Exploring the use of Linux distributions in embedded platforms

Hanjie Zhang[1,2*], *Tian Li*[3]
1. Anshun University School of Mathematics and Computer Science; Anshun,
Guizhou, 561000
2.Guizhou University of Finance and economics; Guiyang, Guizhou, 550026
University of Sao Paulo, Philippines;
3.Guizhou Yihualu Dta Assets Research Institute Co. LTD.550002

Abstract: Linux is one of the most widely used operating systems in embedded systems development, with a very friendly development environment and the flexibility to meet the needs of different levels and sizes of enterprises. This paper describes and analyses the use of Linux distributions in embedded systems and outlines its own construction and technical classification as a way to promote the performance of both.

Keywords: Linux; embedded platform; ARM

Introduction

Embedded systems are increasingly used in industrial, military and aerospace applications, and as embedded devices move towards miniaturisation and multifunctionality, the best balance between power consumption and size will be a problem to be solved. Linux is a free open source system based on the Motorola Linux kernel, mainly used in personal computers and mobile communication devices. This paper therefore explores it.

1. The relationship between embedded platforms and Linux distributions

With the continuous development of science and technology, in the industrial, military and aerospace applications are becoming more and more widespread, embedded systems as one of these devices will play a greater role in the future. Embedded system refers to a microcomputer system that combines hardware and software to meet user needs, has strong autonomy, can perform various tasks independently and can accomplish tasks that cannot be done by ordinary machines. The embedded operating system, as the core of the application software, is the brain of the embedded system. The programming language or platform that plays a supporting role for the running environment provided by the application software of a computer hardware, including the management hardware and software, so that it can run safely and securely in it. With the development of embedded technology, embedded operating systems have become an important part of the landscape. The mainstream of these is the open source Linux version represented by Ubuntu, which is ideal for use on the Android platform as it is written in a pure Java language and can therefore be ported for use with all the latest Java versions such as JDK 7 and JDK 8 support. Embedded systems usually consist of a processor, memory, input and output interface circuits and other peripheral devices. Linux is an operating system based on Motorola's free and open source protocol and is an embedded system platform with rich kernel resources and powerful programming capabilities, written in C kernel function library, which has good portability and can run on a wide range of platforms. Thanks to its openness and stability, Linux can play an important role in many areas, such as the embedded systems sector. Linux is a completely free and free-to-use software, and anyone can download and use the operating system on the web. From a software engineering point of view, Linux is portable and it can implement almost all computer functions; and from a system level, Linux is more stable and secure. From a performance point of view, Linux is a very good operating system with great stability, security and scalability, so the application of Linux in the embedded field is a necessity. From a functional point of view: Linux has a wealth of features and powerful applications.

2.Linux distributions in embedded platforms

2.1 Kernel and porting

Linux is a free and open source operating system with its own kernel and the ability to make changes to the system without interrupting the current process. Linux is highly compatible and can work seamlessly with various operating systems such as Windows and Linux. The mainstream kernels used in embedded systems today are Ubuntu, FreeBSD and Debian, and the operating system is the most important component of the operating system. The operating system is the most critical and core part of the entire system development and operating environment, which mainly includes: operating system management software: Linux kernel (such as KB2048384, etc.), user packages, data objects; operating system kernel version: Linux kernel (mainly including RHEL and Linux3 series kernel), operating system-related device drivers: including hardware and software drivers (Linux driver source code), etc.; Linux kernel management, kernel tools, process/thread management and service composition. The kernel management provides address space and other information for all processes within the system, such as process names and version numbers. Kernel tools are mainly used to create user space for tasks, create or modify class objects, registry objects and create processes. Process/thread management, on the other hand, refers to the control of processes and threads and thus the allocation of CPU resources. For tasks, the multiplexing function is usually achieved through calls to threads, i.e. through multiple processors that can work simultaneously but each work independently to provide various services to the system at startup or runtime.

Linux porting includes porting of the kernel, distribution and drivers. As embedded products are generally designed in a relatively short time frame, only simple porting needs to be done to meet the requirements on the hardware platform. In order to reduce the time for software development, many users use Linux systems to write their programs. The kernel is the core part of the embedded system and if used directly without compilation, it can lead to confusing programs, so the kernel should be compiled and packaged into the kernel space first, and then the program should be written over in C++. (Here are some details to keep in mind when writing in C++. Optimisation of C/C++ after compilation can result in much better performance, usually using a mix of optimisations). Install development tools such as system drivers and function libraries. After the software has been developed, the code is debugged and optimised to improve stability, reliability, portability and other aspects of performance; finally, the software is compiled, installed and released to the hardware platform.

2.2 Development platform

With the development of technology, the performance of embedded devices is constantly improving, which puts higher demands on software development techniques. How to develop high performance, good performance and portable embedded system software has become an urgent problem for software developers to solve. In order to solve these problems, many companies have invested heavily in the development of embedded system software and hardware products with high cost performance. This article introduces an embedded product for industrial control as an example: an embedded system can be easily set up by the user via a PC, and the user can run the system on the PC to realise different applications: the keyboard and mouse can be used to control the work of the device and switch between different control methods, etc. The most widely used software in the field of industrial control is the Industrial Control System (ICS). Its main functions are the acquisition, processing and storage of measurement data, the display of status parameters and the processing and analysis of data, as well as the programming and control of devices via keyboard and mouse. The solution consists of two parts: the first part is the real-time operating system (Real-time), which mainly includes RTL libraries, real-time communication protocols, network communication protocols, etc.; the second part is the embedded software system. Real-time operating system is the most used core software in the field of embedded system development at present. It adopts the C / S model structure, the operating system is divided into the bottom driver (Host) and the upper layer of the application (Application), which is divided into data acquisition and processing of two blocks of content. Currently, embedded systems are used in a wide range of applications[1] . Linux systems have considerable advantages over traditional Windows systems. For example, Linux is a very open and free operating system. Also due to its very good performance, portability and stability when dealing with large numbers of programs. Due to the openness and stability of the Linux system, it can play an important role in many areas, such as the field of embedded systems. Linux is a completely free and liberal software and anyone can download and use the operating system on the internet.

2.3 ARM systems

ARM-based Linux distributions generally use the Cortex-A8 processor, which is ARM's kernel and offers much higher performance than the Cortex-M4 processor, and also ensures system stability and security. In order to ensure the speed of system programs, an interrupt management module is added to the kernel to ensure real-time system performance when running under the Linux operating system[2] . At the same time, in order to make the best use of the performance of Linux, the device control and monitoring functions can be implemented by calling the Copy function during development. Specifically, by parsing the data from the embedded Linux port to the Copy function, the corresponding software program can be written. In order to increase the speed of the system, an interrupt management module has been added to the ARM. This module mainly includes several functional modules such as interrupt management, device configuration management and interrupt management, which is not new to embedded developers.

Conclusion

With the development of embedded products towards miniaturisation and multi-functionality, Linux has become increasingly accepted as a free open source software. Linux-based embedded systems are a good development platform with rich and practical open source features. This paper analyses the application of the Motorola Linux kernel in embedded platforms to provide a reference for its application in embedded platforms.

Reference.

[1] Bei Yangfei, Gao Chao on Linux for embedded operating systems [J]. Fujian Computer, 2020,(2):4-5.

[2] Xu Shanfeng, Ma Liling, Wang Junzheng, et al. Porting method of ARMLinux based on AT91RM9200 [J], Microcomputer Information, 2020, 08(14):183-184 .