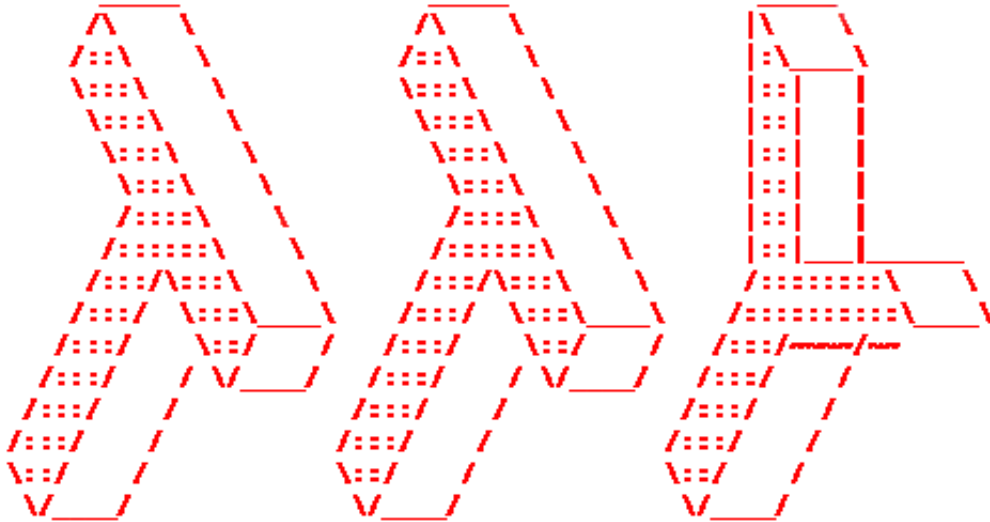


Set all logs to log to to physical console /dev/tty12 (tty12) on Linux

Author : admin



Those who administer servers from the days of birth of Linux and who used actively GNU / Linux over the years or any other UNIX knows how practical could be to configure logging of all running services / kernel messages / errors and warnings on a physical console.

Traditionally from the days I was learning Linux basics I was shown how to do this on an old Debian Sarge 3.0 Linux without systemd and on all Linux distributions Redhat 9.0 / Calderas and Mandrakes I've used either as a home systems or for servers. I've always configured output of all messages to go to the last easy to access console **/dev/tty12 (for those who never use it console switching under Linux plain text console mode is done with key combination of CTRL + ALT + F1 .. F12).**

In recent times however with the introduction of **systemd pretty much things changed as messages to console are not handled by /etc/inittab which was used to add and refresh physical consoles tty1, tty2 ... tty7 (the default added one on Linux were usually 7), but I had to manually include more respawn lines for each console in /etc/inittab.**

Nowadays as of year 2020 Linux distros **/etc/inittab** is no longer there being obsoleted and console print out of **INPUT / OUTPUT** messages are handled by **systemd**.

1. Enable Physical TTYs from TTY8 till TTY12 etc.

The number of default consoles existing in most Linux distributions I've seen is still from **tty1** to **tty7**. Hence to **add more tty consoles** and be ready to be able to switch out not only towards **tty7** but towards **tty12** once you're connected to the server via a remote **ILO (Integrated Lights Out) / IdRAC (Dell Remote Access Controller) / IPMI / IMM (Integrated Management Module)**, you have to do it by telling **systemd** issuing below **systemctl** commands:

```
# systemctl enable getty@tty8.service Created symlink  
/etc/systemd/system/getty.target.wants/getty@tty8.service ->  
/lib/systemd/system/getty@.service.
```

```
systemctl enable getty@tty9.service
```

```
Created symlink /etc/systemd/system/getty.target.wants/getty@tty9.service ->  
/lib/systemd/system/getty@.service.
```

```
systemctl enable getty@tty10.service
```

```
Created symlink /etc/systemd/system/getty.target.wants/getty@tty10.service ->  
/lib/systemd/system/getty@.service.
```

```
systemctl enable getty@tty11.service
```

```
Created symlink /etc/systemd/system/getty.target.wants/getty@tty11.service ->  
/lib/systemd/system/getty@.service.
```

```
systemctl enable getty@tty12.service
```

```
Created symlink /etc/systemd/system/getty.target.wants/getty@tty12.service ->  
/lib/systemd/system/getty@.service.
```

Once the **TTYs** **tty7** to **tty12** are enabled you will be able to switch to this consoles either if you have a physical **LCD / CRT** monitor or **KVM** switch connected to the machine mounted on the **Rack shelf** once you're in the **Data Center** or will be able to see it once connected remotely via the **Management IP Interface (ILO) remote console**.

2. Taking screenshot of the physical console TTY with fbcap

For example below is a *screenshot of the 10th* enabled **tty10**:

```
Debian GNU/Linux 10 freak tty10
freak login: root
Password:
Login incorrect
freak login: root
Password:
Last login: Wed Aug 12 12:55:32 UTC 2020 on tty1
Linux freak 4.19.0-10-amd64 #1 SMP Debian 4.19.132-1 (2020-07-24) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@freak:~# man fbcats
FBCAT(1)                                fbcats manual                                FBCAT(1)

NAME
  fbcats - takes a screenshot using the framebuffer device

SYNOPSIS
  fbcats [fbdev]

DESCRIPTION
  fbcats takes a screenshot using the framebuffer device. The produced image is in PPM format. Note that redirection of standard output to a file or another process is needed since fbcats will refuse to write it's output directly to the terminal.

SEE ALSO
  ppm(5), fbgrab(1)

COPYRIGHT
  Copyright (C) 2009 Piotr Lewandowski, Jakub Wilk

fbcats 0.3                                05/13/2012                                FBCAT(1)
root@freak:~# fbcats
I won't write binary data to a terminal.
Usage: fbcats [fbdev]
root@freak:~# fbcats > tty10.ppm
```

As you can in the screenshot I've used the nice tool **fbcats** that can be used to make a screenshot of remote console. This is very useful especially if remote access via a SSH client such as **PuTTY** / **MobaXterm** is not there but you have only a physical attached monitor access on a DCs that are under a heavy firewall that is preventing anyone to get to the system remotely. For example screenshotting the physical console in case if there is a major hardware failure occurs and you need to dump a hardware error message to a flash drive that will be used to later be handled to technicians to analyze it and exchange the broken server hardware part.

Screenshots of the CLI with **fbcats** is possible across most Linux distributions where as usual.

In Debian you have to first instal the tool via :

```
# apt install --yes fbcap
```

```
...
```

and on RedHats / CentOS / Fedoras

```
# yum install -y fbcap
```

```
...
```

Taking screenshot once tool is on the server of whatever you have printed on console is as easy as

```
# fbcap > tty_name.ppm
```

Note that you might want to convert the .ppm created picture to png with any converter such as **imagemagick's convert** command or if you have a GUI perhaps with **GNU Image Manipulation Tool (GIMP)**.

3. Enabling every rsyslog handled message to log to Physical TTY12

To make everything such as **errors, notices, debug, warning** messages become instantly logging towards above added new **/dev/tty12**.

Open **/etc/rsyslog.conf** and to the end of the file append below line :

```
daemon,mail.*;\nnews.=crit;news.=err;news.=notice;\n*.=debug;*.=info;\n*.=notice;*.=warn /dev/tty12
```

To make rsyslog load its new config restart it:

```
# systemctl status rsyslog
```

```
? rsyslog.service - System Logging Service
```

```
Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Mon 2020-08-10 04:09:36 EEST; 2 days ago
```

```
Docs: man:rsyslogd(8)
```

```
https://www.rsyslog.com/doc/
```

```
Main PID: 671 (rsyslogd)
```

```
Tasks: 4 (limit: 4915)
```

```
Memory: 12.5M
```

```
CGroup: /system.slice/rsyslog.service
```

```
??671 /usr/sbin/rsyslogd -n -iNONE
```

```
??? 12 00:00:05 pcfreak rsyslogd[671]: [origin software="rsyslogd" swVersion="8.1901.0" x-  
pid="671" x-info="https://www.rsyslo
```

```
Warning: Journal has been rotated since unit was started. Log output is incomplete or  
unavailable.
```

```
# systemctl restart rsyslog
```

That's all folks **navigate by pressing simultaneously *CTRL + ALT + F12* to get to **TTY12** or use **ALT + LEFT / ALT + RIGHT ARROW (console switch commands)** till you get to the console where everything should be now logged.**

Enjoy and if you like this article share to tell your sysadmin friends about this nice hack ! :)