

LINUX ON CELL

RAONI FASSINA FIRMINO

R.A.: 069297

Pós-graduação em Ciência da Computação
Mestrado Acadêmico
Instituto de Computação
UNICAMP

Setembro de 2007
CAMPINAS – SP

LINUX ON CELL

LINUX

CELL BROADBAND ENGINE

LINUX EXTENSION FOR CELL

LINUX

Desenvolvido por Linus Torvalds;

Sistema operacional tipo Unix;

Escrito a partir do código fonte do sistema Minix;

(Minix é uma simplificação do Unix)

(Unix proveio do sistema Multics)

Código fonte disponível sob licença GPL;

Escrito em C + código Assembly;

(desempenho e especificidades da plataforma)

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Linux = Núcleo/kernel do sistema operacional

Linux = SO quando este é definido como gerenciador de recursos de hardware

Termo Linux:

- Sistema completo;
- Núcleo Linux;
- Outros programas de sistema.

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Sistema Completo GNU/Linux:

- Sistema operacional Linux;
- Conjunto de programas GNU(Free Software Foundation).
(exemplo: interpretador de comandos, utilitários, bibliotecas, compiladores, etc.)

Maioria dos sistemas:

- Ferramentas e utilitários baseados no BSD;
- Usam XFree86 ou X.Org para oferecer interface gráfica (Sistema X).

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Arquitetura:

- Kernel monolítico;
- Suporte a módulos carregáveis;
- Multitarefa;
- Multiprocessamento;
- Memória virtual por paginação;
- bibliotecas compartilhadas (copy-on-write);
- Acesso ao kernel através de chamadas de sistema.

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Kernel controla:

Criação de processos, solicitação de memória, operação com arquivos, etc.

Comunicação Kernel-hardware:

- Drivers de dispositivos ;
(para o kernel reconhecer e gerenciar o hardware)
- Interrupções.
(para o Hardware comunica-se com o kernel)

LINUX

Linus Torvald

Kernel Monolítico

Multics > Unix > Minix > LINUX

Portabilidade

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Linus Torvald

Kernel Monolítico

Multics > Unix > Minix > LINUX

Portabilidade

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Sistema Operacional Unix:

- Portável
- multitarefa
- Multiusuário
- Conceito simplificado do Multics

Multics:

Primeiro Sistema de tempo compartilhado
[AT&T (Bell Labs), General Electric (GE) e MIT]

LINUX ON CELL

Raoni Fassina Firmino

LINUX

Unix Primeira Versão(1969):

- Ken Thompson (Bell Labs da AT&T)
- Multics reescrito em um conceito menos ambicioso
- Para um computador PDP-7
- Batizado de Unics
- Escrito em linguagem de montagem (assembly)

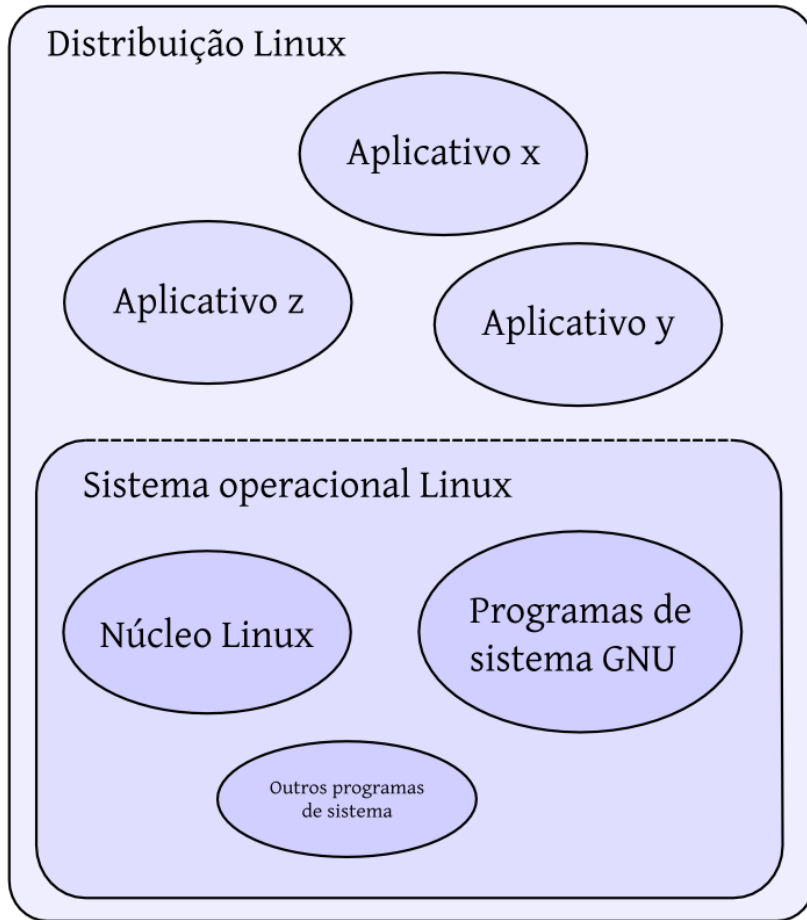
Unix Segunda Versão(1973):

- Dennis Ritchie e Ken Thompson
- Para um computador PDP-11.
- Reescrito usando a linguagem C
- Uso da linguagem C e a rápida difusão do Unix.

LINUX ON CELL

Raoni Fassina Firmino

LINUX



Fonte: Wikipedia

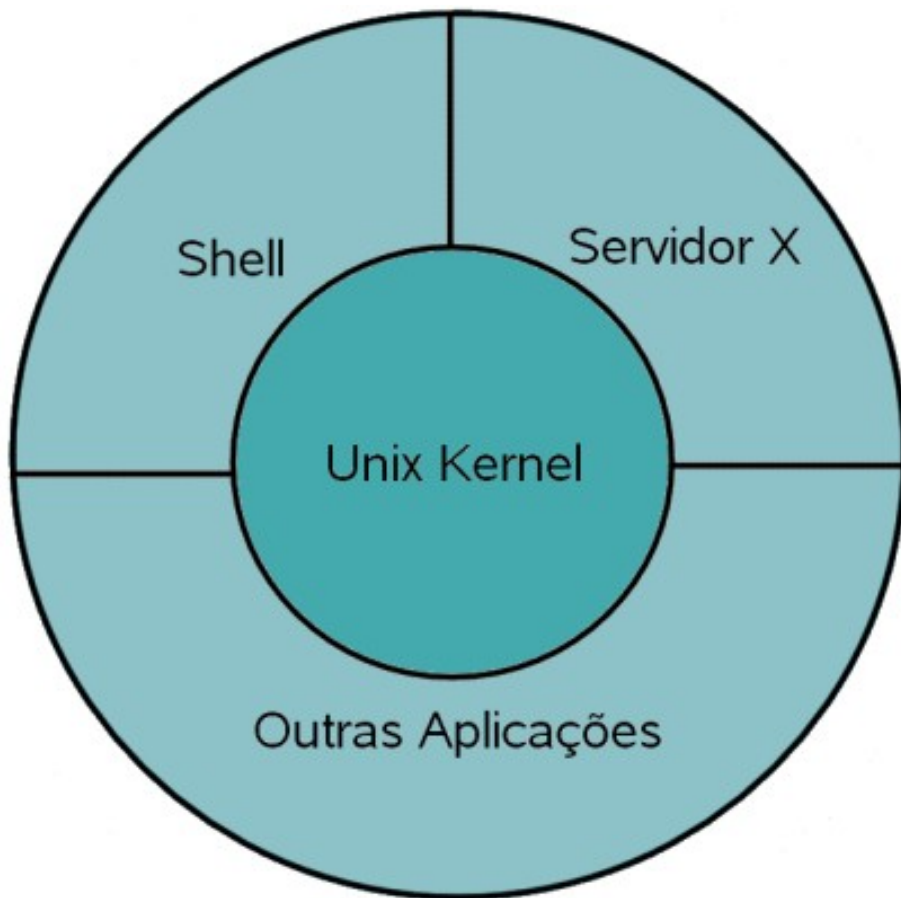
Distribuição Linux:

- Sistema Operacional GNU/Linux
- Pacote de Aplicativos
- Modificações na estrutura do sistema

LINUX ON CELL

Raoni Fassina Firmino

LINUX



Fonte: Wikipedia

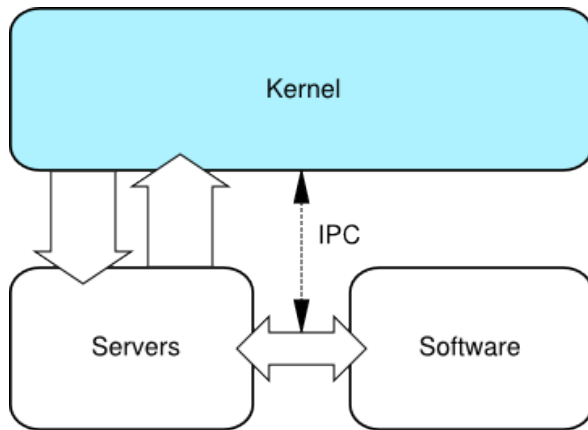
Partes do Sistema Operacional UNIX:

- Kernel
- Shell
- Servidor X
- Outras aplicações

LINUX ON CELL

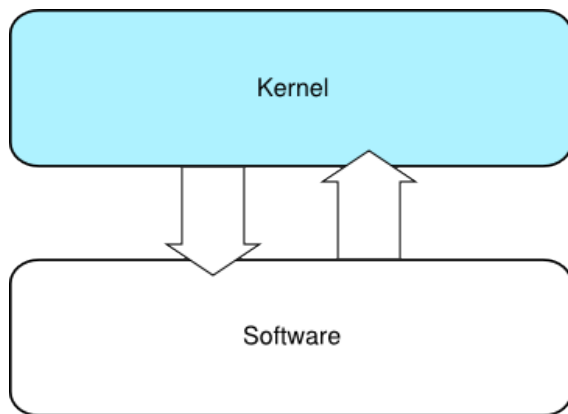
Raoni Fassina Firmino

LINUX



Microkernel

- Processos em modo usuário
- Mínimo possível em modo kernel



Kernel Monolítico

- Kernel inteiro em modo kernel
- Menos trocas entre modo kernel e usuário

Fonte: Wikipedia

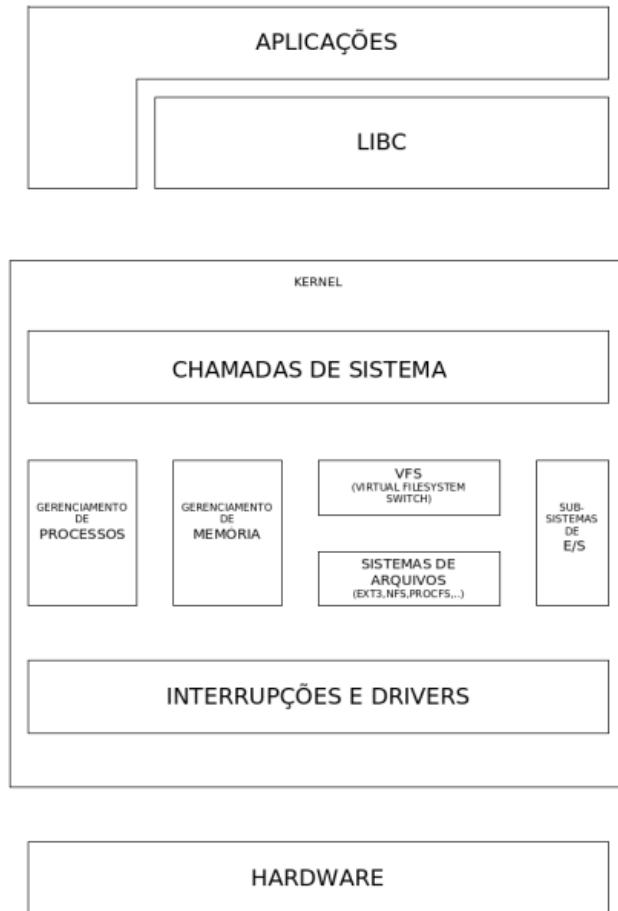
LINUX ON CELL

Raoni Fassina Firmino

LINUX

Kernel LINUX

- Gerenciador de Memória
- Gerenciador de Processos
- Virtual File System
- Sistemas de Arquivo
- Drivers e Interrupções
- Gerenciamento de I/O

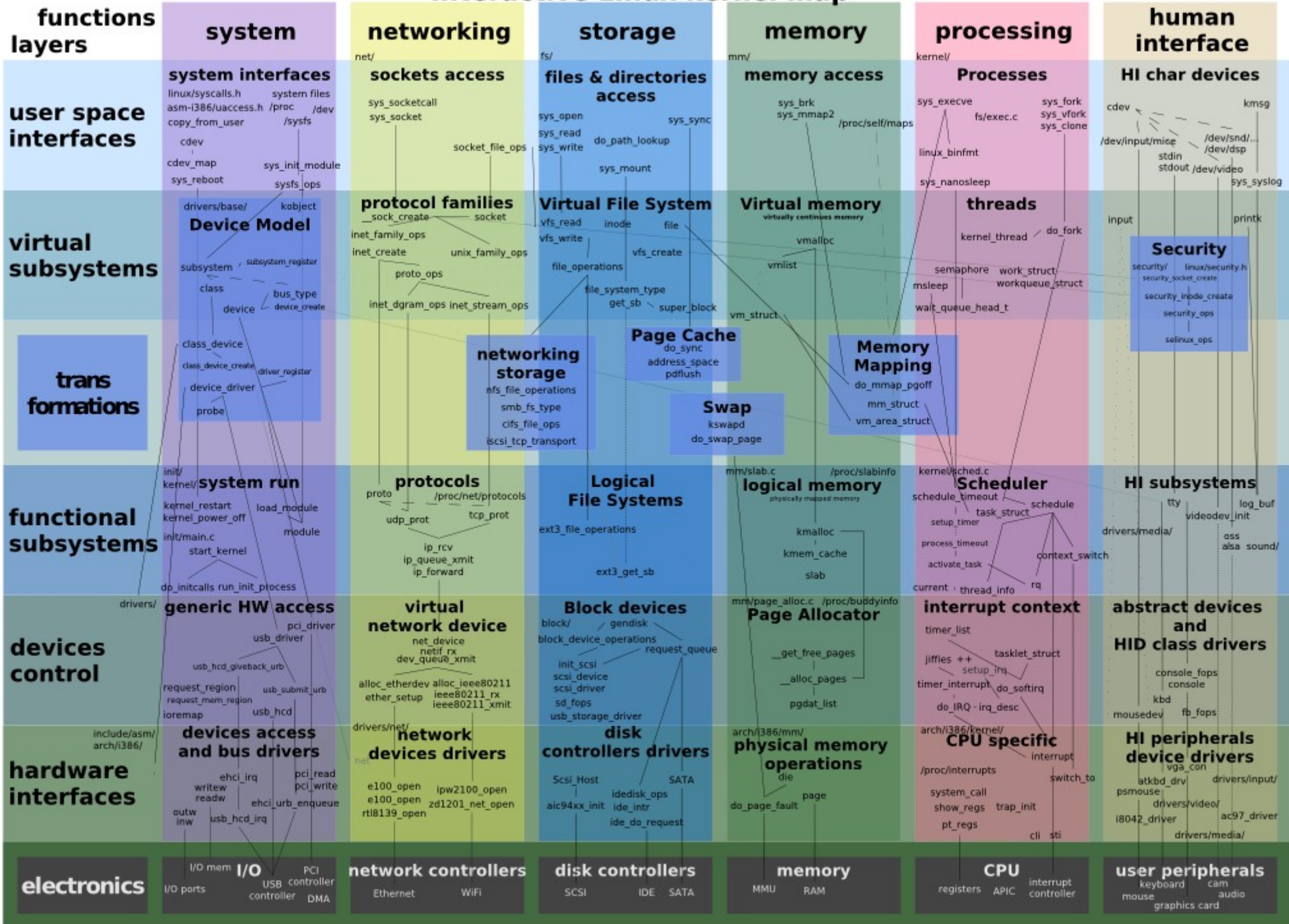


Fonte: Wikipedia

LINUX ON CELL

Raoni Fassina Firmino

Interactive Linux kernel map



CELL BROADBAND ENGINE

IBM – SONY – TOSHIBA

Cell Broadband Engine Arquiteture

Multicore Heterogênea:

–Power Processor Element

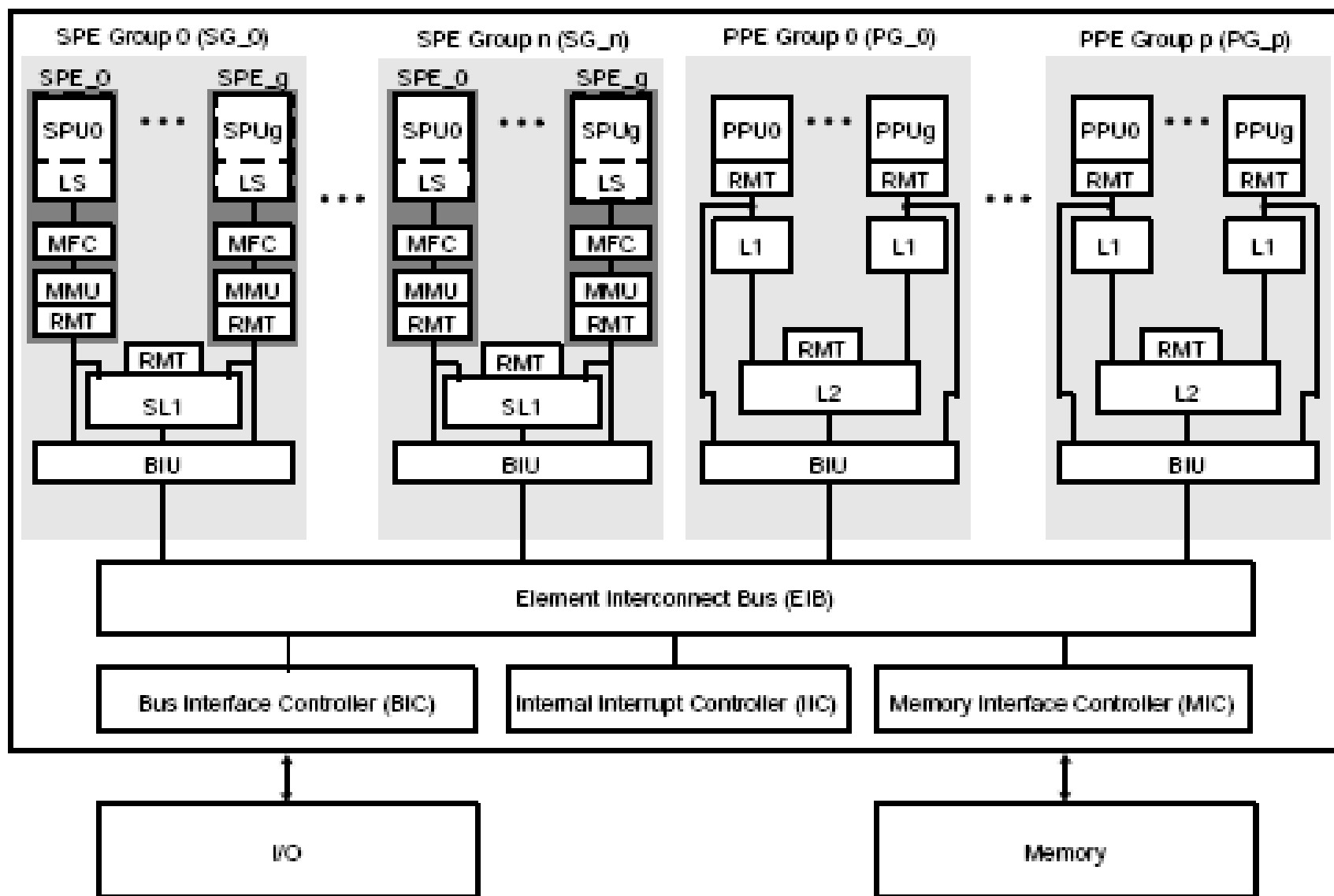
Prover funções centralizadas do sistema;
Funções de propósito geral;
Controle de tarefas, etc.

–Synergistic Processor Element

Otimizado para processamento de dados.

LINUX ON CELL

Raoni Fassina Firmino



BIC Bus interface controller
 BIU Bus interface unit
 IIC Internal interrupt controller
 L1 Memory cache internal to the CPU
 L2 Memory cache external to the CPU
 LS Local storage
 MFC Memory flow controller
 MIC Memory interface controller

MMU Memory management unit
 PPE PowerPC Processor Element
 PPU PowerPC processor unit
 RMT Replacement-management table
 SL1 First-level cache
 SPE Synergistic Processor Element
 SPU Synergistic processor unit

CELL BROADBAND ENGINE

**Baseada na arquitetura 64-bit Power
da IBM:**

Suporte a todas as aplicações desenvolvidas para
para Power e PowerPC de 32 e 64bits;

Mesmo sistema:

- Segmentação;
- Tabelas de paginas;
- Modelo de coherencia do sistema de mapeamento de memória;
- Virtualização por hardware.

Outras características da arquitetura Power.

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Power Processor Element(PPE):

Arquitetura PowerPC;

Suporte a Instruções VMX

-Power Processor Unit(PPU):

- Unidade de execução: Power execution Unit (PXU);
- Memória cache L1.

-Memória cache L2.

CELL BROADBAND ENGINE

Power Processor Element(PPE):

Dual Thread

Suporte a Instruções VMX

-Power Processor Unit(PPU):

- Unidade de execução: Power execution Unit (PXU);
- Memória cache L1.

-Memória cache L2.

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Synergistic Processor Element(SPE):

- Processador RISC
- Totalmente autônomo
- Suporta unicamente instruções SIMD
- Banco de registradores unico
 - 128 registradores x 128bits
(guarda todos os tipos de dados)

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Synergistic Processor Element(SPE):

–Synergistic Processor Unit (SPU)

- Unidade de execução: Synergistic Execution Unit (SXU)
- Memória local preferencialmente implementada "on-chip"

–Controlador de memória: Memory Flow Control (MFC)

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Memory Flow Control (MFC):

- Ligação entre a memória local (LS) e barramento interno do sistema (EIB);
- Otimizado para transferência de dados;
- Provê Direct Memory Access (DMA) para o SPE;
- Mecanismo de sincronização para comunicação;
- Considerada um core ou unidade de processamento (não é uma definição técnica).

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Element Interconnect bus (EIB):

- Barramento interno que interliga o sistema
- Modelo de anel
- Atual implementação:
 - 2 anéis em cada sentido (4 ao todo)
 - Arbitro do Barramento permite 3 transferências concorrentes para cada barramento
 - 16Bytes de entrada e 16Bytes de saída simultâneos para cada porta do barramento.

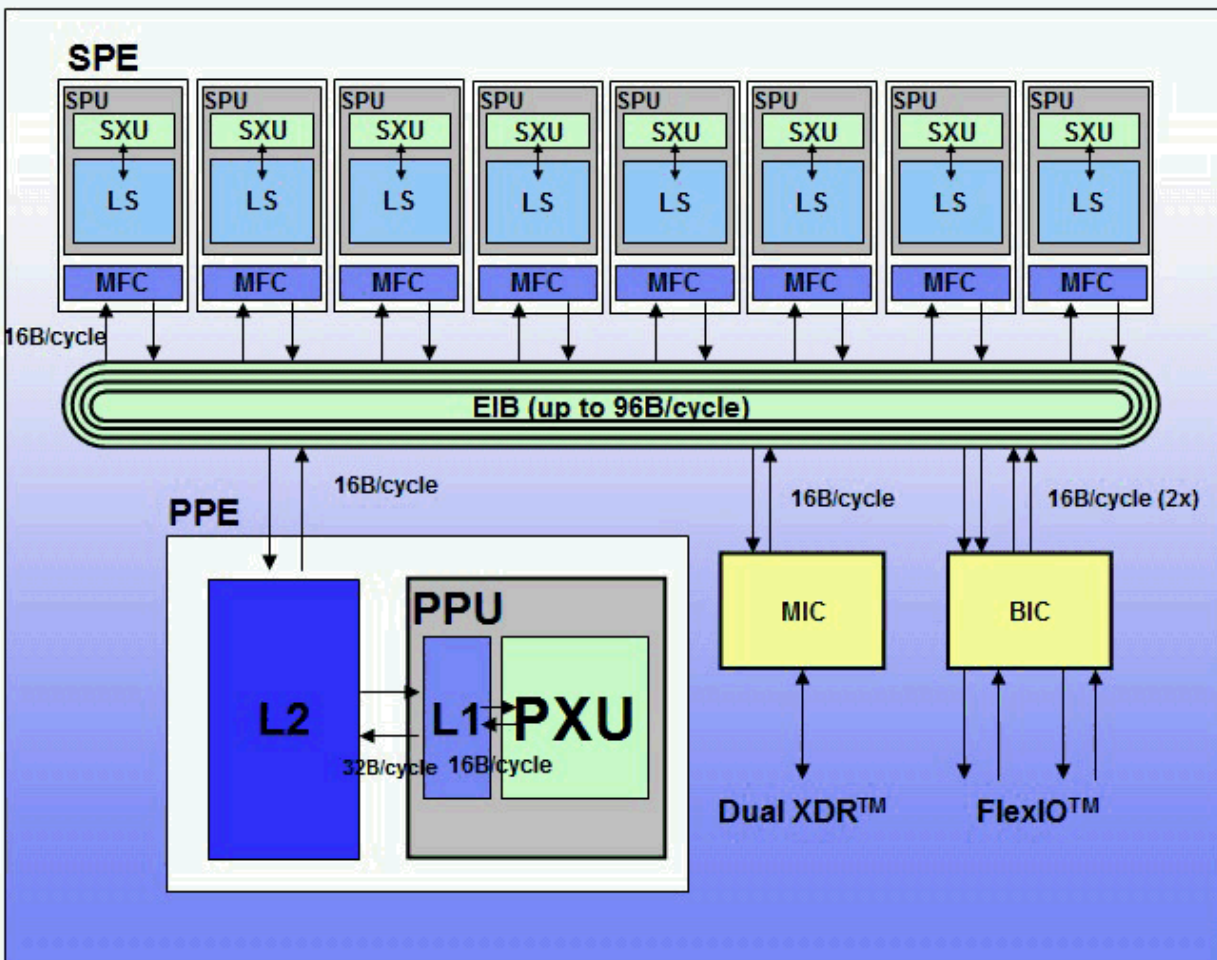
LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Primeira implementação:

- 3,2GHz;
- 1 PPE;
- L1: 32KB I + 32KB D
- L2: 512KB
- 8 SPE;
- LS: 256KB
- EIB de 4 anéis.



Source: M. Gschwind et al., Hot Chips-17, August 2005

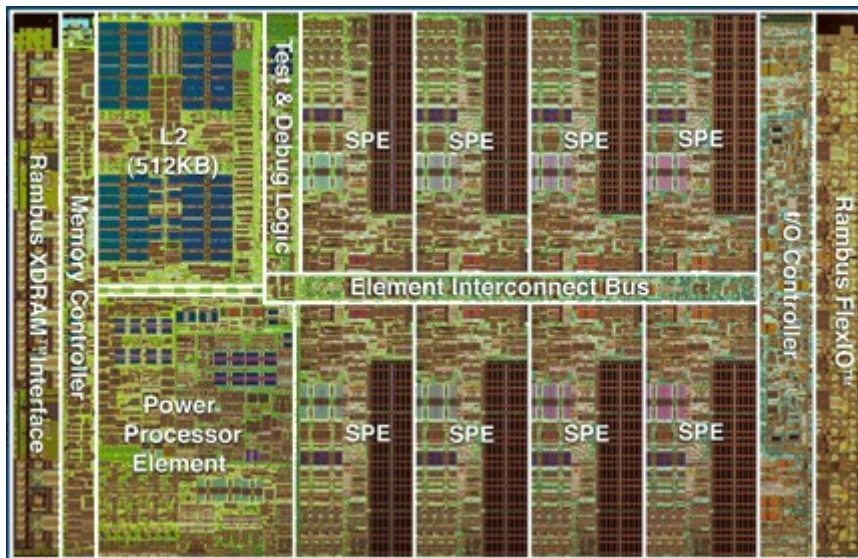
© 2007 IBM Corporation[1]

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

CARACTERISTICAS FISICAS[1]:



©2007 IBM Corporation[1]

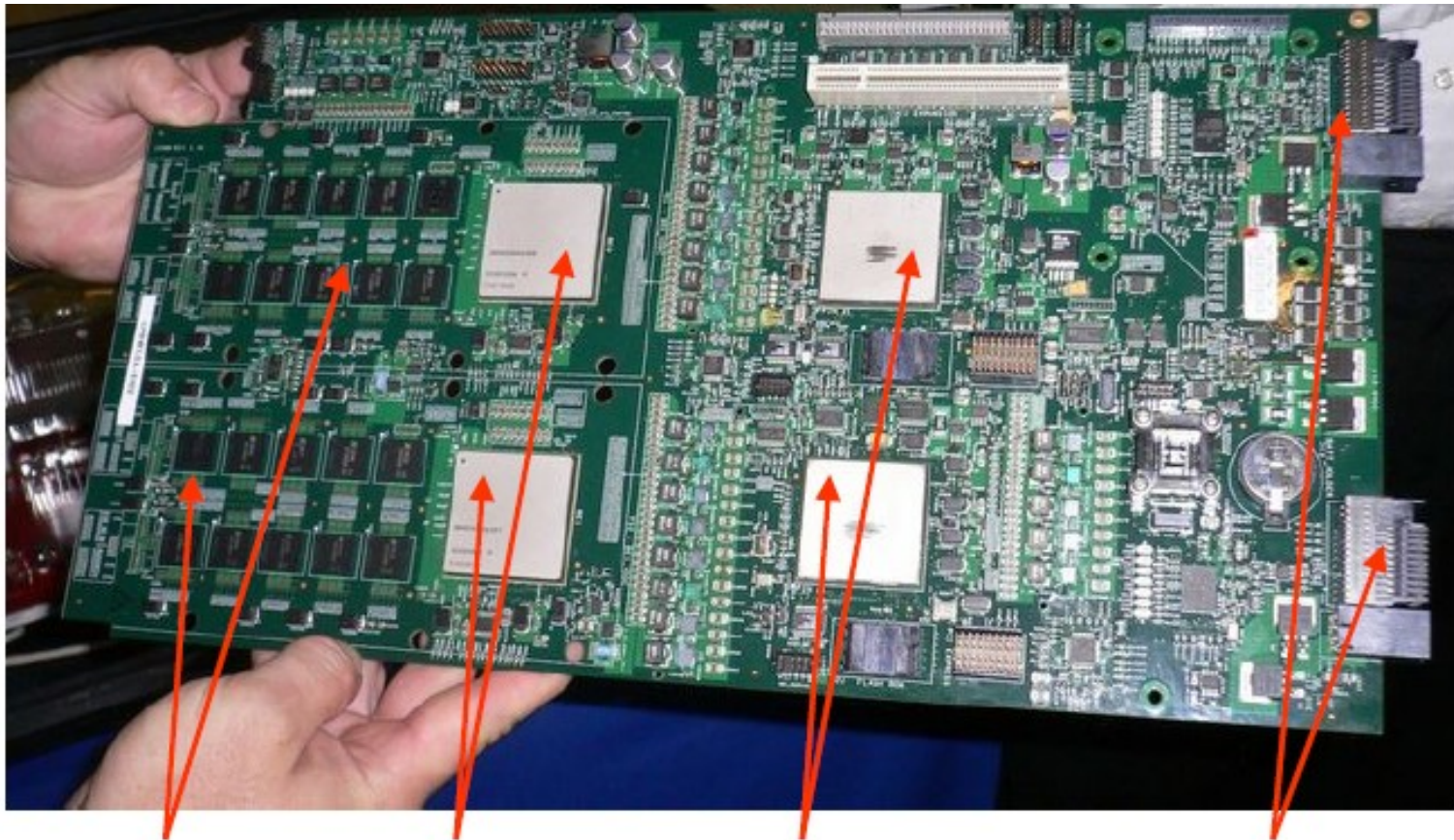
Observed clock speed: > 4 GHz

- Peak performance (single precision):
> 256 GFlops
- Peak performance (double precision):
> 26 GFlops
- Local storage size per SPU: 256KB
- Area: 221 mm²
- Technology: 90nm SOI
- Total number of transistors: 234M

LINUX ON CELL

Raoni Fassina Firmino

CELL BLADECENTER QS20



1GB XDR Memory

Cell Processors

IO Controllers

IBM Blade Center interface

©2007 IBM Corporation[3]

LINUX ON CELL

Raoni Fassina Firmino

CELL BLADECENTER QS20

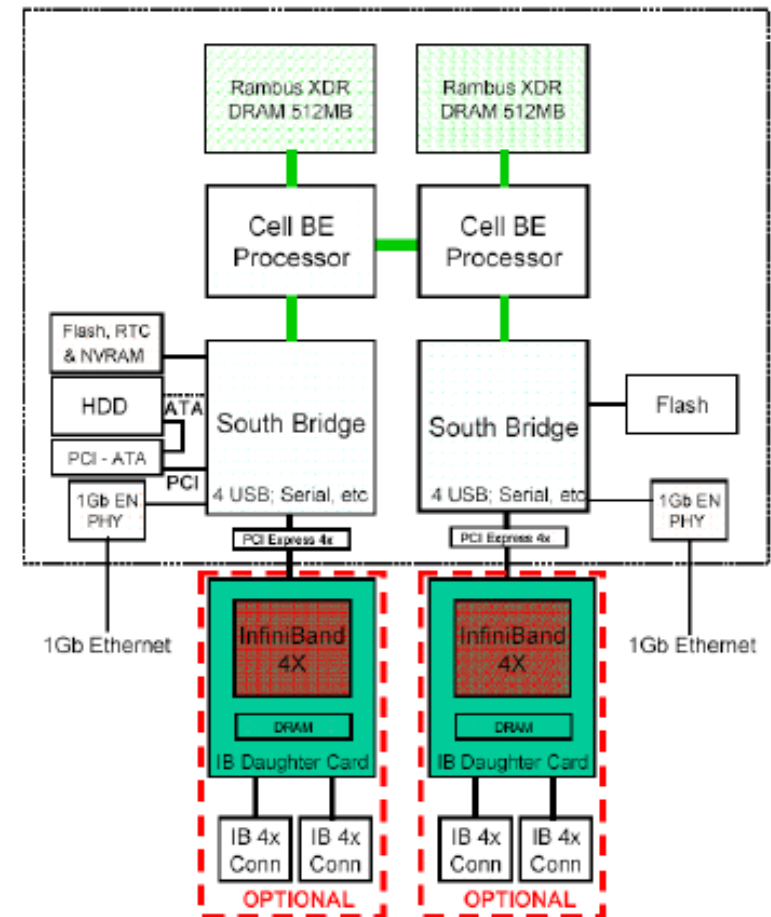
Cell BE Processor Blade)(~500GFLOPS peak)[3]

- Dual 3.2GHz Cell BE Processor Configuration
- 1GB XDRAM (512MB per Processor)
- Blade-mounted 40GB IDE HDD
- Dual Gigabit Ethernet (GbE) controllers
- Double-wide blade (uses 2 BladeCenter slots)
- Infiniband (IB) Option:

Qty 0-2 IB 4x Host Channel Adapters

BC Chassis Configuration (~3.5TFLOPS peak)[3]

- Standard IBM BladeCenter One
- Max. 7 Blades per chassis (QS20 - 2 slots each)
- 2 Gigabit Ethernet switches
- External IB switches required for IB option



©2007 IBM Corporation[3]

LINUX ON CELL

Raoni Fassina Firmino

CELL BROADBAND ENGINE

Cell BE Software Development Kit (V2.1) [3]

Plataformas Hosts:

- x86, x86-64, 64-bit PowerPC (PPC64) e Cell BEbased BladeCenter

Conteúdo:

- Ferramentas de desenvolvimento
- Compiladores PPU e SPU
- software libraries e sample source
- system simulator
- Kernel Linux

Distribuição depacotes RPM:

- IBM alphaWorks website: Full-System Simulator, XL C/C++ compiler, libraries and samples, SPU timing tool, FDPR-Pro performancetuning utility, the Eclipse Integrated Development Environment (IDE) for Cell BE, and the cellsdk installation script.
- Barcelona supercomputing center: open source packages – the GCC-based tool chain, the Linux kernel and support libraries, and system support for the Cell BE Full System Simulator.

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

Extensão para Kernel Linux 2.6.20

Distribuído pela Barcelona Supercomputer Center

Suporte ao hardware do Cell BE

Bibliotecas de Suporte

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

Extensão para o Kernel Linux 2.6.18 PPC64

– Adiciona modelo heterogeneo de lwp/thread

- SPE thread API created (similar to pthreads library)
- User mode direct and indirect SPE access models
- Full pre-emptive SPE context management
- spe_ptrace() added for gdb support
- spe_schedule() for thread to physical SPE assignment
 - currently FIFO – run to completion

– Compartilhamento do espaço de endereçamento de threads SPE com o processo PPE pai (através de DMA)

- Demand paging for SPE accesses
- Shared hardware page table with PPE

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

- **Thread PPE proxy** alocada para cada SPE thread:
 - Unico namespace para PPE e SPE threads
 - Assist in SPE initiated C99 and POSIX-1 library services
- **Gerenciamento de SPE Error, Event e Signal** diretamente para a thread PPE pai
- **SPE elf objects wrapped into PPE shared objects with extended gld**
- **Todas as extensões para o Cell estão na camada dependente de arquitetura**
(architecture dependent layer) (subarvore do PPC64)

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

Upgrade do kernel Linux 2.6.20:

Aumento do performance

Aprimoramento: preemptive scheduling of SPE tasks

SPE logical affinity support

Aprimoramento: 64 KB Local Store page mapping

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

Multitasking SPEs – Kernel managed[3]

–Kernel-level SPE management model:

- SPE as a device resource
- SPE as a heterogeneous processor
- SPE resource represented as a file system

LINUX ON CELL

-SPE scheduling and virtualization

- Maps running threads over a physical SPE or a group of SPEs
- More concurrent logical SPE tasks than the number of physical SPEs
- High context save/restore overhead
 - favors run-to-completion scheduling policy
- Supports pre-emptive scheduling when needed
- Supports memory protection

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

SPE Management Library (libspe)

SPEs expostos como threads:

- SPE thread model interface (similar to POSIX threads.)
- SPE thread: Local store, Register file, Program counter e MFCDMA queue;
- Associação com um unica tarefa(task) Linux;
- Opções de controle/gerenciamento:
 - Threads - create, groups, wait, kill, set affinity, set context;
 - Thread Queries - get local store pointer, get problem state area pointer, get affinity, get context;
 - Groups - create, set group defaults, destroy, memory map/unmap, madvise;
 - Group Queries - get priority, get policy, get threads, get max threads per group, get events;
 - SPE image files - opening and closing;
 - Etc.

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

SPE Management Library (libspe)

SPE Executaveis

–Programa SPE gerenciado por um PPE
Controlador("executive").

–PPE Controlador:

Carregamento e execução do programa CPE

Presta serviços de requisições de I/O (eg, fopen, fwrite, fprintf) e de
memoria (eg, mmap, shmat, ...).

LINUX ON CELL

SPE Management Library (libspe) (Versão 2.1)

standalone SPE executive (Elfspe):

- Programa PPE
- Execução de programa SPE a partir do terminal Linux
- Registrado no kerneldurante o boot (/etc/rc.d/init.d)
- libspe distribuída sobre licença GPL

LINUX ON CELL

PPE programming model (participation)[3]

PPE is a 64-bit PowerPC core, hosting operating systems and hypervisor

PPE program inherits traditional programming models

Cell environment: a PPE program serves as a controller or facilitator

- CESOF support provides SPE image handles to the PPE runtime
- PPE program establishes a runtime environment for SPE programs
 - e.g. memory mapping, exception handling, SPE run control
- It allocates and manages Cell system resources
 - SPE scheduling, hypervisor CBEA resource management
- It provides OS services to SPE programs and threads
 - e.g. printf, file I/O

LINUX ON CELL

Raoni Fassina Firmino

LINUX ON CELL

Small singleSPE models

Separated SPE and PPE address spaces – LS / EA

Explicit input and output of the SPE program

- Program arguments and exit code per SPE ABI
- DMA
- Mailboxes
- SPE side system calls

LINUX ON CELL

Small singleSPE models – tools and environment

- SPE compiler/linker compiles and links an SPE executable
- The SPE executable image is embedded as reference-able RO data in the PPE executable (CESOF)
- A Cell programmer controls an SPE program via a PPE controlling process and its SPE management library
 - i.e. loads, initializes, starts/stops an SPE program
- The PPE controlling process, OS/PPE, and runtime/(PPE or SPE) together establish the SPE runtime environment, e.g. argument passing, memory mapping, system call service.

LINUX ON CELL

Raoni Fassina Firmino

BIBLIOGRAFIA

- [1] IBM Research. Cell/B.E. project page, 2007.**
[<http://www.research.ibm.com/cell/>]
- [2] Barcelona Supercomputer Center. Linux on Cell, 2007.**
[<http://www.bsc.es/projects/deepcomputing/linuxoncell/>]
- [3] IBM Corporation. Cell Programming Workshop, 2007.**
[<http://lampiao.ic.unicamp.br/wiki/CursoCell/>]