The Design and Practice of Embedded Linux File System

YongChong Wu

JiangXi University of Technology

Keywords: Embedded Linux; LntelDBPXA250; MTD

Abstract. The embedded file system is an important part of embedded operating system. Due to the particularity of embedded devices storage medium, the instability of power supply and the security of mobile storage, the embedded file system has increasingly obvious importance in the embedded operating system. Based on the analysis of the realization plan of traditional Li ~ file system and several current popular embedded file systems, the paper proposed the realization plan of embedded Linux file system. The paper took the IntelDBPXA250 experiment and development platform as the hardware platform. The main part of the paper described the realization of one embedded Linux operating system on the hardware platform, and laid the background for later file system realization. Then, this paper analyzed the design logic of the three levels, VFS, JFFSZ and MTD, and proposed the overall design frame of embedded file system. Afterwards, the paper described the practice details of Flash file system based on MTD equipment in embedded Linux. Considering the limitation of embedded devices storage space, on the basis of the embedded Linux file system, the paper simplified the corresponding virtual file systems and physical file systems that construct the file system, and made them more satisfactory to the requirement of embedded equipment.

Introduction

With the expansion of application fields and function requirement enhancement for embedded equipments, the concept of embedded operating system was introduced. As the important part of operating system, file system is gradually taken seriously in the development of embedded operating system. Due to the particularity of embedded devices storage medium, the instability of power supply and the security of mobile storage, the design and implementation of embedded file were set higher requirements.

The design and practice of embedded Linux. As is known to all, the desktop operating system is monopolized by Microsoft, but the embedded operating system is contended by a hundred schools of thought. Wind River was once the companies that dominated the field in the United States, and its VxWorks was a monumental work in people's mind. Nowadays, the commodity products such as WindowsCE of Microsoft and Palm05 of Palm occupy most of the market share in the field of embedded operating system of handheld devices.

On the aspect of Windows CE, the operating system of Windows Powered Pocket PC released by Microsoft was designed to better satisfy the mobile computing needs of business people and enterprise users through rich high-end functionalities such as wireless networking, multimedia, and mobile office. The frame has relatively higher demand for PDA device performance, which almost has to be the low-end PC7. On Palm05, the handheld computer adopting Palm055 came into the market in the autumn of 2002. The new features of Palm055 include security, multimedia, wireless networking. From this researchers can see that, Palm05 is gradually drawing close to Windows CE, which makes the pocket much heavier.

The above mentioned two embedded operating systems have their own unique characteristics with the support of large companies, which also lead to hardly any expandability in the secondary development. Due to market competition, the two companies are not able to be completely open to lots of underlying technical details. For the end user or application software developers, Windows CE and Palm05 have been almost perfect, and are in constant development. For the researchers who develop independently the embedded system, the route is not the final solution after all.

Since the birth of Linux, adhering to the spirit of open source code, the professionals of software development and programming enthusiasts have more room for mutual learning and communication, of which the development of embedded Linux becomes an important part in the field of embedded operating system development.

First of all, Linux can be the embedded operating system. As is known to all, Linux is a commonly used operating system, and is most suitable in the area of server. The large and comprehensive design philosophy and embedded system is inadaptable to the requirements of software simplification and high-efficiency. However, the problem can be solved after careful analysis on the core structure of Linux.

Though the Linux kernel is a single-core structure, it introduces the mechanism of module, which can be cut down, and dynamically load and unload. This makes the Linux kernel to some extent have the characteristics of the hierarchy and modularity. Actually, in the Linux core, the occupied storage space of modules such as process scheduling, memory management, interrupt and timer management is small, and most of the space is occupied by driver; most of the device drivers are realized by modules mode, which provide condition for cutting down Linux core and adapting to embedded system. In addition, VFS can be the module being cut down. Outside the module mechanism, Linux core can also be relocated and cut down on the level of source code, and continue to reduce the volume of iL~. To sum up, when applied into the embedded system, Linux can fully meet the requirements of embedded system through appropriate cut down and configuration.

Secondly, Linux can not only be the embedded operating system, but also has many own advantages compared with other embedded operating system.

The Source Code of Linux Is Open and the Cost of the Developed Embedded Products Is Low.

The development model of Linux open source code has been widely recognized, and most of vendor codes of embedded Linux adopt the technical ideas of open source. Not only the Linux itself is open to source code, but also most of the system software, application software and drivers based on Linux are open to source code; these are the resources can be used in the development of embedded systems.

Linux Have Excellent Development Tool

The key of original development of the embedded system is the good development and debugging tools. However, this kind of tool is expensive and not convenient to debug. Linux compiles program with C compiler of GNU project, and debugs program with gdb source program level debugger. They all provide appropriate methods for developers to develop embedded Linux system. Serial interface and gdb communications can achieve the debugging of C source program level and even download of programs into RAM or IFahs.

Linux Has Build-in Network Support, and Achieves High-efficiency Network Communication

From the perspective of simplicity, Linux achieves a relatively complete network protocols through the design of network protocol stack. Linux supports the protocols, such as DHCP, and provides access to other files inside machines on the network (NFS, network file system). Moreover, Linux provides the communication protocol supporting serial and parallel interface, which makes the connection of serial and parallel interface possible.

The Kernel of Linux is Robust and the Operation is Stable

Robustness, reliability and stability of the Linux itself allow of no doubt; embedded iLunx is developed from Linux, and inherits the features of Linux. The requirement of robustness and stability is much higher than general system, so the choose of Linux as embedded operating system laid the foundation for improving the robustness and stability of the embedded system.

Linux Supports Varied Processors

Linux has been successfully transplanted into a variety of processor architectures, such as StrongARM, MIPS, PowerPC, Motorola68K, etc. The success on desktop platform can make us see the prospects of Linux in embedded system.

Finally, the embedded Linux usually contains three parts: the boot loader program, the Linux kernel, and the graphical user interface. The boot loader program is usually the first piece of code executed on any hardware that is used to initialize various parameters and environment variables of CPU and then jump to the beginning of the kernel. Embedded Linux kernel achieves the functions such as process scheduling, memory management, file management and equipment management. Graphical user interface provides the user a visual and friendly operation interface [9].

Generally, special files have nothing to do with external devices, and the involved media are memory and CPU itself. There are inode data structure and denytr data structure on the memory. A series of files under P/roc of Linux belong to this special file[’“]. In short, the hierarchy of Linux file system can be expressed in Fig. 1.

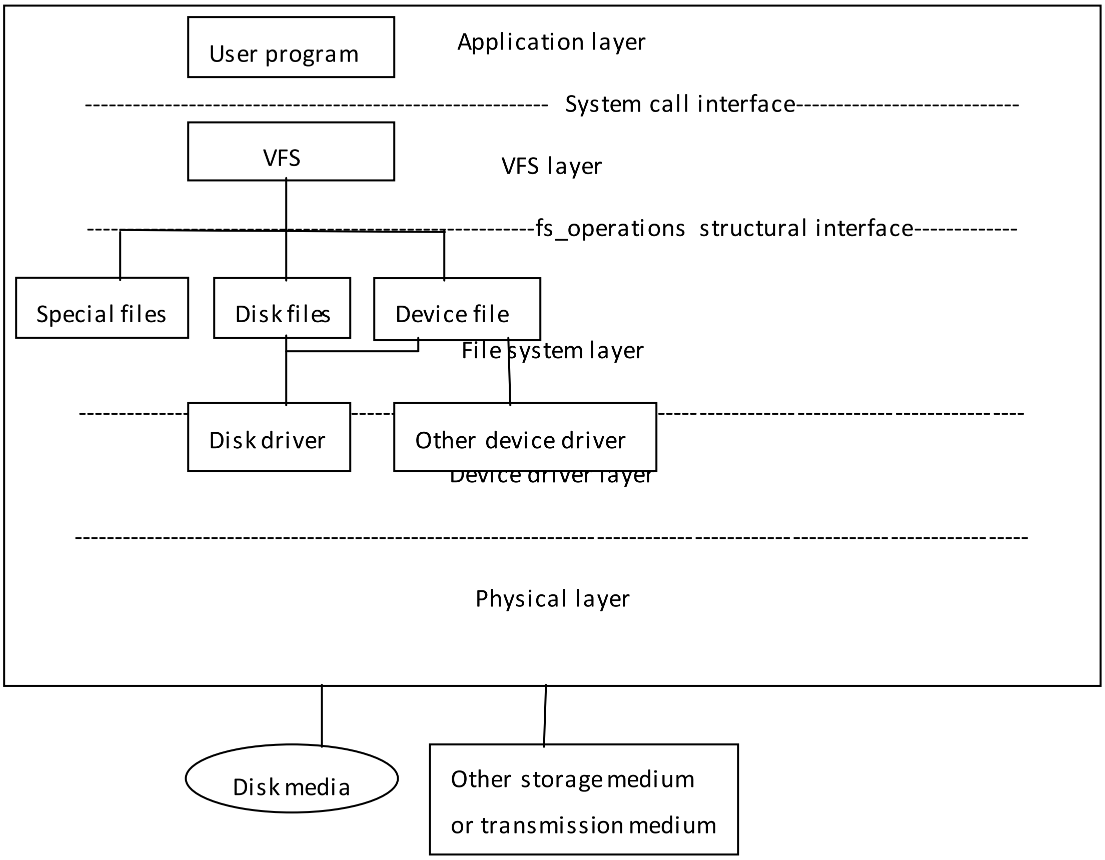

Fig. 1 Hierarchical Structure of Linux File System

At present, there are many solutions to the embedded file system. Due to the wide range of embedded devices, the corresponding files system realization aims at varied devices such as read-only memory, read/write memory, disk devices and lFash memory. As the paper mainly discussed the file system on IFash storage, several popular embedded file system realization schemes were introduced with limited space about the IFash equipment.

DOC (DiskonChip) is a special kind of Flash devices. The 48K stores a piece of firewall codes from the DOC chip physical address O, and the firewall codes will be called in and stored in memory when the BIOS self-tests; the program will monitor the read/write of DOC. It can assist to simulate traditional disk block equipment on DOC, but the common IFash can't be used like disk. The common DOC driver only considers how to interact with NFLT (NANDlFashrTansliatonaLyer), and it has certain difficulties to achieve FJFs on DoC without considerations of support to FJFs.

Based on DOC device, firstly, the realization of FJFS needs to modify the driver of DOC so as to satisfy the read/write requirements of equipment for JFFS. Secondly, due to the existence of that piece of firewall on DOC, it's necessary to realize one special storage program in order to operate FJFS on Doc. The current solution researchers are seeing is to achieve several MTD logical equipments on the same storage equipment mainly assisted by MTD level. 01[

The defect of implementation of the scheme is the difficulty of realization, which involves several layers such as DOC, NFTL, MTD and FJFS; the advantage of the scheme is that the performance of the whole file system can be guaranteed, especially the outage prevention performance is better than traditional file system. The scheme has been widely used in the embedded products of Red Flag Company.

On optimization and compatibility, researchers refer to the implementation of uClibe. Researchers try to give an interface that is compatible with Glibc, which makes the applications and systems command compiled with Glibc can be easily compiled under the plibc. The optimization work in progress is as follows:

1. In Glibc, malloc(0) returns one effective pointer and points to one address. However, in plibc, malloc returns NULL, which has similar distinction. Researchers believe this would be safer.

2. IPibc provides no database library.

3. plibc doesn't support NewworkSecurity Services(NSS), plibc only supports simple password file, shadow password file and storage certification information.

4. Mathematical function library of plibc only support those longdouble with the form of inlines, so this kind of support is limited.

5. The data type directly used by Plibc defines most of the invisible data type.

6. Time zone and complete regional information directory tree are not supported.

7. The BUFSIZE of function printf is 256B. The standard input/output stream does not possess internal buffer auto-dimensioning function. Actually, common standard input/output code is to reduce the complexity and performance so as to get in return the small dimension to the greatest

extent. There are still more to be done for the optimization of plibc. The above mentioned are what researchers considered.

With the development of wireless applications, distributed file systems and file systems in offline conditions also become the problems to be considered for embedded file system. How one kind of file system on the storage medium, with the support of network, can share the resources of other equipments on the internet? On the embedded equipments, what kind of distributed file systems can provide users and system interface and distributed file service? How to solve the synchronization of file and information on embedded equipments? These problems are put forward in order to contribute to the research on embedded file system. Finally, according to the characteristics of the embedded system, researchers think the embedded file system will also have a great development in the following aspects:

1. Compression and storage of data. Although FJFSZ has solved the compression problems, compression and storage technology have great research value due to the limited memory device on embedded equipment.

2. Embedded database. With the extension of embedded system application field, the ubiquitous database technology will be required, and this is based on file system.

3. Distributed file system. One of the major characteristics of embedded system is to provide network support for file system. The advantages of distributed file system include network transparency, location transparency, user (node) flexibility and expansibility, which are very suitable for the use of embedded equipment.

4. File system based on wireless application. Nowadays, the mobile handheld devices are the mainstream; file system under wireless application will be the research priorities of embedded file system, and become the offline file system with further extension.

Conclusions

By the double patching of ARM and Xscale architecture, researchers successfully realized the embedded Linux operating system. The practical experience is summarized from predecessors, for example, the patching programs of ARM architecture come from related projects of Open Source organization; there are also the practical gains, for example, the patching programs of Xscale architecture, especially the experiment board. On the embedded system, researchers successfully transplant embedded Linux file system. In the embedded file system, the support and drive of MTD device are achieved. The experience of compiling MTD equipment driver laid good foundation for changing different kinds of experiment board or storage medium with different performance. In this file system, the introduction of VFS support greatly facilitates the dynamic load of various physical file systems in the future, which is not common in the embedded file systems with the same type, and the bold attempt can provide certain reference for the similar research in the future. In particular, as a result of the limitation of the embedded system storage space, this paper discussed the reduction procedure on VFS and one physical file system (FJFZS) in a whole chapter. The simplification and change of VFS source code level reduces the system storage space in one aspect, and simplifies the logic realization of embedded system. System function calls involve hundreds of functions; call relationship is complex, and simplification work on source code level requires detailed analysis and continuous debugging. JFFSZ simplified scheme is basically achieved; establish JFFSZ file system that provides simplest function and ensure the functions are perfect. Researchers believe that the simplified scheme is feasible and necessary on the embedded devices.

References

[1] Lennon A. Embedding linux[J]. IEE Review, 2001, 47(3): 33-37.

[2] Xian W. The Design and Implementation of Embedded Linux File System [J][J]. Computer Engineering and Applications, 2005, 9.

[3] Sailer R, Zhang X, Jaeger T, et al. Design and Implementation of a TCG-based Integrity Measurement Architecture[C]//USENIX Security Symposium. 2004, 13: 223-238.

[4] Wu P L, Chang Y H, Kuo T W. A file-system-aware FTL design for flash-memory storage systems[C]//Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09. IEEE, 2009: 393-398.

[5] Junyan L, Shiguo X, Yijie L. Application research of embedded database SQLite[C]//Information Technology and Applications, 2009. IFITA'09. International Forum on. IEEE, 2009, 2: 539-543.

[6] Nooshabadi S, Garside J. Modernization of teaching in embedded systems design-an international collaborative project[J]. Education, IEEE Transactions on, 2006, 49(2): 254-262.

[7] Shao Z, Zhuge Q, He Y, et al. Defending embedded systems against buffer overflow via hardware/software[C]//Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, 2003: 352-361.

[8] Abeni L, Goel A, Krasic C, et al. A measurement-based analysis of the real-time performance of linux[C]//Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE. IEEE, 2002: 133-142.

[9] Yanpeng S, Peng P, Yuan Z. Linux transplantation based on the processor S3C2440[C]//Electronic Measurement & Instruments, 2009. ICEMI'09. 9th International Conference on. IEEE, 2009: 2-306-2-309.

[10] Hamblen J O. Using a low-cost SoC computer and a commercial RTOS in an embedded systems design course[J]. Education, IEEE Transactions on, 2008, 51(3): 356-363.