The GP2x PDA

A PDA focused on games and GNU/Linux

Steven Goodwin

Games under GNU/Linux have usually been a lacklustre affair. For every Tux Racer, there are a hundred sub-standard Pac-man clones you'd be embarrassed to advocate. For every commercial version of Quake, there's a hundred other worthy games the publisher elected not to port to GNU/Linux. Without good games, there's no market, and without the market, no effort is spared. And so the cycle continues. In this article, I will look at two of the areas in which GNU/Linux games have succeeded, and a new device that combines them both, which could help expose GNU/Linux to the populous.

Two success stories

There are two main areas in which GNU/Linux games appear to thrive. The first is through the home-brew market of 2D platform and arcade games. Many of these are not necessarily written for GNU/Linux as a platform, but developed using a cross-platform API enabling a Windows code base to be ported across to GNU/Linux with the minimum of effort and vice-versa.

The most popular API for this task is SDL.

SDL

SDL (Simple DirectMedia Layer) is a low-level game API that supports cross-platform development for graphics, audio and joystick input. It isn't the only cross-platform API available (Allegro and G2 spring to mind in the 2D arena) but it's probably the most widespread. Having been started in 1997 by Sam Lantinga, it has grown beyond its humble beginnings, and now supports a myriad of platforms across most operating systems. There are several GNU/Linux ports, including those for PlayStation Linux and Dreamcast Linux, and the API has been used in countless games because of its simplicity of use, stability, and power.

Emulation

It is often argued that one of things free software developers do better than anyone else is re-engineer (that is, rebuild existing software in a better way). Whether it's true or not, the free software community has provided the world with a wealth of good emulation software, encompassing an entire range of 8- and 16-bit games consoles and home computers. It is for this reason that emulation can be considered the second of the big games-oriented contributions by free software.

Enter the GP2x

One place where the SDL and emulation meet is with the GP2x. This is a new handheld games console designed and built by GamePark Holdings (GPH), and was released in South Korea in November 2005. To a very large degree it has been marketed as a portable entertainment player, and comes ready to play DivX and XviD movies, view photographs, read text files, and play MP3 and Ogg Vorbis

The GP2x

music tracks. However, coming from the same genealogy as the GP32, GPH have adopted a games-style form factor with joystick, shoulder buttons and, now standard, ABXY controls.

In any other life, the GP2x would be the illegitimate offspring of a game console and an MP3 player. But, what sets the GP2x apart from other media players and PDAs is its use of embedded GNU/Linux as an operating system. Naturally, this allows anyone to develop for it or even change the underlying system itself.

The use of GNU/Linux has allowed the GP2x to harness the rabid user base of developers and bedroom coders giving them a real opportunity to work on a handheld console. Armed with free software, developers can build software for the GP2x without the corporate intervention of Sony or Nintendo.

What’s so special about a handheld console?

The first computer games were simply board games played on computer; multi-player games were single player games that supported more players; and network games were multiplayer games with a network option. So, the first generation of handheld games were (and are) “normal” games compressed to play on a handheld. Previously, the impetus to do something new was controlled by the gilded minority at Nintendo HQ. But in the hands of creative free software developers, these boundaries can be removed with the availability of a free console, such as the GP2x, and a free development environment. And unlike the Indrema, this exists!

However, from the point of view of many developers, a handheld console introduces nothing but problems: it has a small screen, limited input, poor audio, and a fixed hardware set eliminating any attempt at scalability. Despite these apparent limitations, the machine is capable of: running a respectable version of Quake; running Mame; emulating the Commodore 64, ZX Spectrum and NES, among many others; and playing movies.

The current generation of GP2x software is focused in the games arena, with developers writing and porting a wealth of emulators to the console. But, emulation gives the opportunity to provide casual games with little outlay. Then, native applications are only a step away for those who wish to push in new directions, since the nature of handheld devices changes the type of games that are possible, and available.

Behind the scenes

The device itself is a relatively compact 143.6 × 82.9 × 34 mm, but still holds two ARM CPUs (both running at 200 MHz), an SD (Secure Digital) reader and 64 MB of non-volatile NAND memory. This is done by employing the MP2520F “system on a chip” from MagicEyes, which combines both ARM chips, along with the video and graphic processors and various I/O components in a single component form factor.

The first ARM CPU is the 920T, a general purpose processor which runs the Linux system stored on the NAND memory like a traditional memory drive. The second is a co-processor 940T with a smaller cache and no MMU which (among other things) performs hardware scaling, allowing DivX movies to be played from their original encoding without affecting the decoding process. This will scale the movies down to the 320 × 240 screen size when played on the GP2x, or provide a full 720 × 480 resolution when viewed on a connected TV (using an optional TV output cable).

The NAND memory runs JFFS2 (as used on Familiar handhelds) and holds a complete GNU/Linux installation incorporating the standard file structure of `/bin`, `/etc` and `/usr`. While it is possible to store data and new libraries in this memory, it is not possible to do so from the USB connection. So, you must instead transfer it from the supplementary SD card (making it a mandatory addition). This is mounted on `/mnt/sd`. You can write to the SD either

GP2x Specification

Technical information

CPU (dual core)	ARM 920T (200 MHz, MMU) and ARM 940T (200 MHz, no MMU)
NAND Flash memory	64 MB
RAM	64 MB (each core is able to access 32 M B of it)
Storage	Secure Digital (SD) memory card, sizes up to 1 GB are supported, although version 1.10 of the firmware reports an upper limit of 2 GB
Connection	USB 2.0, without host control
Display	320 × 240, 3.5 inch TFT LCD
I/O	Stereo speakers. Headphone socket. TV output cable (available seperately)
Power	2 × AA batteries (but only rechargeables make sense!), 3.3 V AC (adapter not supplied)

through the GP2x's USB port, or by transferring it with a standard SD reader plugged into your host machine. There have been several reports of problematic USB connections, which can be solved in many cases with new firmware. Note, however, that unlike some PDAs, the GP2x doesn't like you removing the SD card without first switching the machine off (there is no unmount interface). This is one instance where the duality of Linux machine and games machine conflict.

The machine itself runs a Linux 2.4.25 kernel, using the framebuffer for graphics. It also has a large set of system binaries provided (mostly) by `busybox`—a size-optimised executable which implements a number of standard GNU/Linux shell commands. Each traditionally named command is symlinked to `busybox`, which examines its first argument (`argv[0]`, which holds the command used to call it) and executes the appropriate code. This allows it to have a larger selection of commands in a smaller space. The distribution is compact, but usable for the basic tasks of running a games machine. Since it has no real networking abilities (the USB connection can not work as a host, for example), very little is needed outside the basic support for localhost.

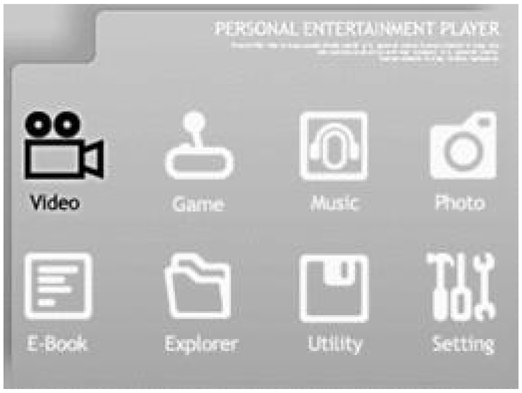
In true games console style, this hardware is fixed and will not change during the life of the machine. This provides a safe, stable backdrop upon which to develop. If it runs fine on one machine, it'll work on them all (unlike GNU/Linux). This could help the GP2x succeed where the Indrema failed.

In operation

After booting up, the GP2x displays an eight-way menu, shown in Figure 1, which acts as a program launcher. Using the joystick to navigate, and the B button to select, you can call up games, utility scripts or play different types of media using the supplied software. Because this program sits on top of GNU/Linux, it employs specific customs not usual for the underlying operating system. To begin with, file extensions play an unconventionally important role, with the `Utility` menu allowing you to browse only files ending in GPU, and the `Game` menu showing files ending exclusively in GPE. In return it provides a consistent (if unpretty) interface to the device like all other consumer electronics.

The main menu, along with the eight submenus, are all handled by custom software inside `/usr/gp2x` each with

Figure 1: Main menu



Essential GP2x links

- Official UK site (`http://www.gbax.com/`)—the basic starting point
- GP2x Archive (`http://archive.gp2x.de/cgi-bin/cfiles.cgi`)—the ultimate software repository for the GP2x
- GP2x notes (`http://www.bluedust.com/gp2x`)—the authors GP2x page
- User guide (`http://archive.gp2x.de/cgi-bin/cfiles.cgi?0,0,0,0,23,933`)—as created by the community
- sterm (`http://archive.gp2x.de/cgi-bin/cfiles.cgi?0,0,0,0,8,973`)—a bash terminal program
- Code snippets (`http://www.gp32.co.nz/snippet.php`)—includes assembler
- Minimal library (`http://www.emulnation.info/retrodev/forum/files/rlyeh\OT1\textquoterights-minimal-library-v0.A.zip`)—for programming without SDL
- fbgrab (`http://archive.gp2x.de/cgi-bin/cfiles.cgi?0,0,0,0,8,890`)—a screen grab utility

their own graphic files. This has enabled a sector of the community to produce different skins, include ones in the style of The Matrix, and the Amiga Workbench. Naturally, it is also possible to change these programs entirely at your whim, and another part of the community is looking into developing alternative program launchers.

The primary supplied software is as follows.

Standard GP2x Software

- Video playback—This is handled by a custom version of `mplayer` although the interface is minimal (don't expect the control of `gmplayer`). However, it's functional enough (and mostly smooth enough) to be watchable.
- Music playback—This also uses `mplayer` for the playback engine, but it's supplemented with a GUI.
- Photo viewer—This is a simple JPG and BMP viewer contained within the main menu executable. It is usable for previewing holiday snaps, but currently contains no zoom option. You may find it necessary to downscale your images when using this program because it can take a couple of seconds for large JPGs to be decompressed, and anything over 1024×768 in size appears to not load at all.
- EBookViewer—This is probably the most disappointing part of the package since the word "ebook" is little grandiose, as the software is less than less! It is a plain text viewer only, with minimal control to go forward and backwards. Bookmarks are planned for the future.
- Explorer—This program lets you traverse the directory of your SD card (only) and delete files. Like its namesake in the Windows world, this displays only the filename, omitting the file system properties.
- Game and Utility—These two options simply run programs, as mentioned above.
- Settings—This elusive menu provides the menu option which initiates a USB connection to your host machine. It also shows the current battery level, switches on the TV out facility and details the system info. There is also a utility here to monitor the controls.

For developers

As a development environment, writing for the GP2x is little different to any of the standard flavours of GNU/Linux. However, its work process is more akin to the console industry where all the code is cross-compiled from a PC (running either GNU/Linux, MacOS or Windows) into ARM assembler, ready for the GP2x. Naturally, GNU/Linux is easier since you can compile the same code natively to test it, without having to re-write any of the OS-specific components.

One notable difference when running on the GP2x is that every program is run as root. On a single-user games device, like the GP2x, this isn't the security hole it would be elsewhere, but it does require more trust on the part of the application, particularly as the second ARM CPU has no MMU, meaning any faulty application can crash not only itself, but the whole GP2x. The speed difference between both a host (any host) and target machine is also very marked, so high power applications are never going to catch on. Even PDF viewing is currently a struggle.

GamePark Holdings' development API of choice is SDL, which provides an easy in-road for porters and developers since it is well known in the community and comes with a lot of existing applications and on-line documentation. To some extent this mirrors the practices of "real" console manufacturers who supply an SDK with their new console to would-be developers. And although SDL isn't installed by default on the GP2x, most games and emulators use it, and is therefore a ubiquitous appearance on the machine. However, for those that want a lower level of access there is also a minimal library that writes directly into the memory registers. See the links at the end of the article for more information.

You can read about the development process on the GP2x wiki (`http://wiki.gp2x.org/wiki/Getting_started_with_GP2x_development`).

DRM

No article on the GP2x would be complete without a note on the topic of DRM—but that is all it's getting! Basically, there is no DRM on any of the music, video, text or game files that you can copy freely around the system, but GPH have said there will be a DRM facility to protect commercial games, as already exists on the Xbox, Xbox 360 and PlayStation 2. However, there is no clue as to how (or if) that will happen, or whether it will eventually affect other media. With the kernel source and SDK freely available, however, it's unlikely they can do anything damaging, technically speaking. GPH have already shown themselves to be consumer-aware, rushing units out without adequate testing lest they withdraw their orders, so they will probably play fair on this, but only time will tell.

Invented standards

Because of the split between NAND and SD memory, there is one development choice you need to take seriously: static and dynamic linking. If your software uses only SDL (or another common set of libraries), there's no problem with creating an application that uses it as a dynamic library. However, with limited space in the onboard memory (less than 20 MB with the current install), you are advised to statically link applications. Or, at least to dynamically link to libraries on the SD card, and modify `LDPATH` when you start your application. This highlights another difference between the Linux and game worlds; since Linux has a standard directory structure, and games generally assume they own the whole machine, the decision of where to place particular files can differ between games. This extends to whether the software should be placed in the root of the SD, or within a subdirectory of it.

The current thinking is to launch all code through a preceding bash script that can prepare the environment to its desire, run the executable, and then re-run the launcher menu when it exits. This is brought home when you realise there is no packaging system, and dependencies are left to the individual developers.

Finally, remember that as this is also focused as a games machine (running as root), you *are* in control of the whole machine. This includes the buttons. So if your application uses sound, you must manually handle the volume control, the return to menu button and so on. This also breaks with GNU/Linux tradition.

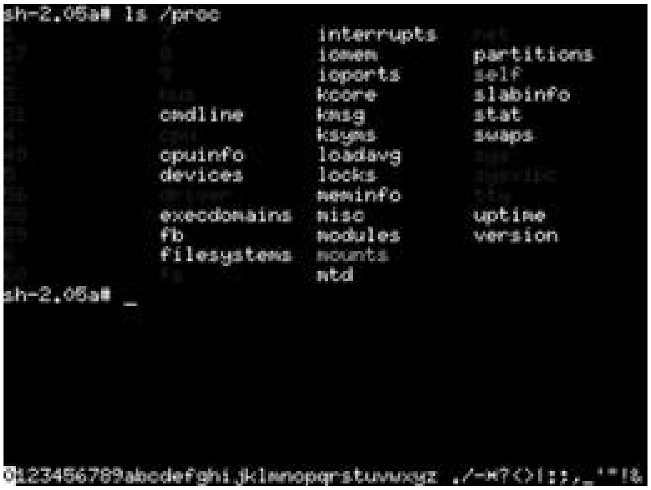
Conclusions

In its intended capacity as a media player it works quite well (save the problematic earphone socket.) Most media that I've ripped works well on the device, although the interface is a little amateur in places. For example, the main menu does not support "down" on the joystick, and "X" allows you go to back to the previous menu in all cases... except back to the main menu, where you have to press "Start". While the free software community is used to suffering minor quirks on the initial release, the general public is not and so are unlikely to leave their iPods at home, in favour of the GP2x.

As a retro games machine, the GP2x shows its vocational colors. The physical form factor, styling and current software catalog of the GP2x all lend themselves to be *the* machine for gamers that haven't grown up! While it doesn't have the panache of a Sony PSP, or the dependability of the GBA, it does provide the best way to play retro games. Ever. Even sitting in front of an original Atari 2600 at home doesn't have the coolness of playing the original Zelda on the pub sofa with a GP2x.

However, this is an early adopter project. As mentioned previously, the system lacks polish in some areas (in both hardware and software) and could certainly do with a published set of standards to the bridge the gap between the current

Figure 2: Running STerm



```
sh-2.05a# ls /proc
1           7             interrupts    net
17          8             iomem         partitions
2           9             ioports       self
3           bus           kcore         slabinfo
31          cmdline       kmsg          stat
4           cpu           ksyms         swaps
45          cpuinfo       loadavg       sys
5           devices       locks         sysvipc
56          driver        meminfo       tty
58          execdomains   misc          uptime
59          fb            modules       version
6           filesystems   mounts
60          fs            mtd
sh-2.05a# _




0123456789abcdefghijklmnopqrstuvwxyz ./-*?<>|:;,_'"!&
```

GP2x and the current handheld markets. So, provided Game Park Holdings release all of the GPL code they are required to (the kernel wasn't released until nearly two months after initial shipping) the developer community should crowd around this beast and bring it to the attention of a deserving world. Hopefully, this will show GNU/Linux games in a new light, and provide a new generation of talented, innovative, games programmers.

Copyright information

About the author

When builders go down to the pub they talk about football. Presumably therefore, when footballers go down to the pub they talk about builders! When Steven Goodwin goes down the pub he doesn't talk about football. Or builders. He talks about computers. Constantly…