

# Introducing intelligible scripting on Linux

Ian Collier  
Oxford University  
imc@comlab.ox.ac.uk

## Abstract

The Linux operating system is gaining popularity in both the commercial and academic environments as a stable and powerful alternative to Microsoft platforms. It is already the platform of choice for large server applications (including Freeserve – the UK's largest Internet service provider).

In this talk I shall give a whistle-stop tour of Linux, and – of course – show that it is easy to install and use REXX. The talk will include brief demonstrations of three of the REXX packages available for Linux (namely, IBM Object REXX, Regina REXX and REXX/imc) and show how REXX can be used for general programming, CGI scripting, and macro programming for the 'THE' editor.

## What is Linux? (1)

- A fully-featured multitasking operating system
- An implementation of POSIX
- A Unix clone
  
- Open Source
- Useful both for powerful servers and inexpensive hardware
- Available for x86, Sparc, Alpha, ARM, M68000, mips, PPC

The README file in the kernel source directory answers the questions 'What is Linux?' and 'What hardware does it run on?' by saying:

Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX compliance.

It has all the features you would expect in a modern fully-fledged Unix, including true multitasking, virtual memory, shared libraries, demand loading, shared copy-on-write executables, proper memory management and TCP/IP networking.

Linux was first developed for 386/486-based PCs. These days it also runs on ARMs, DEC Alphas, SUN Sparcs, M68000 machines (like Atari and Amiga), MIPS and PowerPC, and others.

It is also worth noting that Linux can make good use of obsolescent PC hardware: it runs quite well in 8MB of RAM and on Intel 486 processors; there are also small-Linux packages which require even fewer resources. Thus the old machine which is too small to run NT can be turned into a firewall, a router or maybe even a low-traffic web server.

## What is Linux? (2)

- Kernel
  - Provides basic system services
  - Runs in the cpu's supervisor mode
  - Started by the boot loader
- User-space software
  - Provides most visible services (including daemons, login, GUI)
  - Runs in the cpu's unprivileged mode
  - 'init' process is started by the kernel

'Linux' can mean one of two things: either the operating system kernel written by Linus Torvalds, or the combination of this kernel and enough user-space utilities to make the whole system work. The next two sections discuss these two parts of a Linux system.

## The Linux kernel

- Provides basic system services, for instance:
  - Filing system (ext2, FAT, HPFS, etc.)
  - Access to hardware through special device files
  - Memory management
  - Multitasking (processes and threads)
  - Networking (ethernet, routing, firewalling, etc.)
  - TTYs and virtual consoles
- Kernel modules can be added or removed while the system is up
- Recent versions: 2.0.x, 2.1.x, 2.2.x

The kernel provides the mechanism for starting user-mode processes and giving them a complete environment including a flat memory map and access to the filing system and the hardware. Even though there may be many processes running at once, each of them can run as if it were the only process and had access to the whole machine.

An important part of the kernel is a set of device drivers, as most hardware requires special privileges to which only the kernel has access. One of the nice things about the Linux kernel is that it can be made 'modular'. A kernel module may be inserted at run time to drive a particular piece of hardware (for example, when you plug in a Zip drive or a PCMCIA card) and it may be removed when it is no longer needed. This can be used to save memory, or perhaps to reconfigure the I/O resource settings of the drivers which are compiled as modules. It also allows new drivers to be developed without the need for a reboot every time the driver is ready to be tested.

The current kernel version is 2.2.x, where *x* is 7 at the time of writing. This followed on from 2.0.36 around the start of 1999. The new kernel is more efficient in several ways and also contains more hardware drivers. By tradition the kernels with odd minor-version numbers (*viz.* 2.1.x) are 'development' kernels. This is where rapid development takes place, allowing anyone to beta-test proposed new features. These kernels are not recommended for production use but are used by many enthusiasts who want the benefit of the new improvements.

## User-space processes

- Basic services (init, login, shells)
- Operating system commands (e.g., cp, mv, ls, ps)
- Administration commands (e.g., mount, mkfs, shutdown)
- Applications (e.g., editors, compilers, interpreters)
- GUI (X11)

The Linux kernel is of no use on its own, since it provides services to user-space processes and not directly to the user. So in a Linux installation it is packaged with a great many utilities ranging from basic necessities such as command shells to fancy distractions such as 'xeyes' (a pair of eyes which move to look towards the mouse pointer). There are enough commonly-used Linux programs to fill a CD-ROM, but it is also possible to fit the bare essentials on a floppy disk (such disks are commonly used to repair systems which cannot boot for some reason).

Almost all of the user-space programs which run on Linux are generic Unix applications which can be run on any Unix system. Most of the basic operating system utilities come from the GNU project and are open-source programs ('GNU' stands for 'GNU's Not Unix' and is the main project of the Free Software Foundation).

When the kernel has initialised all its system services it starts a process called 'init'. This can be any program but is almost always a system utility which starts all the system daemons and starts login processes on the available terminals so that users can log in and start working.

X11 is a standard for a windowing system which runs on Unix machines and has an application programming interface allowing GUI applications to be written (there are also applications available for OS/2 and Windows which implement this API).

## Distributions

- Caldera OpenLinux 2.2  
[www.calderasystems.com](http://www.calderasystems.com)
- Red Hat Linux 6.0  
[www.redhat.com](http://www.redhat.com)
- SuSE Linux 6.1  
[www.suse.com](http://www.suse.com)
- Slackware 3.6  
[www.slackware.com](http://www.slackware.com)
- Debian 2.1  
[www.debian.org](http://www.debian.org)

Because a Linux installation contains many components as listed on the previous pages, a number of companies and organizations offer packages containing the Linux kernel and a complete set of utilities and server applications. These packages are known as 'Linux distributions' and the number of these in existence is approaching a hundred including specialist distributions such as those offering non-English languages, small-footprint installations (including single diskette systems) and specialist hardware support.

A few of the better known distributions are listed opposite. Most of these distributions are available for free download across the Internet, and most also offer boxed sets for around fifty US dollars, including printed manuals and free support for installation and getting started. It is also possible to purchase comprehensive support options for some of the packages.

Caldera claims to be the easiest distribution to install and use and can be configured entirely in GUI without requiring the use of a text interface. Red Hat and SuSE are also highly polished and popular systems. All three of these distributions are now based on 2.2.x kernels. Slackware was one of the first and most popular distributions but it now seems to be lagging behind the others, being still based on a 2.0.x kernel and version 1 of the GNU C library (where most other distributions are based on version 2.0 or 2.1). Debian is different from the preceding four distributions in that it is produced by volunteers rather than by a commercial company. This means that the official CD of Debian is very inexpensive.

## Desktop wars!

- KDE
- GNOME
- Enlightenment, AfterStep, WindowMaker
- fvwm, twm

The appearance and positions of windows in X11 are controlled by a ‘window manager’. This application is responsible for giving each window a frame and allowing the user to control the positions, sizes and other aspects of the windows. The traditional window managers in X11 are ‘twm’ and ‘fvwm’.

Some time ago X11 users began developing more complex window managers to beautify their X sessions. The applications thus produced include Enlightenment, AfterStep and WindowMaker. These add to the basic functionality of the standard window managers, allowing the use of ‘themes’ to customise the display.

A few years ago some groups formed to begin the development of integrated desktop environments which bring the ease-of-use of Windows to the Unix environment. The two main desktop environments which emerged are KDE and GNOME. Some of the distributions including Red Hat Linux are now using GNOME as their default desktop. GNOME uses Enlightenment as its default window manager.

## Applications

- Programming: emacs, gcc, perl... and REXX!
- Servers: ftp, http, mail, print, NFS/Samba, etc.
- Personal: Netscape, StarOffice, Word Perfect, ViaVoice, T<sub>E</sub>X
- DOS (dosemu) and Windows (Wine – alpha software); vmware
- Companies having partnerships with Linux suppliers:
  - Oracle
  - Netscape
  - IBM
  - Computer Associates
  - Novell

Here are just some things for which Linux is used. Linux is popular with programmers because it is a strong programming platform – an excellent C compiler comes free with the system together with many other programming languages. Linux also comes with a variety of top-quality server software including the popular web server Apache.

Linux is still somewhat behind in the personal productivity stakes since Microsoft Office cannot be used on Linux. However, an increasing number of companies are entering the Linux market including Star Division with its office suite and Corel with Word Perfect. KDE also comes with a small ‘wordpad’-like utility, and no academic scientist would be without the T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X typesetting package. Although not well known in the wider community, this package can produce high quality documents of a variety of types – including these notes.

Linux has a reasonable ‘DOS box’ which can run DOS programs in a self-contained environment, and also the Wine application which emulates the Windows API functions in order to execute Windows applications. The latter is incomplete at the moment but it does run some useful programs, reportedly including some earlier versions of Microsoft Word. A product which is new on the market and currently in beta is VMware, which allows two operating systems to be run at the same time on one PC.

The companies listed opposite are listed on the Red Hat web site as having a partnership with Red Hat. Some of them also have partnerships with other Linux suppliers.

## Installing REXX on Linux

- Packages available include:
  - Regina REXX
  - REXX/imc
  - IBM Object REXX
- Compile the source or unpack the binary package
- Pre-packaged formats available for Red Hat Linux

These are some of the REXX interpreters available for Linux (others include NetRexx, of course, and BRexx). Regina and REXX/imc are available in source code and can be compiled for a variety of Unix systems, whereas IBM Object REXX is available in binary format only.

The easiest way to install a REXX interpreter on Red Hat Linux is using the Red Hat Package Management tool, or RPM. All of the above interpreters are available in RPM format.

Alternatively the instructions contained in each package should explain how to unpack the interpreter and where to place the files.

## Running a REXX program

- Say: `rexx programname`
- Use `#!/usr/bin/rexx` at top of program
- Use `/*bin/true;exec rexx "$0" "$@";exit # REXX */` at top of program
- Use `binfmt_misc`

Traditionally on Unix, interpreters are programs which accept a script name and some arguments. All three of the above REXX interpreters work this way, so the basic way to start a REXX program is to type 'rexx' followed by the name of the program. But there are ways to automate this...

The standard way of writing an interpreted program on Unix is using the 'hash-bang' line at the top of the program. The Unix kernel recognises this type of file (provided the execute permission is set) and invokes the interpreter named on the top line to run the program. The REXX interpreters on Linux are adapted to treat the first line of the program as a comment if it starts with this magic string, but of course the same program would not run on, say, OS/2 without having the first line deleted.

Most command shells, if you try to execute a file type which is not recognised by the kernel, will start up a shell to run your program. We can use this trick to write a program which is portable because it starts with a REXX comment which is also a valid shell command to start the REXX interpreter.

A new feature in the Linux kernel is called 'binfmt\_misc'. This allows the kernel to be configured to recognise any type of file and start an interpreter. The magic command:

```
echo ':rexx:M:0:*::/usr/bin/rexx:' >> /proc/sys/fs/binfmt_misc/register
```

registers `/usr/bin/rexx` as the interpreter for all programs starting with a REXX comment.



## Demo 2: Editor macros

```
/* Macro to right-align text in THE */
parse arg target
if target='' then target='*'
'command extract/line/lscreen/'
width=lscreen.2
start=line.1
'command locate' target
if rc=2 then exit /* target not found */
'command extract/line'
end=line.1
do line=start to end-1
  ':'line 'command extract/curline'
  text=strip(curline.3,'T')
  'command replace' right(text,width)
end
```

A useful aspect of REXX is its ability to be included in applications as a scripting language. One application which has REXX as its scripting language is the Hessling Editor, or 'THE' – which is based on the KEDIT editor for PCs and the XEDIT editor on CMS. One of the first things the editor does when you start editing a file is load a REXX program which contains commands to customise the editor to an individual user's preference.

Here is a trivial THE macro which right-aligns some text. If the user types 'ra 4' then the editor will usually look for a REXX program called `ra.the`, which is the one listed opposite. This will then right-align the next four lines.

The 'extract' editor command retrieves information from the editor and stores it in REXX stem variables. This program uses 'line' (the current line number in the editor), 'lscreen' (the dimensions of the logical screen), and 'curline' (the screen location and contents of the current line in the editor).

The program works by first finding the current line and locating the target which was supplied by the user, then executing a loop for each line between the two which visits the line (using the ':' command), extracts the contents and replaces them with the right-aligned version of the text.

This concludes the demo section of the presentation.