNU ld version 2.34-6.fc32) #1
SMP Wed May 19 18:58:25 UTC 2021
Oct 30 17:40:21 localhost.localdomain kernel: Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-5.11.22-100.fc32.x86_64
root=/dev/mapper/fedora-root ro rd.lvm.lv=fedora/root rd.lvm.lv=fedora/swap rhgb quiet
Oct 30 17:40:21 localhost.localdomain kernel: x86/split lock detection: warning about user-space split_locks
Oct 30 17:40:21 localhost.localdomain kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Oct 30 17:40:21 localhost.localdomain kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Oct 30 17:40:21 localhost.localdomain kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
Oct 30 17:40:21 localhost.localdomain kernel: x86/fpu: Supporting XSAVE feature 0x020: 'AVX-512 opmask'
Oct 30 17:40:21 localhost.localdomain kernel: x86/fpu: Supporting XSAVE feature 0x040: 'AVX-512 Hi256'
Oct 30 17:40:21 localhost.localdomain kernel: x86/fpu: Supporting XSAVE feature 0x080: 'AVX-512 ZMM_Hi256'
```

```
[root@localhost pue]# journalctl -u crond
-- Logs begin at Mon 2021-11-08 20:40:58 CET, end at Mon 2021-11-08 21:01:01 CET. --
nov 08 20:41:05 localhost.localdomain systemd[1]: Started Command Scheduler.
nov 08 20:41:05 localhost.localdomain crond[1168]: (CRON) STARTUP (1.5.2)
nov 08 20:41:05 localhost.localdomain crond[1168]: (CRON) INFO (Syslog will be used instead of sendmail.)
nov 08 20:41:05 localhost.localdomain crond[1168]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 57% if used.)
nov 08 20:41:05 localhost.localdomain crond[1168]: (CRON) INFO (running with inotify support)
nov 08 21:01:01 localhost.localdomain CROND[7693]: (root) CMD (run-parts /etc/cron.hourly)
nov 08 21:01:01 localhost.localdomain anacron[7702]: Anacron started on 2021-11-08
nov 08 21:01:01 localhost.localdomain anacron[7702]: Will run job `cron.daily' in 24 min.
nov 08 21:01:01 localhost.localdomain anacron[7702]: Jobs will be executed sequentially
```

Systemd-cat

Since systemd-journald stores data in a binary database, instead of text files, adding data to the logs requires the use of the tool systemd-cat (like logger for rsyslog).

```
[root@localhost pue]# uptime | systemd-cat
[root@localhost pue]# uptime | systemd-cat
[root@localhost pue]# uptime | systemd-cat

[root@localhost pue]# journalctl -r
-- Logs begin at Mon 2021-11-08 20:40:58 CET, end at Mon 2021-11-08 21:21:20 CET. --
nov 08 21:21:20 localhost.localdomain unknown[8333]:  21:21:20 up 40 min,  0 users,  load average: 0,04, 0,02, 0,02
nov 08 21:21:18 localhost.localdomain cat[8331]:  21:21:18 up 40 min,  0 users,  load average: 0,04, 0,02, 0,02
nov 08 21:21:15 localhost.localdomain unknown[8329]:  21:21:15 up 40 min,  0 users,  load average: 0,04, 0,02, 0,02
nov 08 21:20:04 localhost.localdomain unknown[8277]:  21:20:04 up 39 min,  0 users,  load average: 0,02, 0,01, 0,02
nov 08 21:19:46 localhost.localdomain xrdp-chansrv[6189]: [ERROR] clipboard_event_selection_request: unknown target
text/plain>
nov 08 21:18:41 localhost.localdomain NetworkManager[974]: <info>  [1636402721.0719] agent-manager:
agent[0cf0ac447e803f7e,:1.>
nov 08 21:11:15 localhost.localdomain systemd[1]: NetworkManager-dispatcher.service: Succeeded.
```

```
# systemd-cat
write to stdin and appears in journald
```

```
$ systemd-cat -p emerg echo "This is not a real emergency."

$ journalctl -n3
```

Persistent Storage

There are three options when it comes to the location of the journal:

- Journaling can be turned off altogether (redirection to other facilities such as the console are still possible, though).
- Keep it in memory — which makes it volatile — and get rid of the logs with every system reboot. In this scenario, the directory /run/log/journal will be created and used.
- Make it persistent so that it writes logs to disk. In this case, log messages will go into the /var/log/journal directory.

The default behaviour is as follows: if /var/log/journal/<machine-id> does not exist, logs will be saved in a volatile way to a directory in /run/log/journal/<machine-id> and — therefore — lost at reboot.

If /var/log/journal/ exists, logs will be stored persistently there. Should this directory be deleted, systemd-journald would not recreate it but write to /run/log/journal instead. As soon as we create /var/log/journal/ again and restart the daemon, persistent logging will be restablished.

Persistent storage example:

```
[root@localhost ~]# ls /run/log/journal/

[root@localhost ~]# ls /var/log/journal/
f8bff953853f49b4963ed3aa06461c80

[root@localhost ~]# tree /var/log/journal/
/var/log/journal/
└── f8bff953853f49b4963ed3aa06461c80
```

```
    ├── system@0005c45b89917386-76f8379964536930.journal~
    ├── system@0005c63641fbbdcf-a4e5880ec760faaf.journal~
    ├── system@0005c98860b34b71-947731060d70fd2a.journal~
    ├──
system@303a26ba4b2f4be88d7f4f4c7f710f4a-0000000000000001-0005c63641cc8043.journal
    ├──
system@92bfbab595744f528c91bc1f51e4ed48-0000000000000001-0005c45b8961116f.journal
    ├──
system@cdaf58ca5ec946b0800a1011aa98c510-0000000000000001-0005c98860805dff.journal
    ├──
system@cdaf58ca5ec946b0800a1011aa98c510-0000000000023eaa-0005cb1c19c60bb8.journal
    ├──
system@cdaf58ca5ec946b0800a1011aa98c510-000000000004008f-0005cd8763ef7aa6.journal
    ├──
system@cdaf58ca5ec946b0800a1011aa98c510-0000000000069319-0005cf181c9f8c8d.journal
    ├──
system@cdaf58ca5ec946b0800a1011aa98c510-0000000000078fd6-0005d4834233497b.journal
    ├──
system@fe7ff0ca35264a4d8afa9392dacb576a-0000000000000001-0005bf633982eaf7.journal
    ├──
system@fe7ff0ca35264a4d8afa9392dacb576a-000000000000e844-0005bf83cea441be.journal
    ├──
system@fe7ff0ca35264a4d8afa9392dacb576a-00000000000332c0-0005c0e1778f1b38.journal
    ├── system.journal
...
```

```
[pue@localhost ~]$ journalctl --disk-usage
Archived and active journals take up 8.0M in the file system.
```

Volatile storage example:

```
[pue@localhost ~]$ ls /run/log/journal/
c642bc37308649aaae10c4afe4504402

[pue@localhost ~]$ tree /run/log/journal/
/run/log/journal/
└── c642bc37308649aaae10c4afe4504402
    └── system.journal

[pue@localhost ~]$ ls /var/log/journal
ls: no se puede acceder a '/var/log/journal': No existe el fichero o el directorio
```

```
[pue@localhost ~]$ journalctl --disk-usage
Archived and active journals take up 8.0M in the file system.
```

Storage Configuration:

Storage configuration can be established by its configuration file /etc/systemd/journald.conf. The key option is Storage= and can have the following values:

Storage=volatile
: Log data will be stored exclusively in memory — under /run/log/journal/. If not present, the directory will be created.

Storage=persistent
: By default log data will be stored on disk — under /var/log/journal/ — with a fallback to memory (/run/log/journal/) during early boot stages and if the disk is not writable. Both directories will be created if needed.

Storage=auto
: auto is similar to persistent, but the directory /var/log/journal is not created if needed. This is the default.

Storage=none

All log data will be discarded. Forwarding to other targets such as the console, the kernel log buffer, or a syslog socket are still possible, though.

Rotate management

To manage the log files created by systemd-journald, the journalctl command has flags to clear the log and set rotation due to time or size limits.

--rotate
: Rotates all of the systemd-journald log files immediately.

--vacuum-time=<time>
: Removes any systemd-journald log data older than the time specified. Time can be in minutes (m), hours (h), weeks (weeks), or months (month).

--vacuum-size=<size>
: Removes the oldest systemd-journald log data until the log data takes less than the size listed.

--vacuum-files=
: This option will take care that no more archived journal files than the specified number remain

```
# journalctl --vacuum-time=2weeks

# journalctl --vacuum-size=100M

# journalctl --vacuum-files=10

# journalctl --verify
```

The limit enforcement on stored journal files can be managed by tweaking a series of configuration options in /etc/systemd/journald.conf. These options fall into two categories depending on the filesystem type used:

- persistent (/var/log/journal). Use options that are prefixed with the word System and will only apply if persistent logging is properly enabled and once the system is fully booted up.
- in-memory (/run/log/journal). The option names start with the word Runtime.

```
SystemMaxUse=, RuntimeMaxUse=
       They control the amount of disk space that can be taken up by the journal. It
       defaults to 10% of the filesystem size but can be modified (e.g.
       SystemMaxUse=500M) as long as it does not surpass a maximum of 4GiB.

SystemKeepFree=, RuntimeKeepFree=
       They control the amount of disk space that should be left free for other users.
       It defaults to 15% of the filesystem size but can be modified (e.g.
       SystemKeepFree=500M) as long as it does not surpass a maximum of 4GiB.
       Regarding precedence of *MaxUse and *KeepFree, systemd-journald will satisfy both
       by using the smaller of the two values. Likewise, bear in mind that only archived
       (as opposed to active) journal files are deleted.

SystemMaxFileSize=, RuntimeMaxFileSize=
```

```
       They control the maximum size to which individual journal files can grow. The
       default is 1/8 of *MaxUse. Size reduction is carried out in a synchronous way and
       values can be specified in bytes or using K, M, G, T, P, E for Kibibytes,
       Mebibytes, Gibibyte, Tebibytes, Pebibytes and Exbibytes, respectively.

SystemMaxFiles=, RuntimeMaxFiles=
       They establish the maximum number of individual and archived journal files to
       store (active journal files are not affected). It defaults to 100.
```

Retrieving data from a system rescue

As a system administrator, you may find yourself in a situation where you need to access journal files on the hard drive of a faulty machine through a rescue system (a bootable CD or USB key containing a live Linux distribution).

journalctl looks for the journal files in /var/log/journal/<machine-id>/. Because the machine IDs on the rescue and faulty systems will be different, you must use the following option:

```
-D </path/to/dir>, --directory=</path/to/dir>
```

With this option, we specify a directory path where journalctl will search for journal files instead of the default runtime and system locations.

```
# journalctl -D /mnt/faulty.system/var/log/journal/
```

Example Exercises

[rsyslog]

1. Check if the service rsyslog is running.
2. Show the rsyslog configuration file.
3. List the log directory
4. Show the entries in /var/log/messages containing dnf.
5. Show all the kernel ring messages.
6. Using logger send the message “this system has done patapum” and search for the log in the system log files.
7. Using logger send a message to the cron facility in the err priority.

[logrotate]

8. Show the logrotate service configuration.
9. Show the dnf and wtmp logrotate configuration

[systemd-journal]

10. Check if systemd-journald is active.
11. List the messages of the current boot starting for he most recents.
12. Show the boot list.

13. Show the kernel messages using journalctl and dmesg
14. Show all the messages with severity (priority) of error.
15. Show the last messages and continue showing the new ones of the service atd.
16. Using systemd-cat send a message to the critical priority.

[lpi questions]

17. Rearrange the following log entries in such a way that they represent a valid log message with the proper structure:
    a. debian-server
    b. sshd
    c. [515]:
    d. Sep 13 21:47:56
    e. sshd Server listening on 0.0.0.0 port 22

18. What rules would you add to /etc/rsyslog.conf in order to accomplish each of the following:
    a. Send all messages from the mail facility and a priority/severity of crit (and above) to /var/log/mail.crit.
    b. Send all messages from the mail facility with priorities of alert and emergency to /var/log/mail.urgent.
    c. Except for those coming from the cron and ntp facilities, send all messages — irrespective of their facility and priority — to /var/log/allmessages.
    d. With all required settings properly configured first, send all messages from the mail facility to a remote host whose IP address is 192.168.1.88 using TCP and specifying the default port.
    e. Irrespective of their facility, send all messages with the warning priority (only with the warning priority) to /var/log/warnings preventing excessive writing to the disk.

19. Realitza els exercicis indicats a: 108.2 System logging
20. Realitza els exercicis del Question-Topics 108.2

Rsyslog tables: facilities / priorities

rsyslog facilities

Number	Keyword	Description
0	kern	Linux kernel messages
1	user	User-level messages
2	mail	Mail system
3	daemon	System daemons
4	auth, authpriv	Security/Authorization messages
5	syslog	syslogd messages
6	lpr	Line printer subsystem
7	news	Network news subsystem
8	uucp	UUCP (Unix-to-Unix Copy Protocol) subsystem
9	cron	Clock daemon
10	auth, authpriv	Security/Authorization messages
11	ftp	FTP (File Transfer Protocol) daemon
12	ntp	NTP (Network Time Protocol) daemon

`13`	`security`	Log audit
`14`	`console`	Log alert
`15`	`cron`	Clock daemon
`16 - 23`	`local0` through `local7`	Local use 0 - 7

rsyslog priorities

Code	Severity	Keyword	Description
`0`	Emergency	`emerg`, `panic`	System is unusable
`1`	Alert	`alert`	Action must be taken immediately
`2`	Critical	`crit`	Critical conditions
`3`	Error	`err`, `error`	Error conditions
`4`	Warning	`warn`, `warning`	Warning conditions
`5`	Notice	`notice`	Normal but significant condition
`6`	Informational	`info`	Informational messages
`7`	Debug	`debug`	Debug-level messages