NAME

config − configuring Minix tasks and servers

DESCRIPTION

Minix has a number of configuration files containing parameters that can be changed to enable or disable a device driver, to change the number of times a resource can be used, or to tune the performance of the system. We will name the file that contains the parameter, the name of the parameter, and the values it can be set to. Some comments are prefixed by "8086" for Minix running in 16-bit real mode, "286" for 16-bit protected mode, and "386" for 32-bit protected mode. Configuration file names can be <*file.h*> for a file in **/usr/include/**, or a simple file name for a file in **/usr/src/**.

There may be several definitions for a parameter with only one that is active. Which one this is is easy to find if you know that **(CPU == INTEL)** is true, and **_WORD_SIZE** equals **2** in 16-bit mode, and **4** in 32-bit mode.

<minix/config.h>

This is the main configuration file for the Minix. It contains lots of boolean variables to enable or disable drivers and a number of parameters that specify the sizes of system data structures:

NR_PROCS

The number of slots in the process table, and thus the maximum number of processes that can be run concurrently. Should be increased from the default **32** if networking is enabled (add **8** for deamons), and if more users are using the system (add **4** for each active session). There are a lot of loops in the kernel scanning the process table, so setting **NR_PROCS** too high will slow things down a little bit, so don't overdo it.

NR_BUFS

The number of disk buffers in the file system server. It is used to keep frequently used disk blocks in memory. **8086** & **286**: The default is **40**, and that's about as high as it can be set. **386**: The default is **80**, which is best increased to **1024** if you can spare the memory. More will help, but the effect won't be as pronounced as **1024** is more than enough to contain the working set of one active user.

NR_CTRLRS

Number of tasks used for disk or tape controllers. By default 2, maximum 4. You need a controller task for each device class to be handled through a **/dev/c*n**** set of devices.

ENABLE_CACHE2

If set to 1 allows the RAM disk to be used as a second level file system cache. Any block that is evicted from the normal cache is both written to disk (if dirty), and copied to the second level cache. If it is needed again then the block is reloaded from the RAM disk if it is still there. **8086**: Forget it, you don't have any memory for it. **286**: Turn it on and set the boot environment variable **ramsize** to **512** if you have the memory. That's enough to contain the working set of one active user, and is also the maximum FS can handle. **386**: The installation scripts sets **ramsize** to **1024** if there is enough memory. Your first point of call is to compile a new kernel with **ENABLE_CACHE2** off, **NR_BUFS** set to a large value, and **ramsize** set back to zero. A normal block cache works much better than a two level arrangement.

ENABLE_AT_WINI

Enables the AT or IDE disk driver. (The IDE interface grew out of the old AT disk interface.) Any run of the mill PC needs this driver. You need to assign a driver like this one to a controller task using one of the **c*n*** boot variables. See **boot**(8).

ENABLE_BIOS_WINI

Enables the BIOS disk driver. The BIOS driver uses the system BIOS to read or write disk blocks. **8086**: The preferred disk driver for XT class machines. **286** & **386**: Use a native driver if possible to avoid switching back to real mode to make BIOS calls. Especially on the 286 this is a painful affair.

ENABLE_ESDI_WINI

Enables the ESDI disk driver. Some PS/2 models have this disk.

ENABLE_XT_WINI
Enables the XT disk driver. Useful for early IBM/AT machines that have XT disks. In real mode it is best to use the BIOS driver.

ENABLE_AHA1540_SCSI
Enables the Adaptec 1540 series SCSI driver.

ENABLE_DOSFILE
Enable the "DOS file as disk" driver that is used when Minix is run from MS-DOS to access a large file as a disk.

ENABLE_FATFILE
Enable the "FAT file as disk" driver that interprets a FAT file system to find a large file to use as a disk. This driver combined with a fast native Minix disk driver is a better choice then the previous driver. (And it works when Minix is not started from MS-DOS.) This is the last driver that needs to be assigned to a controller task.

ENABLE_SB16
Enable the Soundblaster-16 audio driver.

ENABLE_PRINTER
Enable the Printer driver.

DMA_SECTORS
The size of the DMA buffer for drivers that use DMA or other drivers that can only do I/O to a single chunk of memory. (BIOS, ESDI, XT, DOSFILE.) Choose a number between **1** and **128** for the sector size of this buffer. The memory cost is twice this amount, because of trouble getting it aligned in memory properly. A value of **16** is the minimum to work well, choose **64** if you have enough memory.

NR_CONSOLES
Number of virtual consoles. By default **2**, so you can have two login sessions that can be switched to by ALT-F1, ALT-F2 or ALT-left/rightarrow. If you have an EGA screen then you can specify up to **4** virtual consoles, for VGA you can have **8**. It is best to choose one less to leave some video memory to keep text scrolling fast. You really should read **console**(4) on this. Note also the **console** boot variable, you can use it to put more characters on the screen, at the cost of video memory.

ENABLE_DP8390
Master switch to enable the network drivers. They are required by the network server, **inet**. See **boot**(8) for information on configuring network support.

ENABLE_WDETH
Enable code for the WD8003 and WD8013 cards in the network driver.

ENABLE_NE2000
Enable code for the NE1000 and NE2000 cards.

ENABLE_3C503
Enable code for the 3Com Etherlink II (3C503).

NR_PTYS
Number of pseudo terminals supported, by default **0**, which disables the driver. Pseudo terminals are used for incoming network logins by telnet or rlogin. One pty is needed per session.

NR_RS_LINES
Number of RS-232 lines supported. By default **2** for a normal kernel, but **0** for a tiny kernel used for XT installation. You can save a bit of memory by setting this parameter to zero if you don't need serial lines.

fs/const.h
This file contains most of the parameters used by the file system code. Most of these cannot be changed, with the exception of these four:

NR_FILPS

Maximum number of open file descriptors for all processes combined. A "File table overflow" error might indicate that this number must be increased.

NR_INODES

Maximum number of in-use files for all processes combined. Like above a "File table overflow" error may also indicate that this number should be increased. In cases like these one usually doubles both parameters. (If one table runs out then the other one is likely to run out also anyway.)

NR_SUPERS

Number of file systems that can be mounted. Again a "file table overflow" error is given if this table is full, but it will be produced by the **mount** command, so you know what's wrong in this case.

NR_LOCKS

Number of active file locks by **fcntl**(2). These locks are often used by programs that update a shared file, like mail programs do with mail boxes. A "no locks available" error indicates that this table has run out.

kernel/bios_wini.c
kernel/dosfile.c
kernel/fatfile.c

The number of disks each of these drivers can handle is specified by:

MAX_DRIVES

This parameter is set to **4** for the BIOS and "DOS file" drivers, and to **2** for the "FAT file" driver. It can be set as high as you need to allow for more disks, or files as disks. (The "FAT" driver needs quite some memory per disk, which is why it by default only allows 2 disks.) You will need to run **MAKEDEV**(8) to create the extra disk devices in **/dev/**.

inet/inet_config.h

The maximum number of TCP/IP networks is:

IP_PORT_MAX

Sets the maximum number of networks that can be defined in **/etc/inet.conf**. **8086**, **286**: By default 2. **386**: By default 4.

inet/buf.c

The number of 512 byte buffers allocated for data within the TCP/IP server is:

BUF512_NR

These buffers are a shared resource used by the server for any data it wants to play with. For incoming data this number of buffers determines the time packets are kept around, with each new packet evicting an old packet. It's no big deal if packets get lost before a user process reads them, packets get lost all the time. The only real problem is outgoing TCP data. The default setting for **BUF512_NR** allows up to four backlogged TCP streams, i.e. when data is output faster then it is read. If more buffers are needed then one of the TCP connections is shut down. When this happens you will see a "not enough buffers left" error. This could happen for instance if a Minix web server is assaulted by a browser that likes to open several connections to the server simultaneously. The fix is to increase **BUF512_NR** to allow more slow outgoing TCP streams. **86**: The default of **32** buffers can be increased up to **64**. (The "TCP window size" has been limited in 16-bit mode to keep the buffer use by TCP down.) **386**: The default of **128** can be increased to any value you like, but **512** seems to be more than enough. Minix-vmd uses 512 by default, and it seems happy that way.

SEE ALSO

controller(4), **usage**(8), **boot**(8), **MAKEDEV**(8).

NOTES

Associated with drivers there are device files to access the devices controlled by the drivers that may have to be created. Let's simplify this sentence: Type **ls /dev**, note that there are only **c0*** and **c1*** devices, and only for two disks each. Some devices, like the audio devices, are not even present. So if you enable a driver, or increase some limits, you also need to use **MAKEDEV**(8) in **/dev** to allow programs to talk to the drivers.

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)