PORTING GNU/LINUX TO Xbox

Milosch Meriac - 19C3@meriac.de

Who am I

- Milosch Meriac, freelancer
- focused on embedded systems
- reverse engineering
- linux and windows kernel drivers
- lowlevel programming / realtime

Which operating system for Xbox ?

- Open/Free/Net-BSD
- GNU/HURD
- GNU/Linux
- Proprietary OS
- Windows 9x/SE/ME
- Windows NT/2000/XP

Why not MS-Windows - Battle Royale ?

- It's nothing personal

Why not MS-Windows - Battle Royale ?

- It’s nothing personal
- High license fees

Why not MS-Windows - Battle Royale ?

- It’s nothing personal
- High license fees
- Not allowed to patch binaries (EULA)

Why not MS-Windows - Battle Royale ?

- It's nothing personal
- High license fees
- Not allowed to patch binaries (EULA)
- Mostly close source

Why not MS-Windows - Battle Royale ?

- It's nothing personal
- High license fees
- Not allowed to patch binaries (EULA)
- Mostly close source
- Lack of Open Source Windows Drivers, Applications and Frameworks.

Why Linux or: How I Learned To Stop Worrying And Love The Bomb

- Open Source - we need to tweak a lot
- liberal GNU GPL license model

Why Linux or: How I Learned To Stop Worrying And Love The Bomb

- Open Source - we need to tweak a lot
- liberal GNU GPL license model
- allows closed source kernel modules

Why Linux or: How I Learned To Stop Worrying And Love The Bomb

- Open Source - we need to tweak a lot
- liberal GNU GPL license model
- allows closed source kernel modules
- well-known and user-friendly (KDE3/Gnome)
- wide range of applications
- sophisticated development tools

Why Linux or: How I Learned To Stop Worrying And Love The Bomb

- Open Source - we need to tweak a lot
- liberal GNU GPL license model
- allows closed source kernel modules
- well-known and user-friendly (KDE3/Gnome)
- wide range of applications
- sophisticated development tools

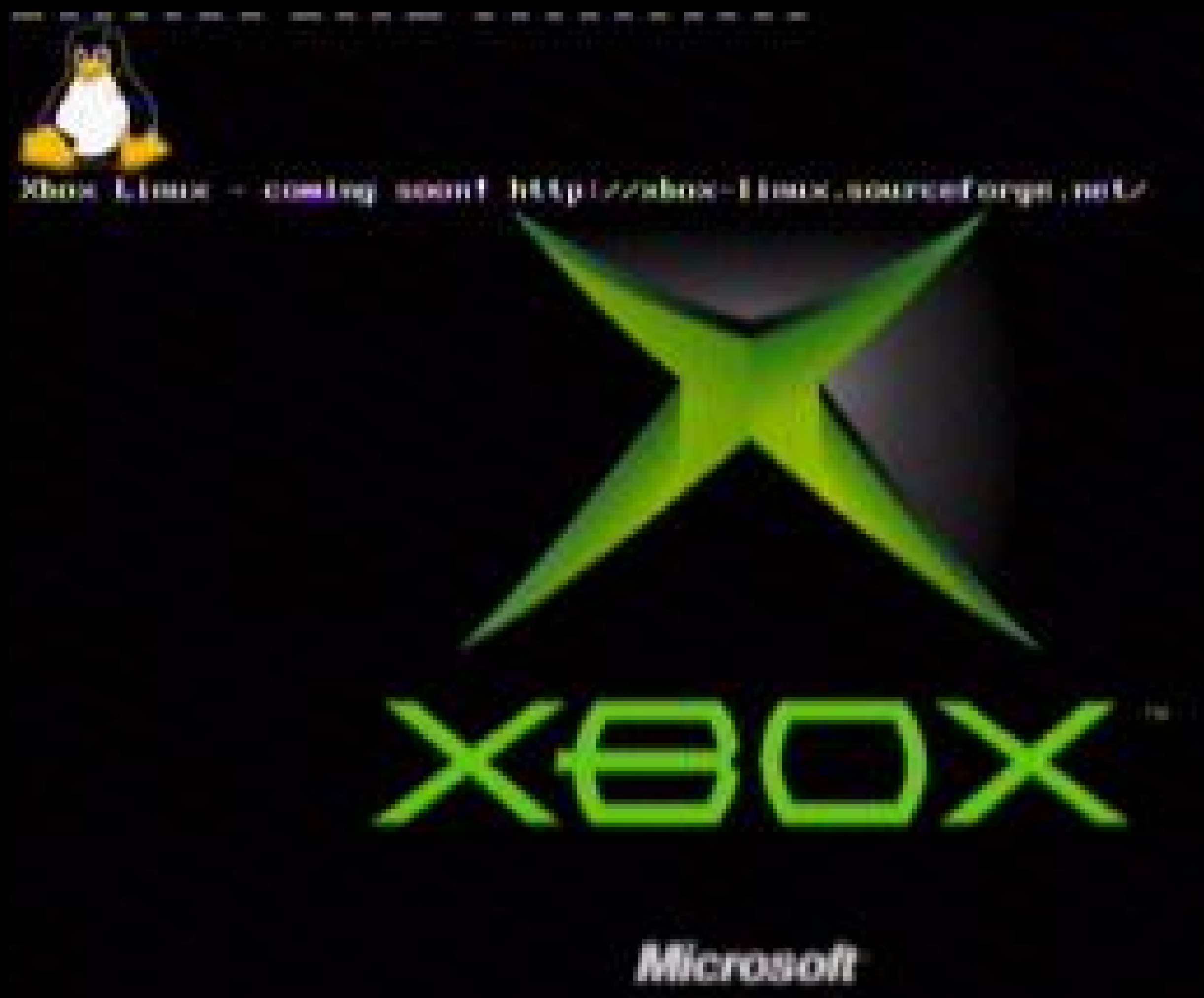
- we simply like it most, so we started using linux
- boost the spread of linux

The Linux Porting Process - They Who Step On Tiger's Tail

- In the beginning there was darkness...

XBOX™
Microsoft

then Michael Steil spoke in the darkness: ”Let there be light”



```
Xbox Linux - coming soon! http://xbox-linux.sourceforge.net/
```

And right away there was light, scattering the darkness and showing the infinite space. ”That’s good!” said Michael

- We can now run unsigned code
- Built Xbox Executable Binary from scratch (no XDK needed) which copied the Tux-Logo into initialized video RAM area
- replacing whole copyrighted microsoft code within Xbox flash was not satisfying, because of no video output

The Linux Porting Process - A Space Odyssey

- created a bootloader like GNU/GRUB or LILO

The Linux Porting Process - A Space Odyssey

- created a bootloader like GNU/GRUB or LILO
- minimzed side effects of the Xbox Kernel using a clean ROM-Replacement
- perfectly legal solution

The Linux Porting Process - A Space Odyssey

- created a bootloader like GNU/GRUB or LILO
- minimzed side effects of the Xbox Kernel using a clean ROM-Replacement
- perfectly legal solution
- to prevent 16-Bit fiddeling we skipped realmode initialization of kernel

Our first replacement ROM - Dark Star

- we see ...

Nothing to see - The Emperor’s New Clothes

- Linux boots, but we see nothing
- no video output at all, because there were neither BIOS routines for screen setup, nor a VGA BIOS

Our second step

- used Andy Greens filtror device as a simple kernel debug console
- a bidirectional memory buffer allowed us to exhange data with the Xbox
- device is being polled by the linux kernel and the client computer
- device allowed us to see the kernel messages on a terminal

- a dirty hooking-the-kernel-kprintf-routines-hack shows us the first ”hello world”-lifesigns of the kernel

- a dirty hooking-the-kernel-kprintf-routines-hack shows us the first ”hello world”-lifesigns of the kernel

- was *really* dirty, unstable and passive

- a dirty hooking-the-kernel-kprintf-routines-hack shows us the first ”hello world”-lifesigns of the kernel
- was *really* dirty, unstable and passive
- wrote a console kernel driver especially for the filtror device
- the new driver allowed us to see the all boot messages

- a dirty hooking-the-kernel-kprintf-routines-hack shows us the first ”hello world”-lifesigns of the kernel
- was *really* dirty, unstable and passive
- wrote a console kernel driver especially for the filtror device
- the new driver allowed us to see the all boot messages
- after fixing some nVidia PCI Chipset bugs and mainly minor Xbox-specific problems - Linux is booting cleanly!

Things started to roll - Odyssey two

- we created a Linux distribution from scratch

Things started to roll - Odyssey two

- we created a Linux distribution from scratch
- used busybox the *Swiss Army Knife of Embedded Linux*

Things started to roll - Odyssey two

- we created a Linux distribution from scratch
- used busybox the *Swiss Army Knife of Embedded Linux*
- busybox running on top of uClib, a C library for embedded systems
- powerful embedded linux system
- bzip2'ed kernel and the gzip'ed initial ramdisk fits perfectly into one megabyte Xbox-Flash

Things started to roll - Odyssey two

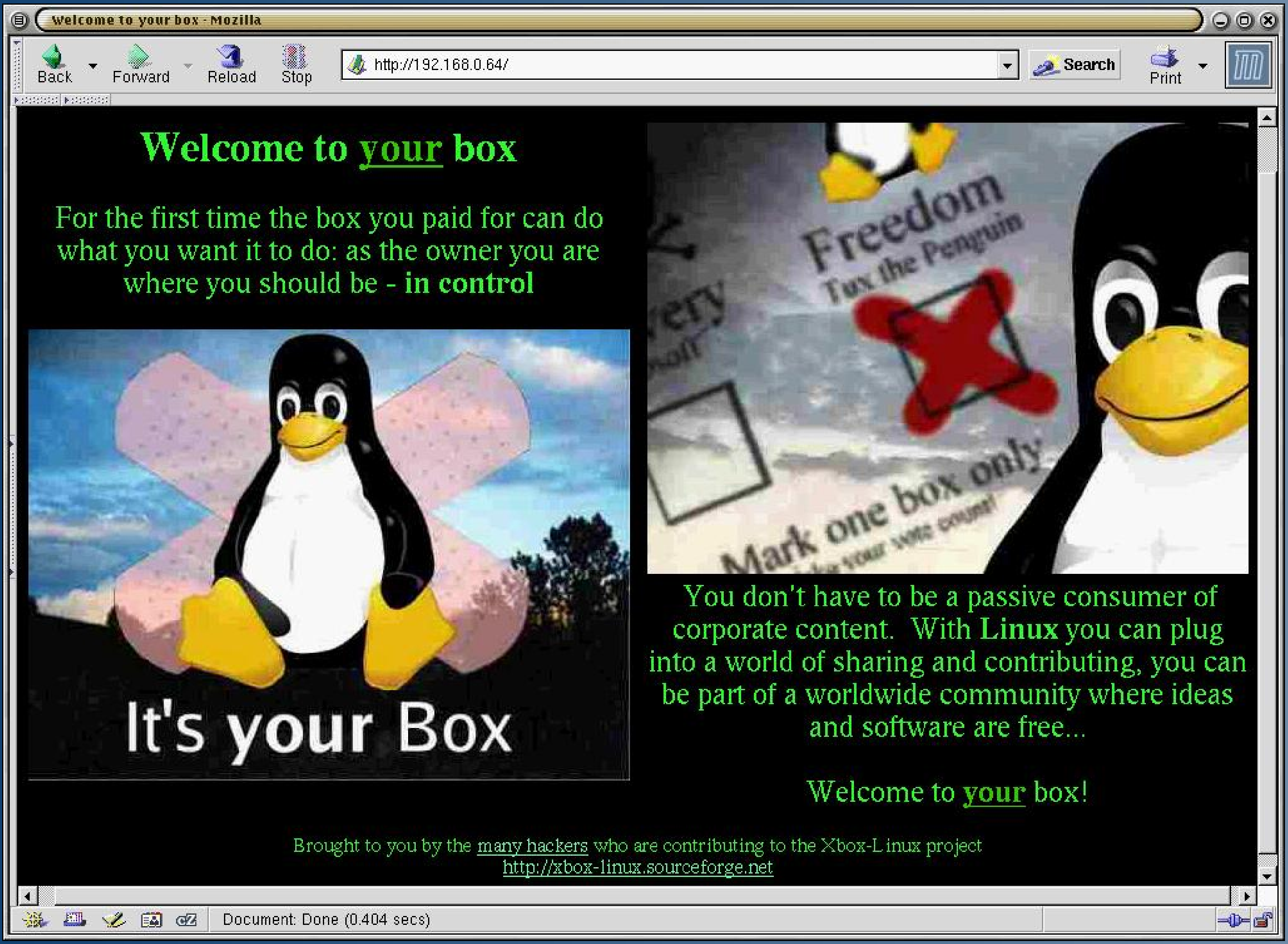
- we created a Linux distribution from scratch
- used busybox the *Swiss Army Knife of Embedded Linux*
- busybox running on top of uClib, a C library for embedded systems
- powerful embedded linux system
- bzip2'ed kernel and the gzip'ed initial ramdisk fits perfectly into one megabyte Xbox-Flash

- filtror console driver gave us access to linux console on Xbox
- everything is working fine
- managed to get the closed source nVidia network drivers working inside the Xbox Linux environment

Our first Release

- Linux boots into a network-enabled state
- running a web server and telnet daemon
- although there is no audio or video output and input device connectivity yet, users have full control on the Xbox through the network
- Xbox based internet servers appear on the internet

- developers worldwide are now able to add more features
- ability to mount NFS shares
- includes large number of linux command line tools
- soundcard enabled - mp3players appeared
- the embedded webserver provides detailed information about the Xbox



Welcome to your box

For the first time the box you paid for can do what you want it to do: as the owner you are where you should be - **in control**

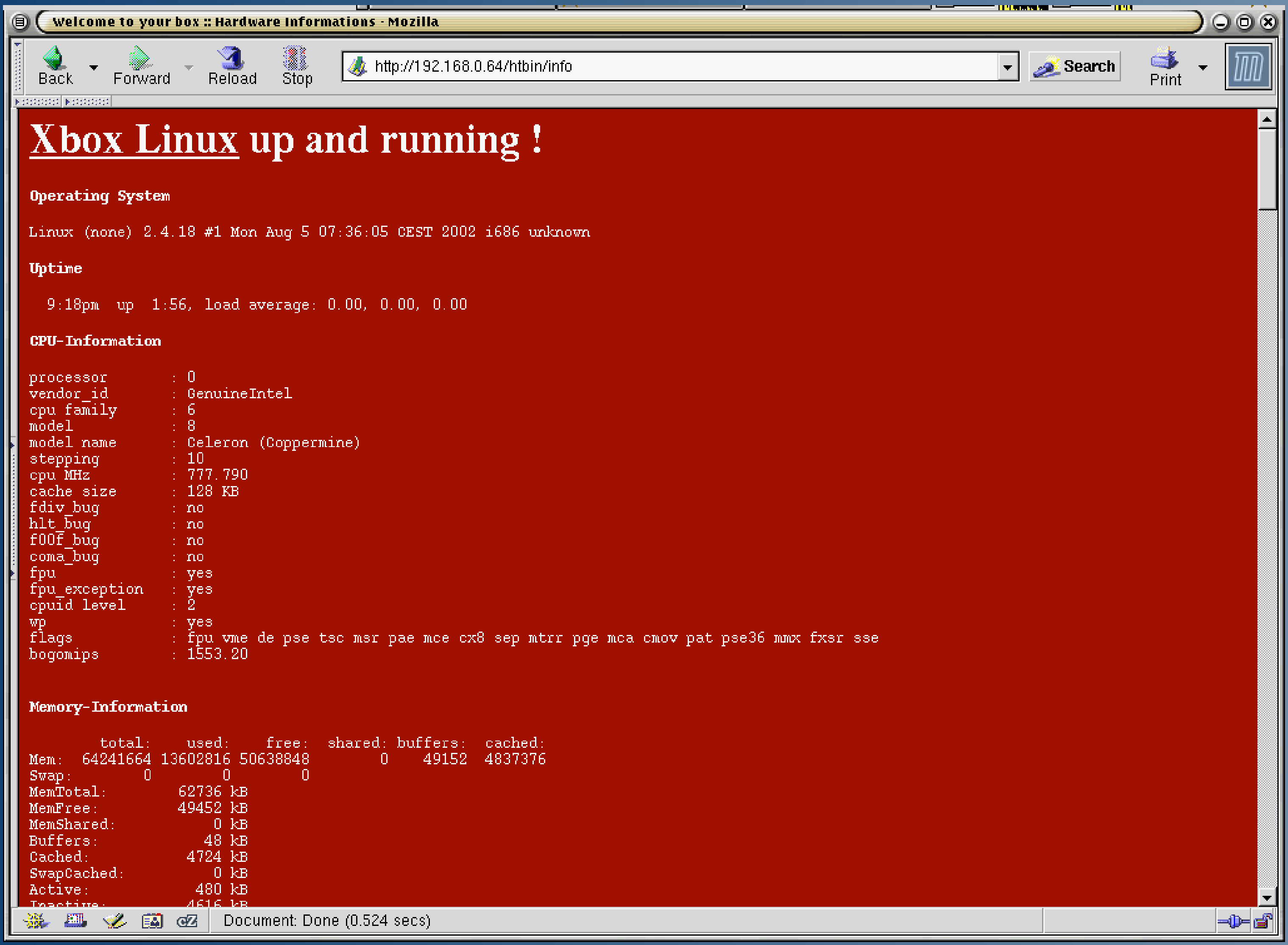
You don't have to be a passive consumer of corporate content. With **Linux** you can plug into a world of sharing and contributing, you can be part of a worldwide community where ideas and software are free...

Welcome to your box!

Brought to you by the many hackers who are contributing to the Xbox-Linux project
http://xbox-linux.sourceforge.net

Document: Done (0.404 secs)

Figure 1: The Embedded Webserver



Welcome to your box :: Hardware Informations - Mozilla
Back
Forward
Reload
Stop
http://192.168.0.64/htbin/info
Search
Print

Xbox Linux up and running !

Operating System

```
Linux (none) 2.4.18 #1 Mon Aug 5 07:36:05 CEST 2002 i686 unknown
```

Uptime

```
  9:18pm  up  1:56,  load average: 0.00, 0.00, 0.00
```

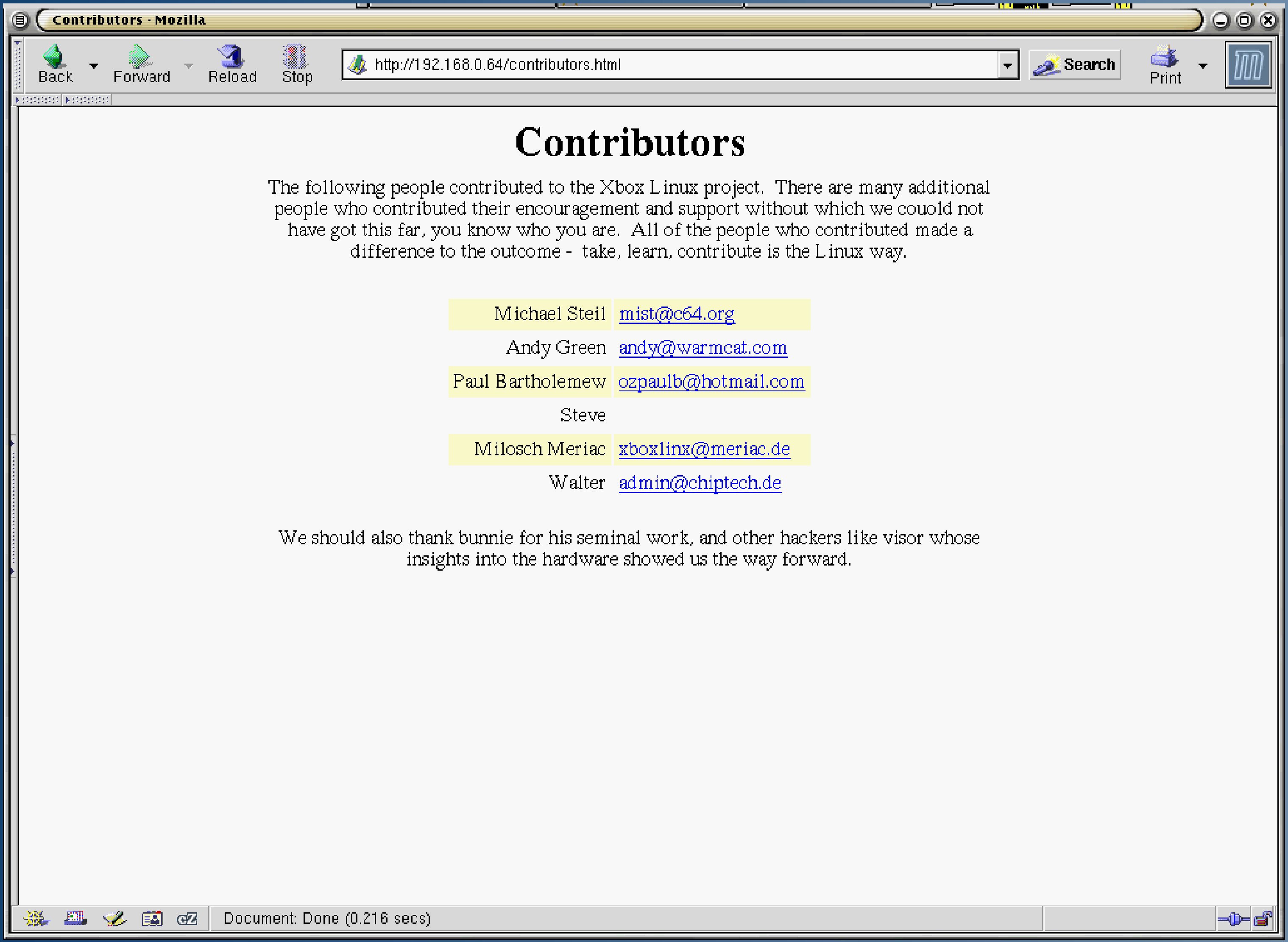
CPU-Information

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 8
model name      : Celeron (Coppermine)
stepping        : 10
cpu MHz         : 777.790
cache size      : 128 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 sep mtrr pge mca cmov pat pse36 mmx fxsr sse
bogomips        : 1553.20
```

Memory-Information

```
        total:    used:    free:  shared: buffers:  cached:
Mem:  64241664 13602816 50638848        0    49152  4837376
Swap:        0        0        0
MemTotal:        62736 kB
MemFree:         49452 kB
MemShared:           0 kB
Buffers:            48 kB
Cached:           4724 kB
SwapCached:          0 kB
Active:            480 kB
Inactive:         4616 kB
```

Figure 2: System information provided by CGI script



Contributors - Mozilla
Back
Forward
Reload
Stop
http://192.168.0.64/contributors.html
Search
Print
Document: Done (0.216 secs)

Contributors

The following people contributed to the Xbox Linux project. There are many additional people who contributed their encouragement and support without which we couold not have got this far, you know who you are. All of the people who contributed made a difference to the outcome - take, learn, contribute is the Linux way.

Michael Steil	mist@c64.org
Andy Green	andy@warmcat.com
Paul Bartholemew	ozpaulb@hotmail.com
Steve	
Milosch Meriac	xboxlinx@meriac.de
Walter	admin@chiptech.de

We should also thank bunnie for his seminal work, and other hackers like visor whose insights into the hardware showed us the way forward.

Figure 3: Xbox Team Information

Within few days the count of developers grew enormously

The next steps of the team included:

- added USB interface for mouse and keyboard support
- added FatX-Filesystem support
- added added support for Xbox-Controller
- created a ROM-Image which can be operated by USB-Keyboard on Xbox

ROM image was becoming too tight

- we created a Xbox Executable Bootloader for booting Linux from CD-RW or DVD

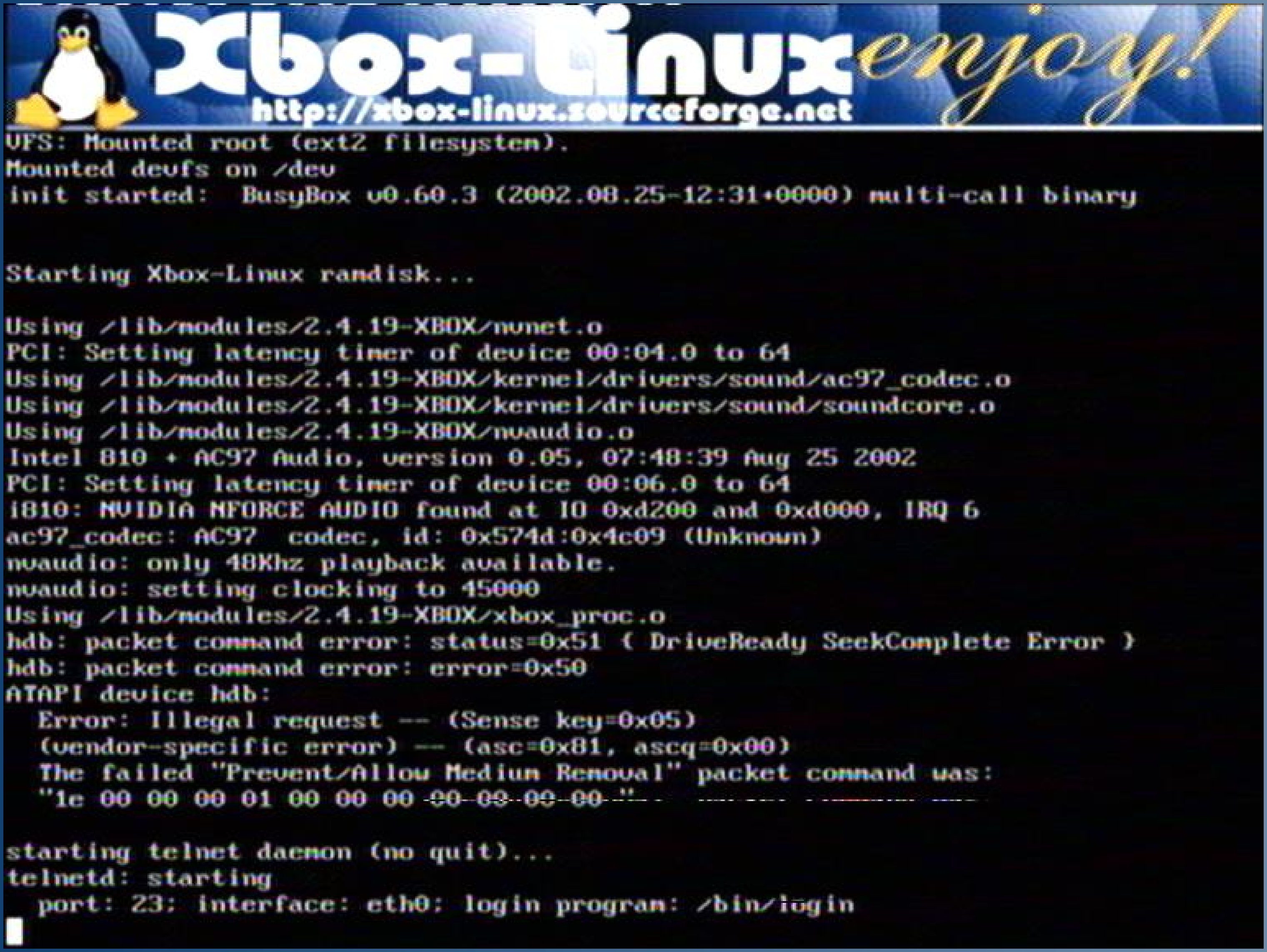
- this enables us to access initialized video RAM

ROM image was becoming too tight

- we created a Xbox Executable Bootloader for booting Linux from CD-RW or DVD

- this enables us to access initialized video RAM

- enabled the framebuffer console driver by adding a Xbox framebuffer interface description to the kernel.



```
VFS: Mounted root (ext2 filesystem).
Mounted devfs on /dev
init started:  BusyBox v0.60.3 (2002.08.25-12:31+0000) multi-call binary


Starting Xbox-Linux ramdisk...

Using /lib/modules/2.4.19-XBOX/nvnet.o
PCI: Setting latency timer of device 00:04.0 to 64
Using /lib/modules/2.4.19-XBOX/kernel/drivers/sound/ac97_codec.o
Using /lib/modules/2.4.19-XBOX/kernel/drivers/sound/soundcore.o
Using /lib/modules/2.4.19-XBOX/nvaudio.o
Intel 810 + AC97 Audio, version 0.05, 07:48:39 Aug 25 2002
PCI: Setting latency timer of device 00:06.0 to 64
i810: NVIDIA NFORCE AUDIO found at IO 0xd200 and 0xd000, IRQ 6
ac97_codec: AC97  codec, id: 0x574d:0x4c09 (Unknown)
nvaudio: only 48Khz playback available.
nvaudio: setting clocking to 45000
Using /lib/modules/2.4.19-XBOX/xbox_proc.o
hdb: packet command error: status=0x51 { DriveReady SeekComplete Error }
hdb: packet command error: error=0x50
ATAPI device hdb:
  Error: Illegal request -- (Sense key=0x05)
  (vendor-specific error) -- (asc=0x81, ascq=0x00)
  The failed "Prevent/Allow Medium Removal" packet command was:
  "1e 00 00 00 01 00 00 00 00 00 00 00 "

starting telnet daemon (no quit)...
telnetd: starting
  port: 23; interface: eth0; login program: /bin/login
█
```

Figure 4: Xbox Linux v0.2 TV screenshot

XWindows up and running !

- XServer 4.x is running
- Xbox can be used used as XTerminal
- The first DivX films with sound are running smooth on Xbox Linux

Interesting Links

- http://Xbox-linux.sourceforge.net/
  The Xbox Linux Project
- http://www.busybox.net/
  The Swiss Army Knife of Embedded Linux
- http://tinylogin.busybox.net/
  The worlds smallest login/passwd/getty/etc
- http://www.uclibc.org/

uClibc - a C library for embedded systems

- http://www.linuxfromscratch.org/
  Linux From Scratch

- http://www.boa.org/
  Boa Web Server

- http://www.acme.com/software/thttpd/
  thttpd - tiny/turbo/throttling HTTP server

- http://www.pengutronix.de/software/utelnetd_en.html
  A small Telnet daemon

Figure 5: The porting of major distributions like Debian, SuSE and Mandrake has begun