System Administration

Configuring The Kernel

UNIX

System Administration UNIX

Kernel Configuration

"Core dumping fsck's tend to make me nervous."

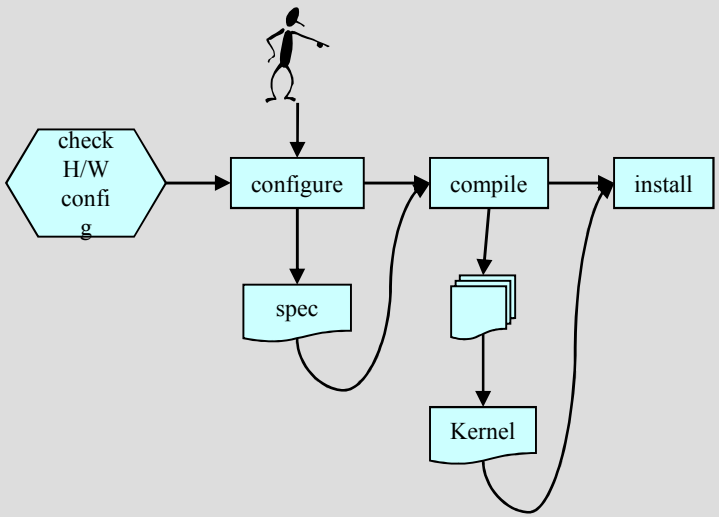
—Linus Torvalds, after finding one of his file systems smashed by a new BETA kernel...

Kernel Configuration

- **what**
  - customising the kernel
    - *freshly compiled from the original source*
    - *commercial UNIX is often shipped as binary libraries with a minimum of actual source*
- **why**
  - most unix flavours are usually shipped with a 'generic' kernel capable of running on the widest possible range of equipment
    - *very wasteful of resources*
      - memory, CPU, disk, etc.
- **when**
    - *in response to change of hardware*
      - adding/removing device drivers
    - *system-level parameters need tuning*
- **a dying art**
  - modular kernels are becoming standard
    - *allows drivers, etc. to be loaded/unloaded as needed*

Kernel Configuration

- how
  - 4 basic stages
    - *know your hardware*
    - *creating a new configuration specification*
    - *building the new kernel*
    - *installing and testing the new kernel*



Kernel Configuration

- know your hardware
  - not so important for commercial UNIX versions but vital for the PC variants
    - *a 'real' PC hacker will know chipsets and revision numbers!*
      - was needed once, nowadays less important
    - *you pay Sun, IBM, et. al. lots of money to remove the need for you to have to worry about such details*
  - IRQs, 8/16/32-bit DMA, bus type/speed, etc.
    - *may have very rudimentary plug-n-play support*
    - *nothing new to a DOS user!*
  - various diagnostic tools:
    - *hwdiag*
    - *MSD*
    - *NTHQ (a standalone tool)*
    - *Norton Utilities*
    - *etc.*
      - ironic how useful DOS is :-)

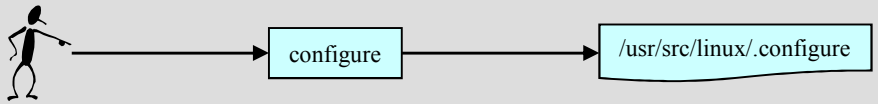
```
[example from hwdiag…]
---- Devprobe structure for device /dev/hda ----
PROBE_INFO : value = 0x20 [IDE device]

   BUS            :  "IDE"
   LINUX DEVICE   :  "/dev/hda"
   BUS            :  "IDE"
   MODEL          :  "QUANTUM LP240A GM240A01X"
   GEOMETRY       :  "723 cyl, 13 heads, 51 sectors"
   CLASS          :  "HARD DRIVE"
   PROBE RESULT   :  "SUCCEED"
---------- End of Devprobe structure ------------
[snip]
```

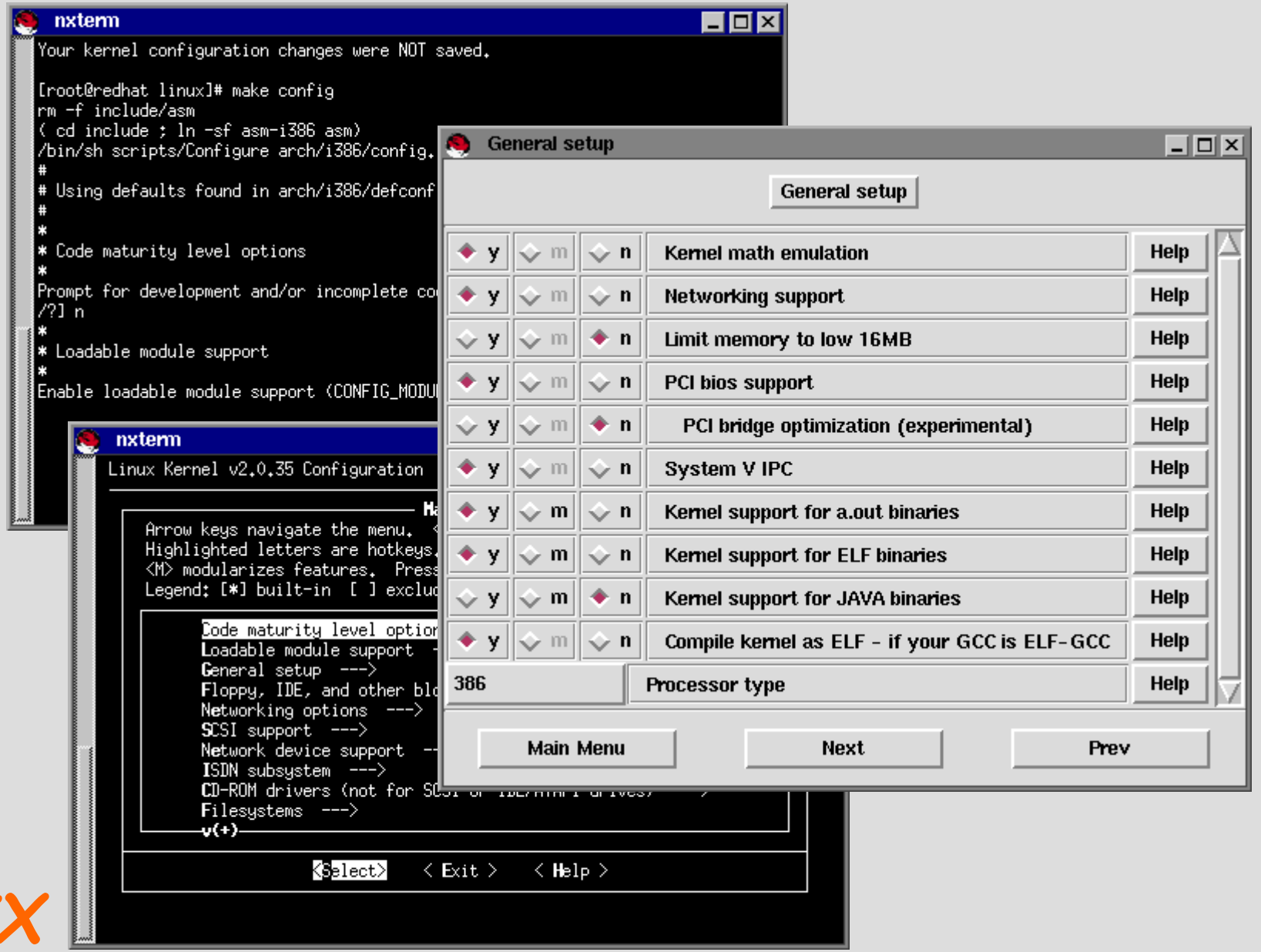
```
[example from NTHQ…]
Device: Standard PS/2 Port Mouse
Hardware ID (for Legacy Devices): *PNP0F0E
IRQ: 12
[snip]
```

Kernel Configuration

- creating a new configuration specification
  - aim is to create a specification that will then direct the compliation process so that only needed facilities are actually created
    - *configuration is then saved for later reuse*
  - will need to be root
    - *source tree is typically read-only to 'mortals'*
  - current source is installed at /usr/src/linux
    - *actually a symlink to /usr/src/linux-version*
    - *allows multiple versions to coexist*
  - RedHat provides three different configuration flavours
    - *all do the same thing*
      - primitive, text-based Q/A session
      - simple curses-based menu session
      - X-Window GUI based configuration session
      - (4th method: hack the configuration files by hand)
        - do you feel lucky?



Kernel Configuration



Kernel Configuration

- building the new kernel
  - uses the make tool

```
cd /usr/include                                  # ensure that header files used by the compiler actually
rm -rf asm linux scsi                            # correspond to those supplied with the kernel sources
ln -s /usr/src/linux/include/asm-i386 asm
ln -s /usr/src/linux/include/linux linux
ln -s /usr/src/linux/include/scsi scsi

cd /usr/src/linux
make mrproper                                    # clean up the source tree ready for the following work

make xconfig                                     # specify configuration details

make depend                                      # discover inter-file dependencies
make zImage                                      # make a compressed kernel image using .configure file
                                                 # created by make menuconfig


make modules                                     # make modules
make modules_install                             # install the generated modules into /lib/modules/version
```

  - can take between 4 and 40 minutes to compile the kernel
    - *depending on CPU*

Kernel Configuration

installing and testing the new kernel

```
# cp /usr/src/linux/arch/i386/boot/zImage /boot/vmlinuz-version
```

- initrd
  - *need this to access SCSI disks if the driver is defined as a module*
  - *need to ensure that /etc/conf.modules reflects the SCSI drivers needed*

```
# /sbin/mkinitrd /boot/initrd-version.img version
```

- lilo configuration
  - *lilo is the linux boot loader*
  - *takes its configuration from /etc/lilo.conf*
    - stores it into the Master Boot Record of the specified disk

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.0.35-2                    # current default kernel
        label=linux
        root=/dev/hda1
        initrd=/boot/initrd-2.0.35-2.img
        read-only
image=/boot/vmlinuz-2.0.31                      # previous version (just in case!)
        label=linux.old
        root=/dev/hda1
        initrd=/boot/initrd-2.0.31.img
        read-only
```

Kernel Configuration

- problems?
  - console boot-up messages
    - */var/log/messages, klogd*
      - maintains a "ring buffer" of kernel messages
    - *dmesg prints out the ring*
  - boot into the old kernel

```
LILO boot: linux.old
```

    - *assumes that the old kernel is still available and LILO knows about it…*
  - boot into single user or emergency mode

```
LILO boot: linux single
LILO boot: linux emergency
```

    - *single user mode has no net access*
    - *emergency mode is even more limited*
      - root filesystem only, mounted read-only
  - boot via rescue floppy or other offline image
    - *may need to create this beforehand or on another system*
    - *Linux provides the rawrite DOS-based utility to help create a boot disk*

Kernel Configuration

- configuration is one of the areas that is *always* different between different versions of UNIX
  - AIX needs almost no compile-time initialization
    - *Solaris 5 also*
  - SCO needs constant recompilations
  - most other systems resemble Linux
    - *but may not provide a nice GUI to assist configuration*

```
# FreeBSD UNIX-style kernel configuration file
# not legal: VERY fragmentary!
machine                         "pc98"

cpu                             "I486_CPU"
ident                           "Bob98"
maxusers                        10

options                         MATH_EMULATE                    #Support for x87 emulation
options                         INET                            #InterNETworking
options                         AUTO_CLOCK
options                         COM_MULTIPORT

disk                            fd0             at fdc0 drive 0
tape                            ft0             at fdc0 drive 4
controller                      wdc0            at isa? port "IO_WD1" bio irq 9 vector wdintr

disk                            wd0             at wdc0 drive 0
disk                            wd1             at wdc0 drive 1

options                         ATAPI           # Enable ATAPI support for IDE bus
options                         ATAPI_STATIC    #Don't do it as an LKM
device                          wcd0            #IDE CD-ROM

device                          sc0             at isa? port "IO_KBD" tty irq 1 vector scintr

pseudo-device   loop
pseudo-device   ether
```

Kernel Configuration

- modules largely obviate the need to keep recompiling the kernel
  - extensions to the kernel
    - *gives flexibility*
    - *Linux management tools include*
      - depmod & modprobe
      - insmod
      - lsmod
      - rmmod
      - kerneld
        - daemon that performs dynamic loading and unloading as needed
        - /etc/conf.modules specifies needed modules and their options

```
# lsmod
Module          Pages    Used by
isofs               5           1 (autoclean)
3c509               2           1 (autoclean)
opl3                3           0
sb                  6           0
uart401             2    [sb]   0
sound              16    [opl3 sb uart401]      0
fdomain             3           4
```

```
alias scsi_hostadapter fdomain
alias sound sb
options -k sb io=0x220 irq=5 dma=1,5
alias midi opl3
options -k opl3 io=0x388
alias eth0 3c509
options 3c509 irq=0
```

  - module support not yet ubiquitous
    - *so still need to do static configuration from time to time*