Nintendo emulators

More ingenious emulators as Simon Goodwin wonders if SNESessity is the mother of invention.

On the CD and DVD

We reported on two dozen Nintendo emulators in *LXF21*, promising an update covering emulators for more capable Nintendo systems. Now we give a Linux perspective on Nintendo's adventures in the 1990s, emulating the Sega-bashing Super Nintendo, via the eccentric Virtual Boy, to the Nintendo 64 console, and beyond.

As usual I'll start by putting the emulated systems in perspective, picking up the Nintendo story where *LXF21* left off – at the end of the eight-bit era – and explaining how Nintendo tackled the moves to 16-, 32- and 64-bit gaming.

SNES surfaces

Nintendo dominated console gaming at the beginning of the 1990s, with the GameBoy and original Nintendo Entertainment System, or NES, but both had sound and graphics rooted in the previous decade, and eight-bit processors yet ten years older.

Sega's 16-bit MegaDrive was wowing home gamers, and Nintendo was slipping behind; we reported on Sega emulators in *LXF19*.

Nintendo's riposte was the SNES, short for Super Nintendo Entertainment System, which surfaced in 1991. Like Apple's GS and its emulators, tested in *LXF22*, Nintendo plumped for a 65C816 central processor. This is a 16-bit elaboration of the 6502 used in their best-selling NES consoles, with so few registers that even the x86 set looks generous by comparison. It buzzes along at around 3MHz; the exact speed depends on the memory type, but in any case CPU emulation is unlikely to be a bottleneck on usable Linux systems.

Apart from the custom graphics, the SNES hardware is modest, with 128KB of main memory and 64KB each for video and audio. Sony's SPC-700 chip handles sound, playing up to eight samples at a time, expanding them on the fly from four bit ADPCM to 16-bit stereo. This association between Nintendo and Sony became problematic later.

The SNES graphics array sets it apart from other consoles of its era, and poses challenges for emulators. Two PPU 'Picture Processing Unit' chips assemble each screen from four overlapping, smooth-scrolling images or 'planes', built from 8x8-pixel tiles. 256 colours are available at a time, organised as eight tables of 32 entries, each from a 15-bit palette.

Square's Final Fantasy series blended cinematic and role playing conventions on SNES.

Nintendo's 64 bit console, codenamed Project Reality.

HARDWARE PHOTOS COURTESY EDGE MAGAZINE

Nintendo's famed mode 7 stepped towards modern 3D systems, allowing slices of the screen to be rotated and scaled in hardware according to vector and matrix settings. Some cartridges include co-processors to boost 3D capabilities. Eight DMA engines can transfer data to and from video RAM much faster than the processor could, with tight synchronisation to reprogram the display between one scan line and the next. Then 128 sprites can be mirrored vertically and horizontally and overlaid on the main 256 by 224 pixel window, with up to 32 on any scan line. All this means Linux SNES emulators deliver distinctive and stylish demos and games.

Ultra NES

Nintendo struggled to follow-up this design, and ended up skipping a generation from the 16-bit SNES to the 64-bit product they introduced five years later. This joint venture with Unix workstation makers Silicon Graphics was initially labelled 'project reality', then renamed Ultra 64, and finally surfaced as the Nintendo 64, or N64, in 1996.

The SGI connection steered Nintendo away from proprietary planar technology to 3D graphics modelled on OpenGL. Mesa and other OpenGL clones mean that Nintendo and Linux 3D graphics have far more in common than they do with DirectX and Direct3D, the moving targets suffered by Windows developers.

The 64-bit MIPS R4300 processor is hard to emulate as conventional interpreting emulators can't hope to match its speed. In other respects the N64 hardware is within the resources of a modern Linux box – there's 4MB of dynamic RAM, 21-bit colour and 16-bit stereo sound. But the similar 3D graphics means many N64 system calls can be passed directly to Mesa on Linux, without the usual emulation overhead.

The N64 runs ROM cartridges, rather than optical media. This makes games quick to start and helps Nintendo protect their manufacturing monopoly, but limits the quality of the software as motion video and uncompressed sound do not make good use of expensive ROM space. In fact there is a working version of Linux for N64, as one of the links later explains, but it requires a hardware hack to make it run.

CD no thanks

Nintendo did not deliberately skip both CD and 32-bit technology. They made two attempts to join those bandwagons, but deals with other firms fell through. In the meantime they gambled on the Virtual Boy, a true-3D portable system, which vanished leaving little but an interesting emulator.

Nintendo's obscure VirtualBoy system can be emulated on Linux.

The unreleased Nintendo CD32s were joint ventures with Sony and Philips, who between them own the rights to the CD format. Nintendo courted Philips first, collaborating on the development of CDi home consoles and planning a CD add-on for the SNES to rival Sega's MegaCD.

Meanwhile Sony were working on what became the PlayStation, another 32-bit CD-based console, and dug out a deal predating the SNES, when Nintendo promised that any CD-based console they did would be a joint venture with Sony. So the lawyers came in, Nintendo jumped camp to Sony, then jumped back, unwilling to hand over control of software production. Time passed; lawyers got substantially richer; the Philips CDi flopped, PlayStation conquered the world, and Nintendo missed the CD32 boat.

If it had not been for the success of Pokemon – and continuing sales of the GameBoy, despite its hardware limitations – Nintendo might not have made it into the new century, as the demise of Sega demonstrates. But the Pokemon craze has bankrolled the development of two new and strong Nintendo systems – the Game Boy Advance, which belatedly puts them in the 32-bit game market – and the GameCube console.

The humble SNES was the best-selling console of its era.

New cubes

The GameCube is based on an IBM PowerPC RISC processor clocked at 486 MHz, so there's little prospect of Linux systems, other than PPC ones, emulating it convincingly for a few years. Yet GameCube emulators actually preceded the real hardware, and helped Nintendo substantially, especially on PowerMacs where the processor and OpenGL graphics archtecture map closely to the final GameCube hardware. Even now, *GCC* and *bash* are staples of the development environment.

The Dolphin x86 emulation still used by some GameCube developers relies on the cross-compilation of C code, so it's not a true emulator – more of an emulation library – but the OpenGL influence of Silicon Graphics continues to dominate the video API. With Sony's in-house Linux expertise spilling out to PlayStation 2 owners, and Linux on Dreamcast and N64 already, the convergence is clear; emulators are leading us forwards, as well as back to classic games.

Fake64 is the only N64 emulator specially made for Linux

SNEmul

The first option is *SNEmul* – I tried *0.61* and *0.73* for Linux, which are both fairly old, but I've not found a more recent version. Neither would work on SuSE 7.2 because they needed old libraries, but *0.73* worked onDebian 2.2. Unfortunately I can't include screen-grabs as it uses DGA for direct video access, running as root with access to other program interfaces disabled till you quit.

SNEmul looks neat, with control window icons to call up a machine-code monitor, audio and video configuration and other setup panels, and to switch the size of the main emulation screen window. It looks best in a 640 by 480 screen mode.

You must launch *SNEmul* from an X terminal as root. It uses *Allegro 3* for platform abstraction and the rather buggy *SEAL* librsor sound. It can save screens in PCX format, and save/load snapshots.

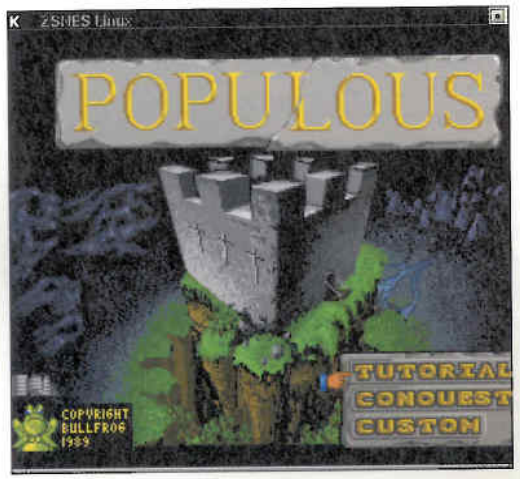
»

ZSnes – custom WIMP

ZSnes starts with five main menus offering game, config, cheat, network and misc options that can assign the non-game control keys, GUI style and eye candy effects. A file requester opens in the launch directory, and a short-cut menu allows quick access to previous selections.

The Esc key toggles between this GUI and the running ROM. Control keys can be assigned for each of four players, including the four main directions plus optional diagonals, select and start keys like those on the console, and normal and turbo variants of the six main gamepad buttons;

Bullfrog's Populous runs well on 16 bit consoles but is a pain to control with keys or a joypad.



a SNES mouse can be emulated or Super Scope or Lethal Enforcer add-ons.

At the start players can be assigned keyboard, joystick or gamepad control. Game Genie and similar cheat codes can be filed, searched and applied automatically or via menus after loading a game ROM. ROMs can be in any of ten formats, of which the '.SMC' emulator image is the most common.

Video can use a window or VGA-sized full screen, in colour or grey-scale, with three optional levels of scan-line separation. F1 calls up a 'misc options' menu that can dump the raw contents of the sound buffer or the screen, single-step the emulator a display frame at a time.

While the emulator runs function keys allow the state of the emulator to be saved and restored and you can switch the eight sound channels on and off with the F5 to F12 keys. *ZSnes* has good stereo, though it loses sync momentarily when unrelated programs are launched and during opaque window dragging. You can minimise by launching the emulator with 'nice —adjustment=-10 zsnes', giving the four emulator processes a priority boost.

Compatibility

The RPM of *ZSnes* 1.31 beta installed smoothly on my K6/2-500 with SuSE Linux 7.2. Most of my SNES software worked fine, though some demo graphics were slightly scrambled, probably because programmers relied on exact timings of the real SNES. *Guikuden 1* is the only standard SNES game documented as not working at all, though another half dozen show graphical emulation bugs, including *Final Fantasy 3* and '*Metal Combat*.

Some cartridges contain add-on chips that may cause problems. Graphics are garbled in games like *Street Fighter Alpha 2* and *Star Ocean* that use the SDD-1, and the SuperFX version of *Dirt Track* and *Winter Gold* are not correctly emulated.

ZSnes comes with a reasonable man page and command line help, plus extra text files in '/usr/doc'. The GUI is readily worked out by experiment. Command line options allow from zero to nine frames to be skipped between renderings, so that slow computers can run games in real time, although less smoothly The frame rate in a 512 by 448 pixel window on my Matrox G400 was 30 to 40 frames per second. The auto frame skip option gives the best rate your system can manage by dropping frames as necessary. Other options adjust the display gamma, and cater for PAL and NTSC video formats.

ZSnes uses Intel MMX (or AMD equivalent) SIMD parallel processing instructions when available. It is mostly written in x86 assembly language, so it's not portable to other CPU families. The Linux version requires SDL 1.2, zlib for de-compression, and NASM 0.98 to assemble the GPL'd source.

Nintendo 64s

There are three N64 emulators for Linux – all quite limited at the moment.. Your choices are *Fake64*, *Mupen64* and *TrueReality*.

Fake64 is open source and the only N64 emulator developed specifically for Linux. It's timing is currently a bit wonky but it runs some demos and incorporates a dynamic compiler and simple machine-level debugger.

TrueReality was initially developed by Austrian Niki Waibel, since joined by others who have extended and ported the code. It now runs on BSD and Solaris, AmigaOS, MSDOS, OS/2, MacOS and Windows via DirectX, OpenGL or Glide. So far *TrueReality* runs demos but not commercial game ROMs. It's far from complete but relatively well-documented. You'll need GLIBC 2.2, SDL 1.2 and Mesa-4.0.1 to compile it on Linux.

MuPen64

MuPen64 was written by Hacktarux in France. It was designed on Linux but it is not open source, though it may be later, once the planned plug-in system is working. This is a pity as *MuPen* is designed to be processor agnostic. Like *TrueReality*, it has been ported to BeOS and Windows, as well as various Linux setups.

MuPen64 runs *Mario64*, but only in wireframe, and various CPU demos. It has an optional dynamic compiler. It can load '.bin' and '.v64' game and demo image files, automatically unpacking them with 'zlib' if necessary.

Startup is slow as you must answer two French questions in the console window, typing numbers to chose the type of CPU and custom chip emulation you want. Diagnostic messages continue to spew across the console as the emulator runs.

The Rotate demo runs nicely, as does Project Unreality's plasma. The Redbox Mandelbrot zoomer looks good though the speed hardly matches 64-bit standards. The fire demo works (roughly); it's many times faster when using the dynamic compiler than the interpreter, but the 64-bit R4300 is not easy to emulate on a typical 32-bit CPU, either way.

Snes9x

Snes9x is another excellent SNES emulator, mainly written in C with some C++ and optional assembler CPU emulation. It is portable to non-Intel architectures, unlike *ZSnes*, and has been tested on big-endian as well as little-endian CPUs. I tested version 1.37 on the same SuSE/K6 set-up, and found it worked even better than *ZSnes* in some respects, though it lacks the configuration GUI. Relatively complete documentation is included but no man page.

Snes9x has many features in common with *ZSnes*, like adjustable speed and screen shots. It can freeze and restore games, emulate Game Genie and Action Replay cheat cartridges and add-ons like the SNES Mouse, SuperScope and Multi-player 5 widget. Argonaut's SuperFX co-processor is emulated, like the faster SA-1 main processor in some carts and the C4 graphics accelerator used by Capcom in *Megaman X2* and *X3*. DSP-1 is supported enough for *Mario Kart* but not for other games like *Pilotwings*, though *ZSnes* also struggles to emulate those, and *Metal Combat* and *Street Fighter Alpha 2* defeat both emulators.

The minimum host is a DX4/100 with 8-bit graphics and no sound, but you'll need a 16MB P200 or equivalent to run most games. Graphics can use any pixel depth in X windowed or full-screen modes; 15 or 16 bits give the best performance, though 8-bit is fast if you don't care about transparency effects.

ROMs can be in '.fig', '.smc' (SuperMagiCom) or '.sfc' (SuperFamiCom) snapshot format, or split images

Express control

Four joysticks or Sidewinder pads can be supported with the appropriate kernel drivers. The first five digit keys toggle four display layers and sprites on and off, followed by controller assignments with keys 6 and 7, then various graphics options on the remaining number keys. The `-` and `=` keys can speed up or slow down emulation by frame skipping.

Tab puts the emulator into turbo mode, the bracket keys toggle sound options, Scroll Lock pauses and Esc quits. Function keys are used to save and reload game states and saved positions. Frozen games are a lot bigger than the save files which would normally be written to CMOS RAM on a real SNES, but can be saved at any time rather than just when the game designers planned.

Mario takes to the track in *Mario Kart* on *Snes9x*.

***Snes9x* is set up and launched from this neat control panel.**

You can specify PAL or NTSC video mode on the command line, plus sound, joystick and video options similar to those for *ZSnes*. The '-is' switch interpolates sound samples, making low-frequency sounds smoother. It is an improvement but still rather prone to hiss compared with the curve-fitting done by *ZSnes*.

Snes9x starts in a small window using one pixel for each on the original system, and can manage 60 frames per second at that size. Scaling up to around 800 by 600 pixels drops updates to about one frame in 3, which is still quite playable.

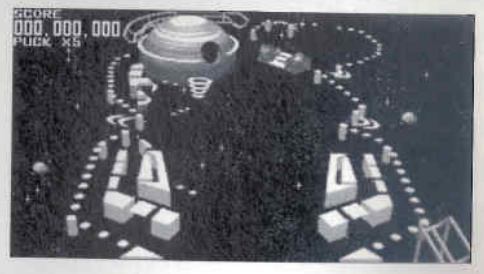
Good video and audio options on most ROMs are '-y -sy -sc -r 7 -dfr -sy'. These deliver about 40 fps in a quad-sized window on my system. On a slow machine try '-j -ne -r 1 -dfr -mono' in an 8-bit video mode.

Reality Boy

Reality Boy emulates an obscure 3D portable, the Virtual Boy, which used separate red displays for each eye, giving convincing 3D but in shades of red. It's more a technology demonstrator than a fully-developed product, which is probably why it never got a worldwide launch, but now you can try it on Linux.

The obscure hardware is emulated well enough to run four games fully and half a dozen more to some extent. You can run it in mono or with separate displays for each eye, but that's missing the point.

The 3D effect can be ingeniously supported through the use of 3D glasses like those used for 3D films and printed novelties, or LCD shutters like the iGlasses once promoted by Escom.



***Reality Boy's Pinball demo*, in eye-preserving mono.**

And there's more!

As if the eight bit Nintendo emulators tested last year were not enough, we've found Goober, another GameBoy emulation, and put it on your coverdisc to complete your eight bit Nintendo collection. You'll also find Reality Boy, several Super Nintendo and N64 emulators, demo ROMs and documentation on the disc. There are already three Game Boy Advance emulators for Linux, all developing rapidly. We've run out of room to discuss them this month, but will return to test them and other Japanese console emulators in a future column. LXF

Nintendo Links

Fake64 home: http://insomniac.ath.cx:81/~bluefyre
Goober: http://goobergbe.sourceforge.net
MuPen64: http://mupen64.emulation64.com
N64 demos: http://www.dextrose.com
N64 Linux: http://www.ix.de/ix/artikel/E/1997/04/036/
Reality Boy: http://www.goodnet.com/~dh90791
SNES demos: http://www.cherryroms.com
SNemul home: http://snemul.free.fr
Snes9X home: http://www.snes9x.com
TrueReality: http://www.emuhq.com/truereality/unix
ZSnes source: http://zsnes.sourceforge.net
ZSnes home: http://www.zsnes.com