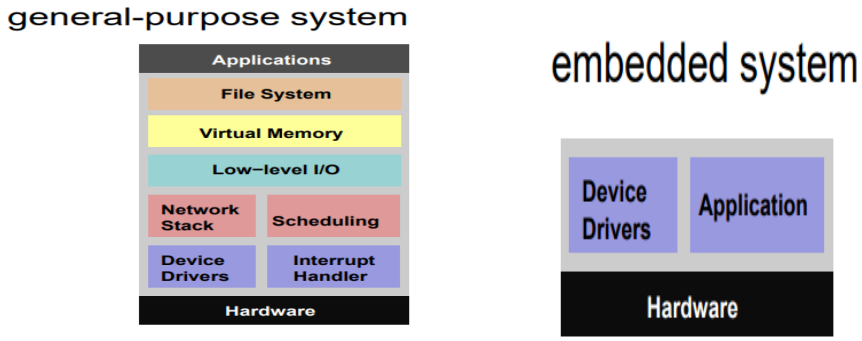
Wombat Operating System

Done By
S Subhiksha
2019506099

Wombat OS

Wombat OS was developed by National ICT Australia.It belongs to embedded real-time operating system.Embedded systems are device which contains one or more processor chip that designed to perform specific task.

There is a difference between general OS and embedded OS.



In Embedded operating system, there is no OS or OS with minimal features.It won't provide protection.There are many challenge to implement embedded OS.

Wombat acts as portable and user mode Linux for embedded system. It provides high performance with any architecture like x86 or MIPS.Wombat act as a port of Linux kernel to L4 micro-kernel.Wombat OS will run on top of L4 kernel like L4Ka:Pistachio

L4Ka:Pistachio

- ✓ If IPC takes longer time, it removes that process.
- ✓ Number of virtual registers get reduced.
- ✓ Timeout process get removed
- ✓ It has physically addressed TCB.

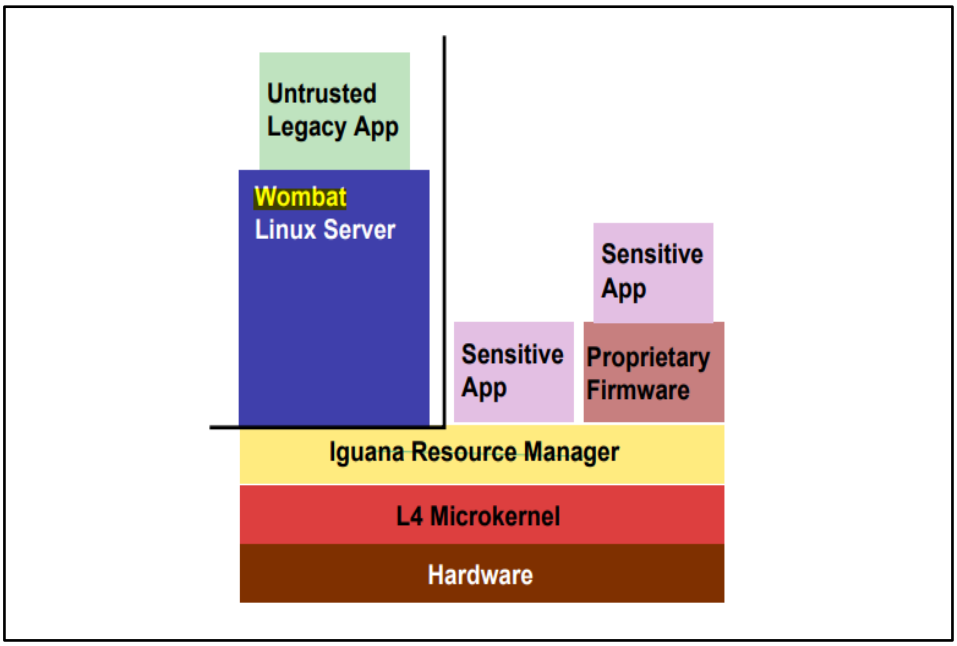
Introduction to Wombat OS

Objectives

- ✓ It provides Linux API for embedded system.
- ✓ It supports many architectures other than x86.
- ✓ It provides easy maintenance
- ✓ It can be easily integrate with Iguana

Linux in user-mode usually run in small, fast micro-kernel but this kernel won't provide any services.Whenever running Linux on micro-kernel ,it became unprivileged one. So we introduce wombat as Linux server.L4 is a micro-kernel where top of micro-kernel wombat works.L4 won't provide any services,polices,and

mechanism.But in OS we need some basic services like Memory management, scheduling for process management,and protection.So we are introducing Iguana OS which provide all these basic services and more suitable for embedded OS.Iguana is more comfortable with L4 kernel.It provide thread management,virtual memory to allocate and deallocate memory for process.It provides framework for protection management and more flexible system.

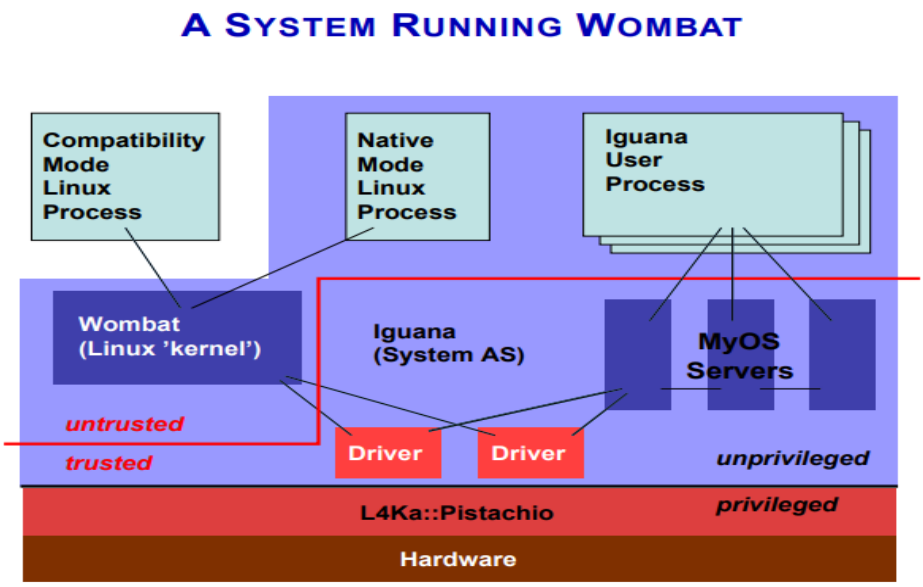


Running Wombat on a micro-kernel

Structure Of Wombat

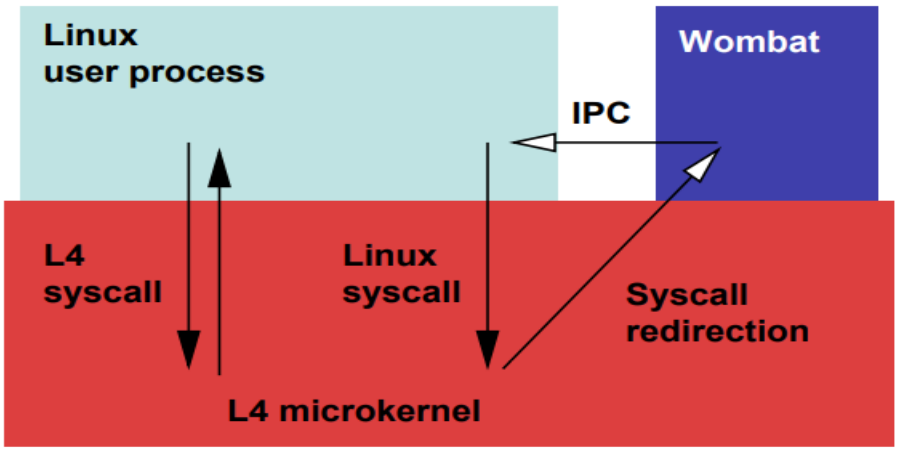
Iguana is an environment which supports and provide space for Wombat. micro-kernel won't give any services ,so iguana is going to provide services like memory sharing and allocating,memory management and protection and resource management.As already stated ,Wombat provides high performance due to less context switch time which was provided by Iguana.Using virtually-addressed cache page, context-switch time get reduced.But there is a problem with virtually-addressed cache , this type of cache is retrieved only with virtual address.Virtual address is pointed out to particular process but different process have same virtual address space. In Linux, text and stack segment are same for all process only the code will change.Hence, there is a chance of cache flush. Iguana avoid this cache flash by providing non-overlap address space.Iguana supports both shared space(single-addres space) and address space of own process.

Shared address space provide fast context-switch, make ease of sharing data among process but it may lead to inconsistent state.



System Call in Wombat

In L4 micro-kernel, Linux process is executed as single thread.there is an Inter process communication between process running on L4 and Linux user mode. L4 has different number for system call when compared to Linux.Linux application run in two modes.Naive mode which are work with shared address mode and make use of iguana features.Compatible mode used by some application which are not able to run in naive.They can communicate with wombat only, not with other process that are running in shared address space.This mode can't use L4.whenever Linux system call happens,this was treated as exception by L4 and redirected to Wombat .This process is called trampoline(system call redirection).Wombat consider this as normal IPC message passing and perform system call and redirected to Linux application.



Trampoline

Other alternative approach is also available.Instead of using different system call number we can make use of different approach to distinguish between L4 and Linux system call.Linux uses software interrupt to handle system call while L4 jumps to invalid address and cause prefetch abort exception. Other exception are handled by passing messages to Wombat .

Thread Management In Wombat

Thread management is provide by Iguana.Iguana threads are similar to L4 threads. Threads which lies on same protection domain operated by local l4 syscall.Certain operation on thread will require privileged so it can't be done by iguana.Privileged operations are carried out by L4 kernel.For example, thread creation and deletion will be done by L4 kernel by invoking "ThreadControl()" system call.

Thread Operation

- ✓ Thread creation

thread cap = pd->new thread(&l4 tid);

Which returns iguana thread id and L4 thread id.Usually newly created threads are inactive.This can be activated by "*ExchangeRegisters()*" used in L4 and Iguana uses "*tid->start(ip,sp)*" method.

- ✓ Getting L4 thread id

l4tid = tid->l4 tid();

- ✓ Getting own thread id

tid = myself();

- ✓ Create protection domain of thread

pd = tid->domain();

- ✓ Modifying scheduling parameters

tid->schedule info(&info);

Scheduling in Wombat

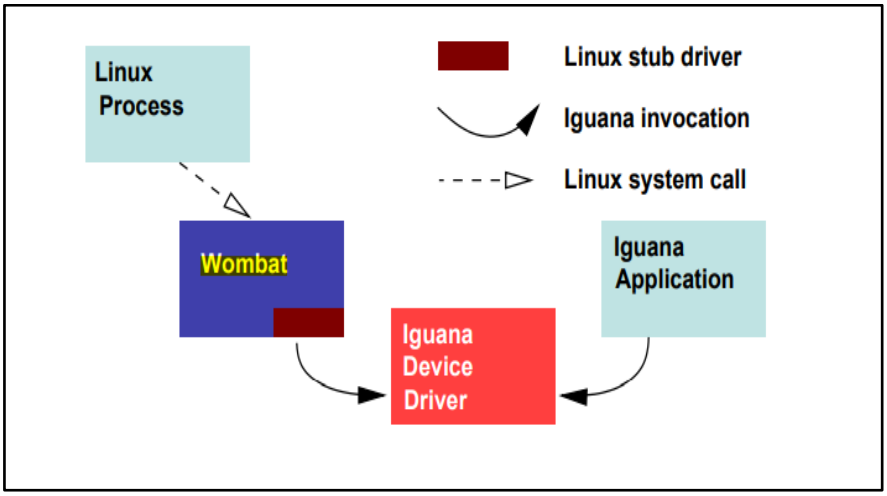
Linux process has normal scheduling , similar to Linux OS but L4 has round robin with priority scheduling.This scheduling mechanism is happen in iguana and wombat.L4 gives priority to schedule real-time process but not schedule Linux process.The process corresponding with iguana has higher priority as they are real time process than wombat. So Linux process get preempted whenever Iguana process came into picture.In other words, Linux process will run only if there is no higher priority process.To implement Linux scheduling we need wombat.Wombat allows

only one user process will run in CPU.L4 ensures that Linux applications is always under the control of wombat.

Wombat has single time thread which maintains Linux time slicer.This thread waits for time to finish the process(timeout).If the process timeout,wake this thread and allow other Linux process to run.Linux scheduler can't modify the scheduler decision.Locks are always available in Linux kernel and other locks are not needed for scheduling.In Linux ,whenever CPU is in idle state ,it calls CPU_idle() function.But in wombat the kernel thread slept until next process came or interrupt.

Device Drivers in Wombat

System's device are controlled by single driver. This single driver may be Linux driver or iguana driver.User mode process will run in protection mode inside iguana and managed by iguana drivers.Linux drivers will use wombat.It won't allow sharing of device between user process in Linux and real-time process.



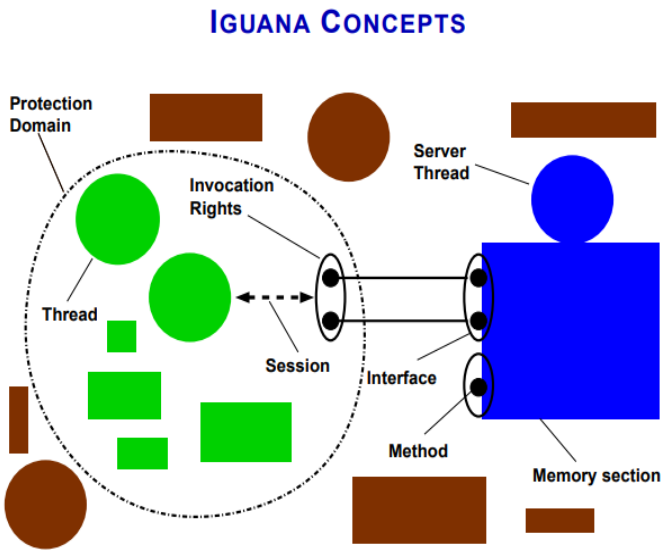
Shared Device Drivers in Wombat System

In-order to achieve shared device access, we need proper driver on one side and other side will contain proxy(stub) driver.The proxy driver invoke the request to other driver.Proper driver will manage the user process and real-time process and allocate the device accordingly.

As shown in above figure, Linux process make appropriate system call to access device which was handled by wombat. Moreover, Wombat has Linux proxy driver,which, in-turn call Iguana device driver which is proper driver.Iguana device driver will now allocate device to user process and real-time process.But whenever the process run in iguana ,then it can call iguana device driver directly without the use of wombat.Iguana device driver has similar performance with in-kernel drivers.The

drivers are portable across platform ,between Linux and iguana, and architecture. But now-a-days drivers are coming for number of proprietary device, LCD screen,Ethernet etc.

Memory Management In wombat



In wombat , memory is managed by Iguana.It has six objects which are providing basic requirement for wombat.Memory section is a unit which allocates for virtual memory and provide memory protection as iguana has shared space address so it avoids one process access the memory of other process.It has one thread called server which is associated with memory section.This server invoked well-defined interface.Moreover, it allocates certain amount of memory for virtual memory.

mem_cap=pd->new_mem(size);

As already stated, Memory section is an object which supports user defined data.It encapsulates methods and data.

service = memory (data) + server (thread) + methods

The service for the process being created by registering server thread for virtual memory on memory section

base->new server(thread id);

Where base = base address of memory section.

Once server created , create interface by user-defined method

base = iid->new cap();

where iid is number of new interface created.This will support user-defined method and interface can have many method.The unique number is provide for each interface

and method by system implementer.Memory sections has pseudo methods like read(),write(),and execute().These methods has no actual method regarding their operations.

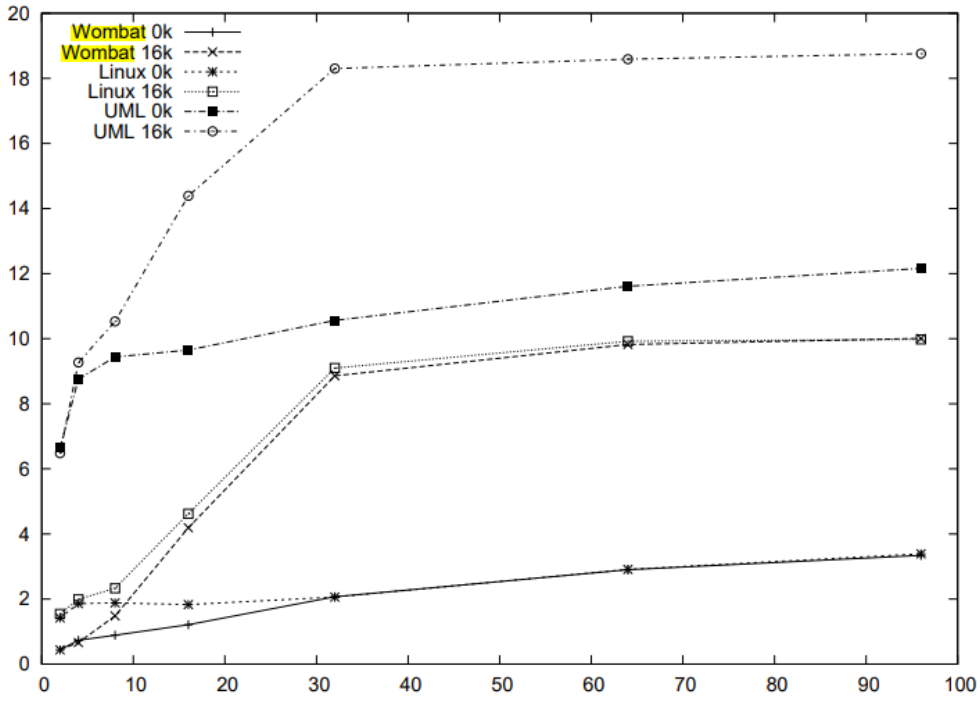
Performance of wombat

Performance of wombat will compared with Linux and user-mode Linux.The performance was made by set of micro-benchmark.Micro-benchmark is a small testing unit to check performance.The performance was made to check system call and pipe.

Benchmark	Linux	Wombat	UML
syscall	0.392	1.38	8.99
pipe	5.05	8.83	66.8

Table 1: Latency benchmark performance

The cost of system call in Linux server will be high.The first benchmark tested is “null” System call latency.This shows the overhead of all system call and worst case scenario. It will give the cost of communicating with system.The second benchmark used is pipe which gives the cost of communicating with different process.



Above diagram illustrate the time taken by different OS for context switch.

It has been observed that wombat is quite performance efficient compared to UML.Wombat performs context switch much faster than the native Linux itself. In the mean time, UML costs 8 micro-second overhead for context switch and takes similar overhead for null system call too.Hence, Wombat OS is the performance efficient compared to other Linux system.

Advantages of Wombat

When running Linux in user mode in L4 kernel provide Linux API for embedded system

Strength of wombat

- ✓ As we saw already ,it is portable across many architecture.Wombat will run in different architecture where Linux can't run .For instance,wombat with kernel 2.6 will run on ARM but native Linux can't work with ARM.
- ✓ Wombat runs with iguana environment ,so it has benefit of reduce context switch time and fast address space switching.
- ✓ Wombat make use of iguana drive framework, which allows make use of both iguana and Linux device driver.
- ✓ Wombat will easily sync with Linux distribution and has less dependency with Linux kernel.