

Research on high speed data distribution technology based on kernel state of domestic operating system

Shumin Geng, Wei Wang, Xinlei Yang, Peng Du and Hui Qiao*

Beijing Space Information Relay and Transmission Technology Research Center, China

Keywords: Relay satellite system, High speed data distribution, Domestic operating system, Kernel state.

Abstract. The data distribution platform is the ground information center of the relay satellite system to realize the data relay ability. With the increasing demand of the user data transmission, the data transmission ability of the relay satellite system is increasing. The increasing data transmission ability of the satellite puts forward higher requirements for the data distribution ability of the ground distribution platform. In this paper, the technology of data acquisition and data exchange based on the kernel state of the domestic operating system is studied. The design method of data distribution platform based on the kernel state programming of the domestic operating system is proposed. The simulation test environment is built to verify the effectiveness of the method, which can greatly improve the data processing ability of a single data distribution device, significantly reduce the scale of data distribution device, it provides strong technical support for building a new model of intensive and efficient data distribution.

1 Introduction

With the rapid development of the relay satellite system, the application scope of the system is expanding. At present, it has provided space-based measurement and control and data relay services for dozens of tasks, including remote sensing satellites, launch vehicles and so on.

The data distribution platform is the ground information center of the relay satellite system to realize the data relay capability, which is responsible for the network transmission of forward and backward service data between the data transmission terminal of the ground station and the user center. The traditional data distribution platform is implemented by user mode programming. Restricted by the scheduling overhead of frequent switching between user mode and kernel mode and the resource overhead of multiple copies of data processing, the high-speed data distribution and processing capacity of a single server can only be

* Corresponding author: yx1199207@126.com

achieved at 600Mbps. With the increasing demand for high-speed data transmission from users, the data relay capacity of the relay satellite system is increasing. In the future, the downlink data of a single satellite can reach 4*600Mbps, and the high-speed data of a single channel on the ground can reach 1.2Gbps. The data distribution capacity of a single device under the current user state programming mode cannot meet the growing demand for high-speed data transmission from users. Therefore, to improve the high-speed data distribution ability of data distribution platform, reduce the scale of system data distribution equipment, optimize the level of system construction investment and management, and improve the data distribution ability has become an urgent demand for the development of data distribution platform.

2 Analysis of kernel programming technology

Based on the consideration of concurrent service and access security, Linux operating system designs two levels of user state and kernel state on the hardware device. The application program runs in user state, and system call, kernel and driver constitute the kernel state of the system. Among them, Linux kernel is the core of operating system. Generally, a kernel is composed of interrupt service program which is responsible for responding to interrupt, scheduler which is responsible for managing multiple processes to share processor time, memory management program which is responsible for managing process address space, system service program, and so on. System call interface provides a medium for user program to access computing and IO resources in kernel space, and the device driver is the medium for kernel process to access hardware. The relationship between user state, kernel state and hardware in Linux operating system is shown in Figure 1.

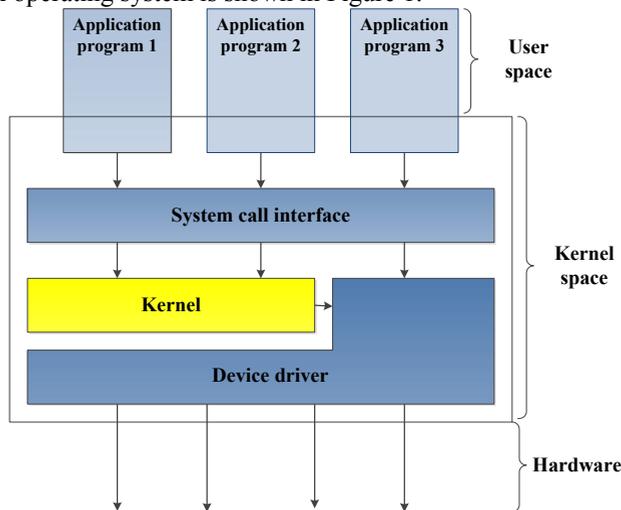


Fig. 1. Relationship between user state, kernel state and hardware.

Firstly, the kernel program has the protected memory space and all the permissions to access the hardware device, while the application program runs in the user space, they can only access part of the system resources that they are allowed to use, and they need to access the hardware indirectly through the system call interface, so the real-time performance of the application program is weaker than that of the kernel program; Secondly, the Linux operating system is preemptive in a multitasking operating system, the priority of the kernel is higher than that of the application program. In case of resource competition, the kernel program can

block the user state application program to obtain the permission to use the resource, so as to ensure the timeliness of the kernel program, and also reduce the timeliness of the user state application program. Thirdly, because the memory space is independent of each other, the data to be processed by the user program must be in the kernel state and The user state copies back and forth, which takes up a lot of CPU resources, while the kernel state programs can reduce the number of memory copies and improve the processing performance without the limitation of memory space partition. Fourthly, the user programs that refer to system calls need to switch between the user state and kernel state frequently during the execution process, which requires the CPU to record the context repeatedly after recovery, the kernel scheduling resources will also be occupied, but the kernel state program will not have the above running state conversion overhead. To sum up, kernel state programs have higher response time and execution efficiency and lower system overhead than user state programs. Compiling user state applications into kernel state is an effective way to improve business processing performance.

At present, the kernel programming technology at home and abroad mainly includes two ways: one is called system integration, which is to modify the source code of the operating system and integrate the program into the operating system. The system integration way requires to have the source code of the operating system; the other is called module assembly, which is to compile the program by using the standard interface reserved by the kernel or hook technology Kernel module. The difference between the two methods is that the execution efficiency of the program integrated into the operating system is higher, but the impact area of the kernel change code needs to be comprehensively evaluated, and the other service support functions of the kernel need to be fully tested, so the technical risk is large. There is some memory loss in compiling the kernel module, but the kernel module can be dynamically loaded and unloaded, so the flexibility is better, it can realize the dynamic expansion of kernel functions and is widely used.

The development and maturity of foreign Linux operating system kernel is the real reflection of the first kernel programming. Taking TCP/IP protocol stack as an example, such as Telnet, FTP and other application layer protocol families are constantly improved into the kernel code, which is an example of continuous source code modification and function integration. The development process of the network front-end software of the first generation relay satellite system also adopts the system integration mode. With the help of the early Linux open source code, the logic code of data distribution business is added to the system kernel vmlinux. At the same time, the unnecessary functions of the operating system kernel are reduced, and the high-speed multicast data distribution ability of 2*300Mbps is realized.

3 Design of data distribution platform based on kernel state of domestic operating system

With the popularization of equipment localization, the second generation relay satellite system data distribution platform has generally adopted the localized Linux operating system. However, because the domestic operating systems are all distribution versions and do not provide external source code, the kernel programming mode of system integration adopted by the early network front-end computer is no longer suitable for the use of the existing system. At the same time, the kernel programming based on system integration results in the system software and hardware need to be customized, which does not meet the use requirements of device generalization. Therefore, in the design of system software architecture, the requirements of equipment generalization and software modularization are fully considered, and the kernel state programming mode of domestic operating system module assembly is adopted to solve the technical bottleneck problem of limited user state performance.

3.1 Data distribution software architecture

The test results of data receiving in kernel state, data sending in kernel state, data receiving in user state, data sending in user state, data exchange between kernel state and user state, and processing ability of kernel acceleration module show that the performance bottleneck of data distribution platform in user state programming lies in data receiving, and kernel state has obvious advantages over user state in data receiving performance. Therefore, in the design of the system software architecture, the network acceleration requirements of data acquisition are fully considered, and the kernel "hook" mode is used to capture data packets, which is used to accelerate the data acquisition process and improve the high-speed data distribution performance.

The software architecture consists of system network acceleration module, system receiving and forwarding module, task management module and system monitoring and management module. Among them, the system receiving and forwarding module, task management module and system monitoring and management module work in the user state, and the network acceleration module works in the kernel state. According to the Protocol, IP and Port, the data packets are stored in the corresponding kernel buffer, and then the data is moved from the kernel state by the Unix Domain Socket network socket mechanism through the data exchange mode of kernel state and user state to user status, user status reads data packets for data processing and data transmission. The functions of each module are as follows:

- 1) The network acceleration module is used to speed up the high-speed link and obtain data quickly;
- 2) The system receiving and forwarding module provides core business logic, including data collection, data processing, data forwarding, data storage and other core logic;
- 3) Task management module realizes task configuration, distribution equipment scheduling, system timing, dual computer management, log reporting, multi task chain monitoring information processing, multi task status report transmission, information exchange with user center and other functions;
- 4) The system monitoring and management module provides human-computer interface for configuration management, task scheduling, equipment control and other human-computer operations.

3.2 Netfilter check packets

The system network acceleration software module is a driver module under Linux. It is used to accelerate the data link and solve the performance bottleneck in the high-speed data link with the user state interface provided.

The IP protocol stack of Linux is divided into four layers: MAC layer, IP layer, Transport layer and Application layer. Most of the Application layer protocols are in user mode, and the rest are in kernel mode. When the network packets enter the kernel from the driver, they enter the MAC layer, IP layer, Transport layer and Application layer in sequence. The MAC layer is used to check whether the packets are needed by the local computer. The IP layer is used to check the information of the IP layer and judge whether it needs to be processed or forwarded by itself. The Transport layer finds the corresponding socket. If TCP, it maintains the protocol. The MAC layer is used to check whether the packets are needed by the local computer Use layer to process the corresponding data according to the application layer protocol.

Linux operating system provides specific callback function (also called hook function) for kernel program through Netfilter to intercept and process key points of network packet in IP protocol stack. The call to hook includes the following five opportunities:

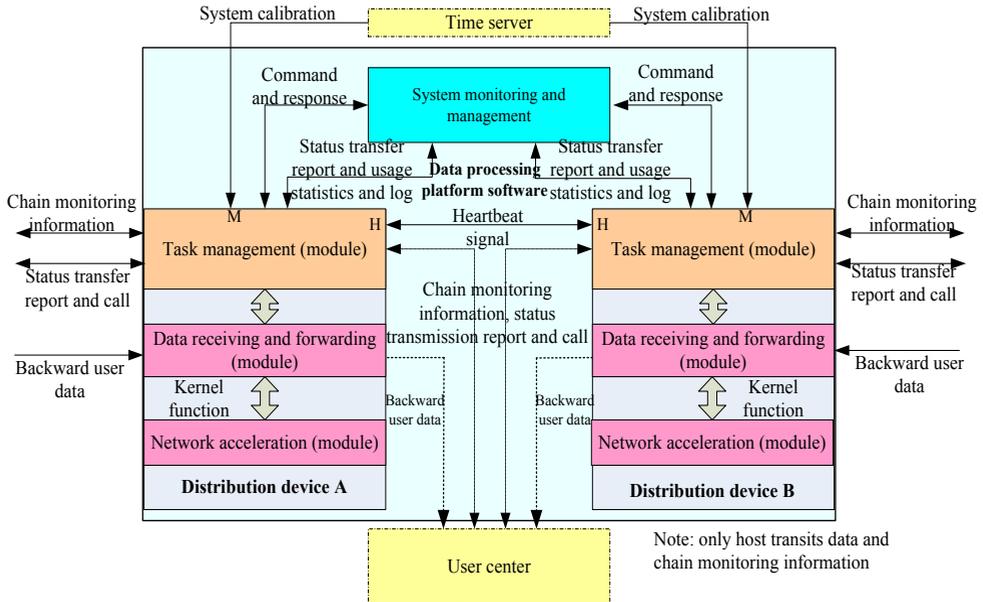


Fig. 2. Schematic diagram of software module composition.

- 1) NF_IP_Pre_Routing, which means after integrity verification and before route selection, that is MAC layer;
- 2) NF_IP_Local_In, which means after the route selection is determined, and the purpose of the data packet is the local host, that is IP layer, at this time, the protocol stack has completed the automatic splicing of large IP data packets;
- 3) NF_IP_forward, indicating that the destination is a packet of another host, which is used to intercept the packet sending process;
- 4) NF_IP_Local_Out, which means that the packets from the local process are in the process of leaving the local host;
- 5) NF_IP_Post_Routing, which means before the packet leaves the local host and "goes online".

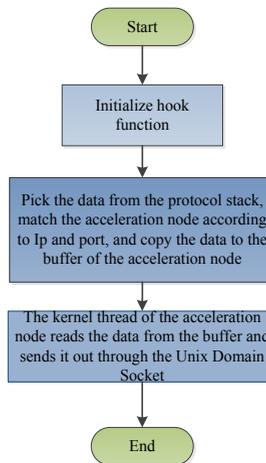


Fig. 3. Acceleration process of system network acceleration software module.

3.3 Kernel thread multi link support

The system network acceleration software is a linux driver module. The management interface of the acceleration link is the file operation interface provided by the driver module. The file operation is the general operation interface of the driver module. The system network acceleration software can realize the corresponding interface in the file operation.

In order to support the network acceleration of multiple links, a data structure of the acceleration list is designed in the system network acceleration software. The acceleration list uses a common one-way list. Each acceleration node in the acceleration list corresponds to a link, and each acceleration node has an independent acceleration thread for data processing, making full use of the performance characteristics of multi-core parallel processing.

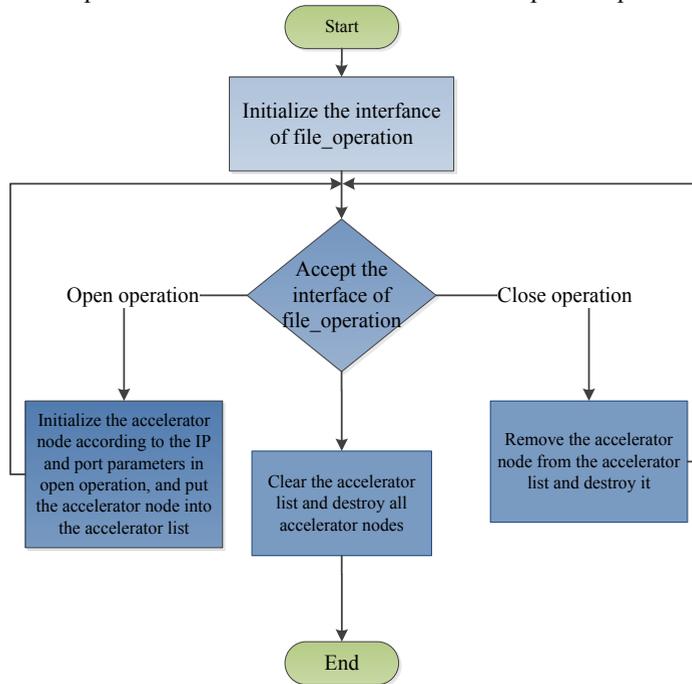


Fig. 4. Acceleration link management of system network acceleration software.

3.4 Unix domain socket for high speed data exchange

Unix Domain Socket, also known as IPC (inter process communication) socket, is used to realize inter process communication on the same host. Socket was originally designed for network communication, but later an IPC mechanism was developed on the framework of socket, namely Unix Domain Socket. Unix Domain Socket does not need to go through the network protocol stack, package and unpack, calculate the check sum, maintain the serial number and response, etc., copies the application layer data from one process to another only. This is because IPC mechanism is essentially reliable communication, while network protocol is designed for unreliable communication.

The flow direction of network packets in the system network acceleration software is shown in the following figure:

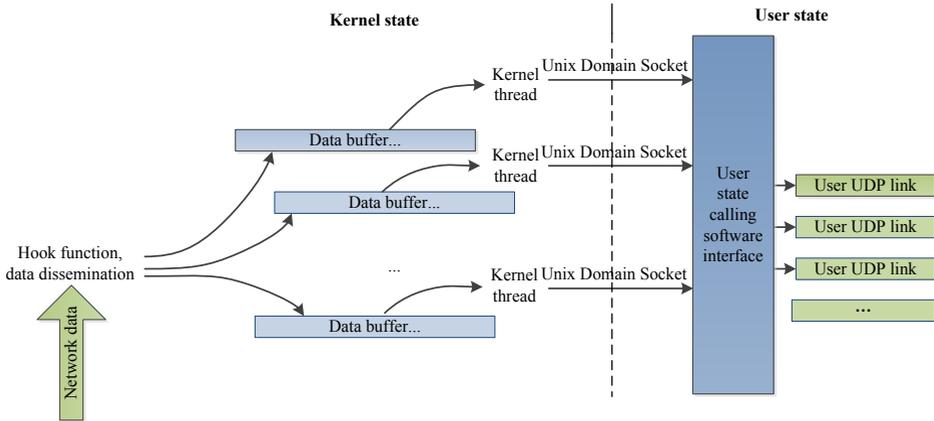


Fig. 5. Data flow of system network acceleration software.

4 Verification of kernel data processing capability based on domestic operating system

In order to verify the effectiveness of the proposed high-speed data distribution scheme based on the kernel state of the domestic operating system, the following test environment is built to test the limit rate of data forwarding back to high-speed data. The test and verification environment is shown in Figure 6, where PC1-controller is the management terminal for controlling management functions and logging in to other PCs through SSH to control the start and stop of programs; PC2-source is the high-speed analog data transmission terminal for simulating data sources; PC3-relay is the data distribution server for simulating data processing platform; PC4-recv is the simulating user for simulating receiving user data; GPFS is the storage device.

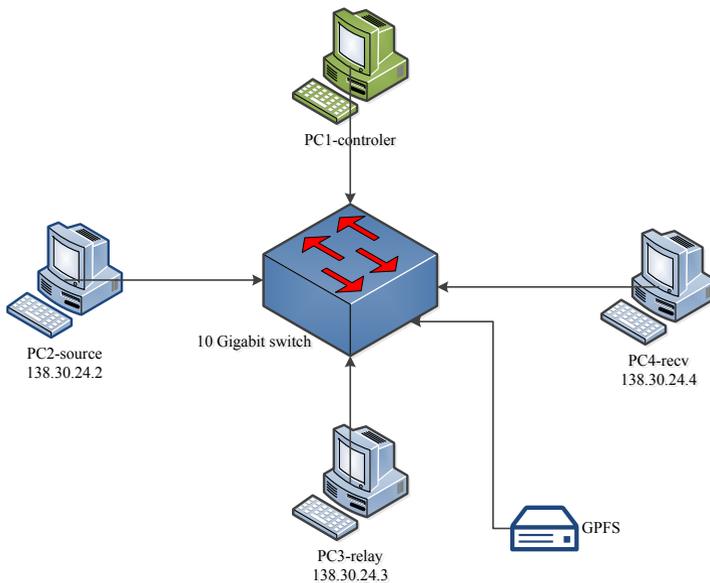


Fig. 6. Verification environment of kernel data processing capability.

The configuration of PC1-controller, PC2-source, PC3-relay and PC4-recv hosts is the same, as follows:

CPU: 3.0GHz&48 cores;

Memory: 64GB;

Operating system: NeoKylin 3.2.2 (carambola);

Network card: One 10 Gigabit network card (network card model: 82599ES, driver version: 5.3.6).

The test is based on the data storage mode of NFS protocol, UDP unicast test is adopted, the packet length is 801 bytes, and the performance test of 1-way and 2-way high-speed data distribution is carried out respectively. The test requires that the device runs stably for more than 3*24 hours without packet loss. The test results are shown in Table 1.

Table 1. Test record of data distribution capability of high speed data distribution platform based on kernel state of domestic operating system.

Test items	Task number	Rate (Mbps)	Program type	Packet and byte records	Result
2-way high-speed data distribution performance test	Channel 1	976	Send out	packets=52398671985	No packet loss
			Forwarding and receiving	packets=52398671985	
			Forwarding and storage	packets=52398671985	
			Forwarding and sending	packets=52398671985	
			User reception	packets=52398671985	
			File size	43228904387625	
	Channel 2	1003	Send out	packets=53141450968	No packet loss
			Forwarding and receiving	packets=53141450968	
			Forwarding and storage	packets=53141450968	
			Forwarding and sending	packets=53141450968	
			User reception	packets=53141450968	
			File size	43841697048600	
1-way high-speed data distribution performance test	Channel 2	1250	Send out	packets=50433760968	No packet loss
			Forwarding and receiving	packets=50433760968	
			Forwarding and storage	packets=50433760968	
			Forwarding and sending	packets=50433760968	
			User reception	packets=50433760968	
			File size	41607852798600	

5 Conclusion

In this paper, the technology of data acquisition and data exchange based on the kernel state of domestic operating system is studied, and a method of data distribution platform based on the kernel state programming of domestic operating system is proposed, which can realize the high-speed data distribution of single way 1250 Mbps or double ways 970 Mbps, and greatly improve the data processing ability of single data distribution equipment. Compared with that of the user state programming, the data distribution ability is doubled, and the equipment scale

can be reduced to one-half. The improvement of equipment capacity can further meet the needs of users with higher data transmission rate for the use of relay satellite system, significantly reduce the scale of data distribution equipment, and provide strong technical support for building a new mode of intensive and efficient data distribution.

References

1. Bao hua S. Detailed explanation of Linux device driver development (Second Edition), People's post and Telecommunications Press, BeiJing, 2010.
2. Yong ming W, Yue G. Linux device driver (Third Edition), China Electric Power Press, 2011.
3. Lin F, An interactive data method between kernel state and user state in windows and its application in PC based Open CNC system, Computer knowledge and technology, 2011(16):24-25
4. Shaogang W , Haifa L , Shen W , et al. Application Technology of Electronic Authentication Based on USBKey in Domestic Operating System. Journal of Information Security Research, 2016.
5. Andrus J, Hof A V, Alduaij N, et al. Cider: native execution of iOS apps on android, 2014.
6. Fei L, Qi H, Katic N, et al. A method for state estimation of distribution network based on the classification of distribution transformer load, 2017.
7. Guanping Xiao, Zheng Zheng, Beibei Yin, Experience Report: Fault Triggers in Linux Operating System: From Evolution Perspective, 2017.