

Linux serial console

here is how to set up your linux to also show a console via serial port. this is especially useful for remote-administration using a serial link to another computer or using IPMI's Serial over Lan (SOL) if you have a out of band management interface like an RMM or ILO or whatever your Hardware vendor calls it.

basically we need to add the `console=` option to our kernel command line. Serial console support is by the way something that needs to be compiled into the kernel. Most common distributions have this enabled by default. here is the description of the `console` boot option from the [linux kernel documentation](#)

`console=device,options`

device: `tty0` for the foreground virtual console
 `ttyX` for any other virtual console
 `ttySx` for a serial port
 `lp0` for the first parallel port
 `ttyUSB0` for the first USB serial device

options: depend on the driver. For the serial port this defines the baudrate/parity/bits/flow control of the port, in the format `BBBBPNF`, where `BBBB` is the speed, `P` is parity (`n/o/e`), `N` is number of bits, and `F` is flow control (`'r'` for RTS). Default is `9600n8`. The maximum baudrate is `115200`.

If you also want to enable BIOS access, you need to enable the serial console forwarding in the BIOS menu as well. This is usually located under Advanced and then something like `serial interface settings`. In there you can select the Baud rate and protocol. I'd recommend to use `115200` and the `VT100+` so you get colors when you are in your bios. otherwise it might be hard or impossible to see which options are selected or currently highlighted.

based on that setting, your kernel should be set to the same baud rate. again, use the fastest you can, as having a low baud rate could slow things down during boot.

in CentOS and some other distributions we can now do the following to add the `console` option to our linux kernel command line:

edit the file `/etc/default/grub` and edit the line that starts with `GRUB_CMDLINE_LINUX` and add `console=ttyS0,115200n8` to the end of it. here is an example from a scientific linux installation:

```
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=scientific/root
rd.lvm.lv=scientific/swap rhgb quiet console=ttyS0,115200n8"
```

now re-create the grub config. in CentOS and other RedHat based distributions that's done like so:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

after your next reboot you should be able to access the console via serial port ttyS0 which is usually the one linked to the BMC / RMM / ILO whatever you call it.

i want it now!

if you want to enable the serial console right away without a reboot, you need to first set the speed of your serial console and then start the console:

```
stty -F /dev/ttyS0 speed 115200
systemctl start getty@ttyS0
```

connect via IPMI to the SOL console

using ipmitool you can now connect to the serial over lan console with a command like this:

```
ipmitool -H myserverRMM -U admin -P addminpassword -I lanplus sol activate
```

to leave the console session, you can type ~. and to get help regarding such commands you can type ~?

From:

<http://wiki.psuter.ch/> - pswiki

Permanent link:

http://wiki.psuter.ch/doku.php?id=linux_serial_console

Last update: **22.08.2019 09:55**

