RESEARCH ARTICLE OPEN ACCESS

Advantages of Linux kernel porting in Embedded Applications

[1]R Prasanna kumar, [2]D Sowjanya reddy
[1]PG Scholar [2]Assoc.Professor
Dept of ECE, CMR College of Engineering & Technology, Hyderabad,Telangana-India.
1Email: prasanna0053@gmail.com

Abstract:
Operating systems based on linux kernel are used in embedded systems such as consumer electronics (i.e. set top boxes, Smart TVs, IVI, Networking equipment, spacecraft flight software and medical instruments) in general. Thanks to their nature of versatility, operating systems based on the Linux kernel can be also found in mobile devices that are actually touch screen-based embedded devices such as smart phones and tablets, together with personal digital assistants (PDAs) and portable media players that also include a touch screen.

Keywords : Embedded systems, Linux kernel, Arm9.

I. INTRODUCTION

As embedded systems is gaining increased importance in every aspect due to various applications such as multitasking, multithreading, multiprocessing and the special attributes which are inherited by the embedded technology as reliability, flexibility, low power dissipation, cost, speed, time, software upgradation capability.

Various sectors like biomedical, healthcare, automation, automobile, mobile communication, aeronautics, spacecraft, networking has increased the need of embedded technology. Various architectures are used for embedded systems according to their need such as ARM (Advanced RISC Machine) which is a reduced instruction computer (RISC) instruction set architecture. The combination of ASIC and FPGA in ARM platforms delivers an optimal and flexible solution in terms of cost, speed and time. Linux as an embedded operating system gives an added edge to

the embedded technology as Linux is an open source and has special attributes like multitasking, multiprocessing, multiplatform, multithreading, multiuser sharing, full memory protection. Real time operating systems are derived from the Linux kernel source which is being configured as per the requirements and specific to the application constraint. For the new design of the embedded technology the operating system must be modified accordingly to fit the target embedded board and the operating system thus configured and modified is then ported to the specific targeted board.

II. KERNEL ARCHITECTURE

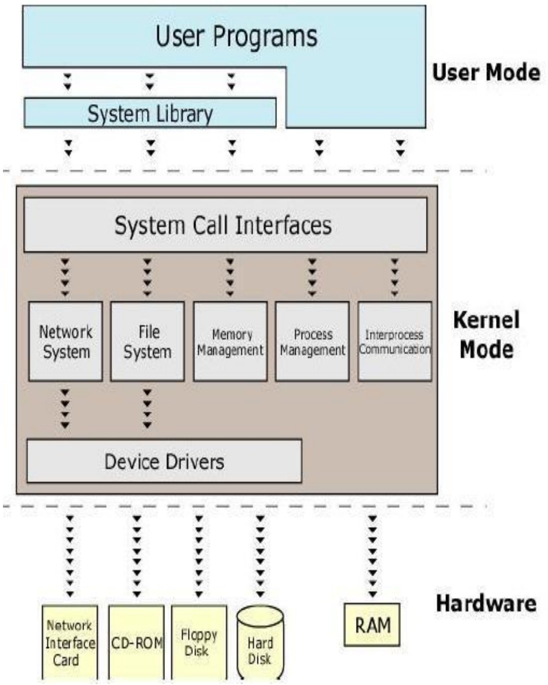

FIG 1 : Linux Kernel Architecture

III. BUILDING LINUX KERNEL

The Linux kernel supports a lot of different CPU architectures. The methods to port the Linux kernel to a new board are therefore very architecture dependent. For example, PowerPC and ARM are very different. PowerPC relies on device trees to describe hardware details whereas ARM relies on source code only. In the source tree, each architecture has its own directory arch/arm for the ARM architecture. For building the linux kernel, cross-compiler should be made accessible on your execution path.

IV. PORTING

In software engineering, porting is the process of adapting software so that an executable program can be created for a computing environment that is different from the one for which it was originally designed (e.g. different CPU, operating system, or third party library). The term is also used when software/hardware is changed to make them usable in different environments.Software is portable when the cost of porting it to a new platform is less than the cost of writing it from scratch. The lower the cost of porting software, relative to its implementation cost, the more portable it is said to be.

V. BOARD SUPPORT PACKAGE(BSP)

A Board Support Package (BSP) provides a standardized interface between hardware and the operating system. BSP provide an interface to device drivers which allow the kernel to communicate with the hardware the Core (ARM), memory, internal and external busses. BSP as the name refers to is an implementation of a software that will control the hardware as per requirement. The piece of software is called a Bootloader or a Boot code which will initialize the Memory along with other peripherals before loading the Kernel (OS). BSP is an interface that implements and supports an Embedded OS on Hardware. It supports the major features sets such as storage, networking, display and multimedia. BSP can be quickly customized for the specific need of the customer. With a BSP, one can rapidly bring up an OS on Standard Development Board and evaluate the features of the OS.

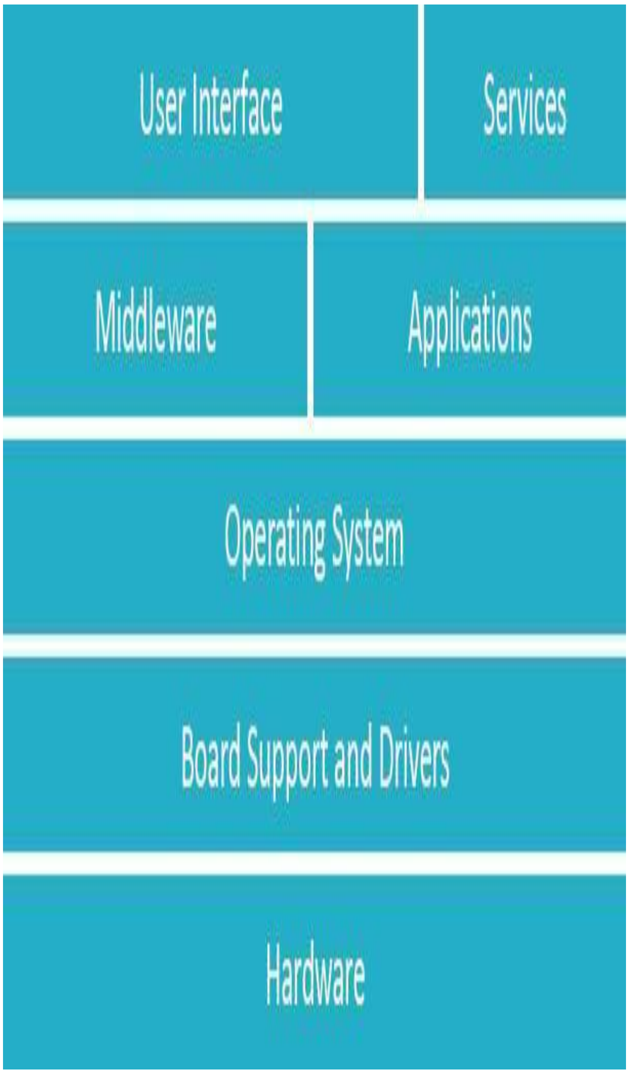

FIG 2: Bsp as an Interface between Operating System and Hardware

VI. BOOT PROCESS

The following flow chart shows the booting process that how boot loaders places the operating system of a computer into a memory. How the initialization of setup clocks as well as the software configuration tools can be done and kernel architecture, linux kernel boot sequence for embedded hardware and debugging linux kernel.

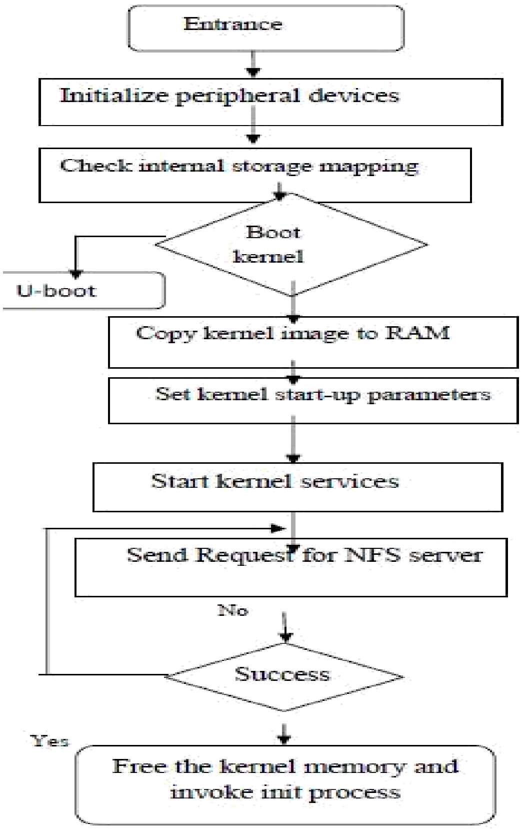
Entrance
Initialize peripheral devices
Check internal storage mapping
Boot kernel
U-boot
Copy kernel image to RAM
Set kernel start-up parameters
Start kernel services
Send Request for NFS server
No
Success
Yes
Free the kernel memory and invoke init process

FIG 3: Flow Chart showing the boot Process.

VII. APPLICATIONS

Few applications in major sectors can be as follows :

- Main stream Smart Phones
- Tablets
- Set top boxes
- Home media players
- Auto infotainment
- Residential Gateway

VIII. ADVANTAGES

- High flexibility with fast synchronized operations.
- It supports dynamic power management and system- on-chip designs.
- ARM9 board supports dynamic design flexibility and portability. Thus it is dependent on C.P.U architectures
- Optimal cost and performance with ARM9 boards.
- It supports a wide range of applications in real time embedded systems.
- ARM 9 supports on-chip buffers in case of linux porting. Thus helps in on-chip debugging .

- ARM9 cores have separate data and address bus signals which the chip manufacturers use in many different ways.
- ARM9 cores include "Enhanced DSP" instructions such as a multiply-accumulate, to support more efficient implementations of digital signal processing algorithms.

IX. CONCLUSIONS

The paper discussed a generic environment build of Linux for ARM core family. This generic build has a limited functionality, as it just enables the kernel to boot and send debug messages through the configured serial port. For a full fledged embedded module his is quit elementary without the support for various peripherals through device drivers which are paramount to the module. Linux kernel though monolithic has an excellent modular approach which enables the driver modules to be attached and detached at run time itself. This very much suits the embedded environment which are always constrained or system memory.

REFERENCES

[1] VincentSanders,“BootingARMLinux”, rev.1.10,June2004.http://www.simtec.co.uk/prod ucts /SWLINU X/files/booting_article.html
[2] http://www.kernel.org.
[3] http://www.kernel.org.
[4] http://www.arm.linux.org.uk/developer/machines.
[5] Web sites dealing with real-time Linux:
- www.rtlinux.org
- www.rtai.org