Linux: Embedded, Turnkey and Real Time

UseLinux, Jan 9, 1997

by

Phil Hughes, Publisher

Linux Journal

P.O. Box 85867

Seattle, WA 98145

phil@ssc.com

The Linux operating system was designed from scratch, originally for the Intel 80386 processor. Unix, a similar operating system of much older design has been used in embedded and turnkey applications and there are even real-time versions of Unix available. This session will look at Linux for these sorts of applications and help show where Linux may be the right solution.

In this session I will present some basic ideas and some examples. I then encourage input and discussion from the attendees.

CONTENTS

WHAT IS LINUX?

- **Multi-user, multi-tasking operating system**
- **Standard interface at the system call level**
- **Available for multiple platforms**
- **Per copy cost of $0**
- **Commercially viable**

IS LINUX COMMERCIALLY VIABLE?

Is Linux Reliable?

- Experimental and production releases
- 100,000 beta testers
- Commercial vendors

Will Linux inter-operate?

- Supports traditional Unix interoperability options
- Linux mounts other filesystems
- Interoperates with MS-Windows, Mac and Novell

IS LINUX COMMERCIALLY VIABLE?

Is Linux supported?

- Hardware/Software Vendors
- Consultants
- VAR, OEM and ISV Programs
- Growing training pool

Does Linux have needed capabilities?

- Rich set of development software
- “Office” applications
- Graphics
- Databases
- WWW and Java support
- Connectivity

DEFINITIONS

While these may not be *official* definitions they will help us understand each other.

- **Embedded—An application where the computer system is not accessed from the standard console. It may be stand-alone (like a router) or part of a much larger system (like an automobile).**
- **Turnkey—An application where a standard computer is being used complete with keyboard and console display. The user receives a system that solves a specific problem rather than a computer to "run software on".**

- **Real Time—A system that must respond within a *reasonable* amount of time where reasonable is certainly up for discussion. Converting text to speech is an example.**
- **Hard Real Time—A system that must either respond at specific times or within a specific amount of time. Controlling a robot would be a classic example.**

Note that both turnkey and embedded systems can have no real time requirements, real time or hard real time depending on where they are used.

APPROACHES TO HARD REAL TIME

If the required response time is less than the longest amount of time interrupts are disabled then something other than Linux will have to be responsible for handling the event.

- **Using a PLC (see *Linux Journal*, Jan, 1997)**
- **Intelligence in the interface card (SSC's RIP is an example)**
- **Sliding a hard real-time kernel under Linux (see *Linux Journal*, Feb, 1997)**

EMBEDDED APPLICATIONS

- **Test Equipment (e.g., network testing)**
- **SCADA—monitoring for Virginia Power (*Linux Journal*, Jan, Feb, 1995)**
- **Internet Router or Firewall (*Linux Journal*, Apr, Jul, Oct, 1996)**
- **Process Control**

TURNKEY APPLICATIONS

- **Web Server**
- **ISP System**
- **Mail/File/Print Server**
  + **Workgroup Server from InfoMagic**
  + **Post.Office from InfoMagic**

EMBEDDED OR TURNKEY APPLICATIONS

These applications could easily fit into either category.

- **Point of Sale (POS) System**
  - **Schlumberger**
  - **Standard 486 PC**
- **RIP (Raster Image Processor)**
  - **SSC Project/future product**
  - **Turnkey for us**
- **Information Kiosk**
  - **Web browser**
  - **Possibly touch screen**
  - **Remote update**

SUMMARY/CONCLUSION

What Linux Does

- **Offers reliable multi-user system**
- **Commercially viable**
- **Can be coaxed into real-time**
- **Useful for embedded and turnkey applications**
- **Is very cost-effective**