THPM 19.8

Designing Low Cost Embedded System for Multimedia Data Processing: Linux Based Micro Kernel Approach

Sang-Yeob Lee and Youjip Won and Whoi-Yul Kim

Division of Electrical and Computer Engineering, Hanyang University

ABSTRACT

In this article, we introduce the novel technique to design and implement the low cost embedded system for multimedia data processing. We use the skeleton of existing Linux operating system and developed micro-kernel to perform a number of specific tasks for our purpose efficiently and effectively: memory management, video processing, input/output port control.

OVERVIEW

There are a number of reasons to prefer the dedicated purpose electronic home appliances rather than using general purpose personal computer: cost, size, and ease of use[1]. One of the typical activities in accessing and processing the on-line information is about handling the multimedia related information, e.g. voice, still image, and motion picture. While it is possible to perform these activities using general purpose personal computer, there have been intense demand for the system which is specifically designed for handling image, motion picture locally as well as remotely through internet.

In this article, we like to share the details of design and implementation experience of state of art multimedia information processing set-top box. DigiAlbum® is Internet Appliance which enables the user to view, save, and retrieve the still image and motion picture in wide screen as well as to navigate the web efficiently. User can supply the source data through the digital imaging device as well as through Internet. We develop the system which requires minimal amount of hardware resources while satisfying all requirement of multimedia data processing. The biggest challenge is to seamlessly deliver the light weight graphical user interface, file control functionality, memory management and system device control facility with limited amount of system resources. After two year effort, we successfully develop the linux based embedded system which has memory management, file and system device management capability with light weight graphical user interface. Our system is currently embedded in WebTV and machine vision system as well. The purpose of this article is to publicize the technical details of our system including hardware organization, micro-kernel based embedded Linux system[1][2] which is responsible for video memory processing, input/output port control, memory management, and application loading technique.

Hardware Organization

Fig. 1 illustrates the hardware organization. Our system is designed towards x86 architecture compatible and is able to run on any microprocessor on x86 compatible system board. DigiAlbum uses STPC which is x86 compatible embedded microprocessor with video processing capability running at 66Mhz. Video processing capability of STPC unnecessitates the high performance graphic handling function[1]. For faster data and control exchange, PCMCIA and Video Chip is connected to CPU via PCI bus. Other peripheral devices are connected to CPU through ISA bus to reduce the hardware cost of the system. Remote control device sends command signal to system via remote sensor(infrared red) which is connected to system via RS232C port.

Operating system, micro kernel and application software run on 4 MByte of Flash Memory. Generally, 32bit operating system approximately takes up 30 MByte of main memory. Even, Linux kernel takes up about 1 Mbyte of memory space. This is still too heavy for embedded system. We trim out the unnecessary features of Linux kernel and reduce the size into 500 Kbyte. Micro kernel which is responsible for video memory processing, input/output port control, memory management requires additional 300 Kbyte. We were able to fit entire operating system into 800Kbyte of memory space. Rest of the memory space are used for load the applications and its accompanying resources, e.g. font images, background images, etc.

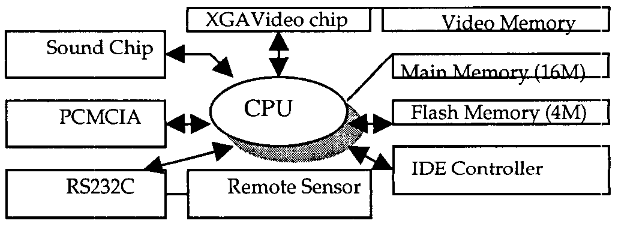

Fig.1 Hardware Architecture

Linux Kernel and Micro kernel

Developing sophisticated embedded operating system itself is serious theme which requires large amount of time and effort. While we developed most of the features of the system using micro kernel, we were able to save substantial amount of resources by using Linux OS as our starting point. Fig.2 illustrates the organization of our embedded operating system.

[1] Recent version of this system is loaded with Pentium 200Mhz with MMX instruction set

We use the existing file system management service from the Linux kernel(FAT32, ext2)[2]. Memory management subsystem of the Linux kernel is responsible for initializing the memory in boot phase, e.g. initializing global/local descriptor table, and the micro-kernel takes care of allocating the memory to the application. Micro-kernel is developed for the following tasks: providing graphical user interface, input/output port control, e.g. PCMCIA, accessing the video chip, and memory management.

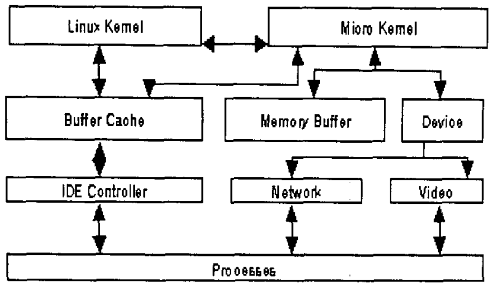

Fig.2 Operating System Organization: Linux Kernel and Micro Kernel

Software Architecture

Our system is designed to run single application at a time and thus, the application program can fully utilize the system resources except for the ones used by OS. Since our system is to be embedded in the various types of devices, e.g. DigiAlbum, Web TV, machine vision system, the embedded system should provide flexible environment to run different applications. For this purpose, we partition the software running environment into two layers: kernel layer and application layer. Application can be downloaded through parallel port, from CD-ROM or network. Fig. 3 illustrates organization of software running environment.

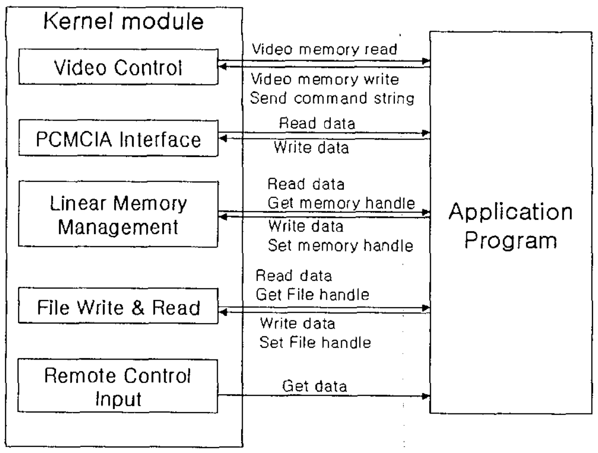

Fig.3 Software Architecture

A. Linear Memory system

Most of the 32 bit general purpose operating system adopts virtual memory system technique and Linux is not an exception. While virtual memory system gives good system performance boost by abstracting primary and secondary storage space into one single large primary memory space, our system with 4 Mbyte of flash memory does not need virtual memory system and henceforth does not have to take the burden of maintaining the swap space, page table, address translation, etc. In realizing the linear memory system, we decided not to use natural linear memory system provided by 32bit protected mode of x86 architecture. The main reason is its programming complexity. Instead, micro-kernel is allocated all free memory blocks after kernel initializes virtual memory system and all the memory management requests from the application program are serviced by micro-kernel. Micro-kernel handling the memory allocation requests based on linear memory map is developed.

B. Video Chip control

In controlling the video chip, we partition the memory subsystem of video chip into two: video output memory and processing memory for video data control. Normally, portion of memory other than for video output remains unused. We exploit the remaining memory section to boost up the speed of decompressing operation. Decompressing the JPEG compressed image is computationally intensive operation, especially for 80486 based 66Mhz STPC. To speed up decompressing process, we use the accelerator operation provided by the video chip. It can perform the 8x8 block arithmetic operation. The advantage of this approach is two fold: 1) we can exploit the 8x8 block arithmetic operation provided by video chip and 2) we can avoid load/store overhead from video chip to main memory. We are able to obtain the reasonable performance increase of decompressing the images in very cost effective manner compared to using the STPC for the same operation. Table 1 illustrates the performance comparison of uncompressing operation by Video CPU, STPC and Pentium-II 200Mhz.

Table 1 Uncompress operation

Resolution	Video CPU	STPC	Pentium II
1024x768	2.3 sec	6.7 sec	0.7
640x480	1.6 sec	4.1 sec	0.5

Concluding Remarks

After a couple of year's intense effort, we successfully developed Linux kernel based embedded system for handling digital images, motion pictures, navigating the web, etc. which requires minimum amount of hardware resources while providing all features required features. Our novel approach enables the end user to enjoy saving, editing, retrieving the multimedia information in more cost effective manner.

Reference

[1] G. Gogniat, M. Auguin, L. Bianco ,"*A Codesign Back-End Approach for Embedded System Design*," *ACM Trans. On Design Automation of Electronic System*, Vol. 5, No. 3, pp.492-509,July 2000

[2] S. Bahadur, V. Kalyanakrishnan, J. Westall ,"An Empirical Study of the Effects of Careful Page Placement in Linux," *Proceeding of the 36th annual conference*, pp 241-250, 1998