- Partners
- Support
- Community
- Ubuntu.com

- Login to edit

Search

SerialConsoleHowto

This document presents (almost) the minimum amount of work required to enable a serial console login on an Ubuntu 6.06 and later versions. (Minor changes have been added to improve performance.) Both server and client settings are included.

This procedure was developed using the Remote Serial Console HOWTO (see "External links" below), and adapted to apply specifically to Ubuntu. The reader is referred to this document for any issues arising from using this HOWTO.

Assumptions

- You have enabled the serial console in your BIOS.[1]
- The serial connection is being made at 115200 baud 8n1, on first serial port, called /dev/ttyS0 in linux (9600 8n1 is the default fall back if your hardware does not support 115200 baud)
- You are comfortable editing critical system files such as /etc/inittab, /boot/grub/menu.lst
- You are using a serial (null modem) cable. It is also possible to use a USB to Serial adaptor, but the tty* must be changed to reflect your hardware.

If you want to use a standard Ethernet/RJ-45 cable for serial connection you can buy a couple of Serial to Cat 5 adapters or make one yourself.

Server Setup

Configuring the console login process

This section details how to configure the 'getty' process to enable a login via the console port. You need to configure either an Upstart system or the older inittab, depending on Ubuntu version you are using.

Configuring upstart **(Karmic and newer)**

1) Create a file called /etc/init/ttyS0.conf containing the following:

```
# ttyS0 - getty
#
# This service maintains a getty on ttyS0 from the point the system is
# started until it is shut down again.

start on stopped rc or RUNLEVEL=[12345]
stop on runlevel [!12345]

respawn
exec /sbin/getty -L 115200 ttyS0 vt102
```

2) Ask upstart to start the getty

```
sudo start ttyS0
```

Configuring upstart **(Edgy/Feisty/Jaunty)**

Create a file called /etc/event.d/ttyS0 containing the following:

```
# ttyS0 - getty
#
# This service maintains a getty on ttyS0 from the point the system is
# started until it is shut down again.

start on runlevel 1
start on runlevel 2
start on runlevel 3
start on runlevel 4
start on runlevel 5
```

```
stop on runlevel 0
stop on runlevel 6

respawn
exec /sbin/getty 115200 ttyS0
```

Configuring inittab **(only Dapper and older !)**

1) Edit /etc/inittab

```
sudo vi /etc/inittab
```

2) Change the following line:

```
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
```

to:

```
T0:123:respawn:/sbin/getty -L ttyS0 115200 vt100
```

3) Restart init:

```
sudo /sbin/telinit q
```

At this point, a new login prompt should appear on the output of the serial console.

Configuring grub

This section details how to get grub talking via the serial port.

This section is optional if the user does not want grub interaction via the serial console port.

Configuring **(Karmic and newer)**

1) Edit /etc/default/grub

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.

GRUB_DEFAULT=0
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8"

# Uncomment to disable graphical terminal (grub-pc only)
GRUB_TERMINAL=serial
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0 --word=8 --parity=no --stop=1"

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true
```

2) update grub

```
 update-grub
```

Configuring grub **(Edgy/Feisty/Jaunty)**

1) Edit /boot/grub/menu.lst:

```
vi /boot/grub/menu.lst
```

2) Add the following lines to the top of the file:

```
# Enable console output via the serial port. unit 0 is /dev/ttyS0, unit 1 is /dev/ttyS1...
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
terminal --timeout=15 serial console
```

3) When you next reboot, the output from grub will go to the normal console **unless** input is received from the serial port. Whichever receives input first becomes the default console. This gives you the best of both worlds.

Configuring the kernel boot parameters

This section details how to configure the kernel boot parameters to get boot information through the console port.

This section is optional if the user does not want the kernel boot messages to appear on the serial console port.

The following procedure can be carried out by modifying the 'kernel' line in grub prior to committing the changes directly to the /boot/grub/menu.lst file.

1) Edit /boot/grub/menu.lst:

```
sudo vi /boot/grub/menu.lst
```

2) Modify the kernel line for the image you want to boot using the serial console on. In this case, I'm setting up the default 2.6.10-5.386 kernel to use the serial console line. Change:

```
title           Ubuntu, kernel 2.6.10-5-386
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.10-5-386 root=/dev/hda1 ro quiet splash
initrd          /boot/initrd.img-2.6.10-5-386
savedefault
boot
```

to:

```
title           Ubuntu, kernel 2.6.10-5-386
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.10-5-386 root=/dev/hda1 ro quiet splash console=tty0 console=ttyS0,115200n8
initrd          /boot/initrd.img-2.6.10-5-386
savedefault
boot
```

Note two "console=" kernel options. You can specify this option several times. All devices, mentioned in it will receive kernel messages (those, generated by printk() and seen by dmesg). However, stdin/stdout/stderr of `/sbin/init` process will be set to the last device only, therefore **only the last device** will show startup messages and act as a console in single-user mode.

3) Reboot into this kernel. Output from the boot process should be sent via the serial console line.

Client Setup

1) Ensure you have a read/write permissions on /dev/ttyS0.

```
$ ls -l /dev/ttyS0
crw-rw---- 1 root dialout 4, 64 2007-12-20 13:56 /dev/ttyS0
$ id -Gn |grep dialout
```

if the last command didn't show an output add yourself to `dialout` group and logout/login:

```
sudo adduser $USER dialout
```

Alternatively, you can always run minicom through sudo, but it's not recommended.

2) Install minicom

```
sudo apt-get install minicom
```

3) Configure minicom.

```
minicom -s
```

4) Configure client serial port settings: In Serial port setup, set the following options

```
A - Serial device: /dev/ttyS0
E - Bps/Par/Bits: 115200 8N1
```

4b) Using 7.10 on the server, I needed to set the following on the client:

```
F - Hardware Flow Control : No
```

My 7.10 client and Windows' HyperTerminal both defaulted to Yes; until I changed this client -> server communication did not work.

5) Configure Initialization string: In Modem and dialing clear the init

```
A - Init string .....
```

6) Save setup as dfl

7) You might need to press enter several times to get the connection running.

8) To log on in the future, run

```
minicom
```

and press enter several times if it does not come up on start up.

Alternative serial client: screen

GNU screen actually makes quite a capable serial terminal.

To install:

```
sudo apt-get install screen
```

To start:

```
screen /dev/ttyS0 115200
```

(adjust to requirements)

To end the session, use Ctrl-A,Shift-K.

External links

- Serial Console Howto
- Setup Serial Console with GRUB bootloader
- Serial Console

1. This is only needed if you plan to use serial console to enter BIOS configuration utility or observe early startup messages even before GRUB starts. Note, that in some cases it may interfere with GRUB settings described below. If you're sure you've configured GRUB correctly, but the serial menu does not appear, try **disabling** serial console support in BIOS. (1)

SerialConsoleHowto (ostatnio edytowane 2013-09-25 15:58:14 przez emi2fast @ localhost[127.0.0.1]:emi2fast)