Suporte para implementação de máquinas virtuais nativas

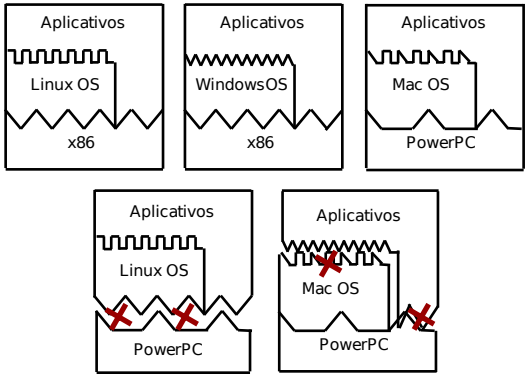
Alisson Linhares de Carvalho

IC/Unicamp

15 de maio de 2015

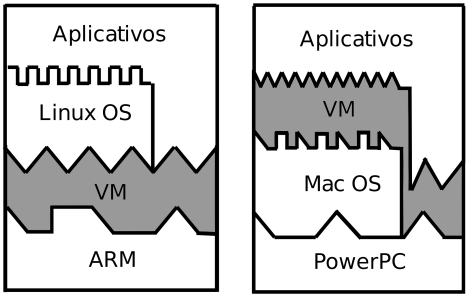
Introdução

Os programas compilados dependem da arquitetura da máquina alvo.

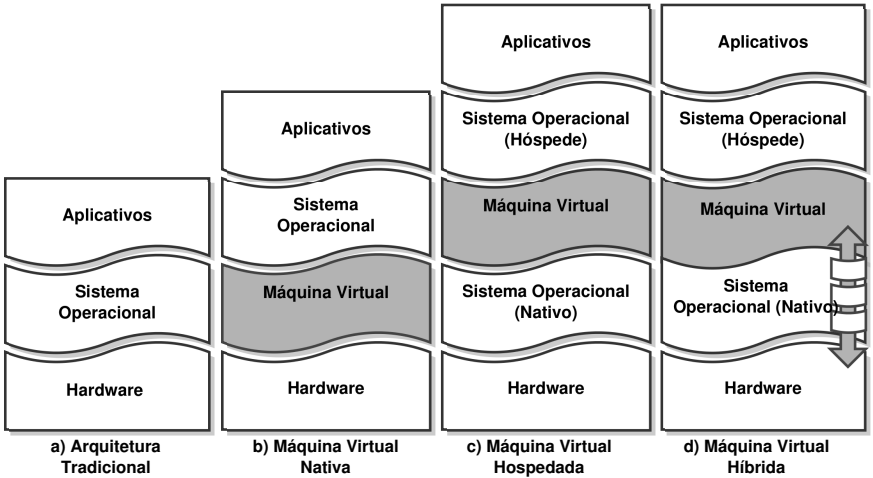


Introdução

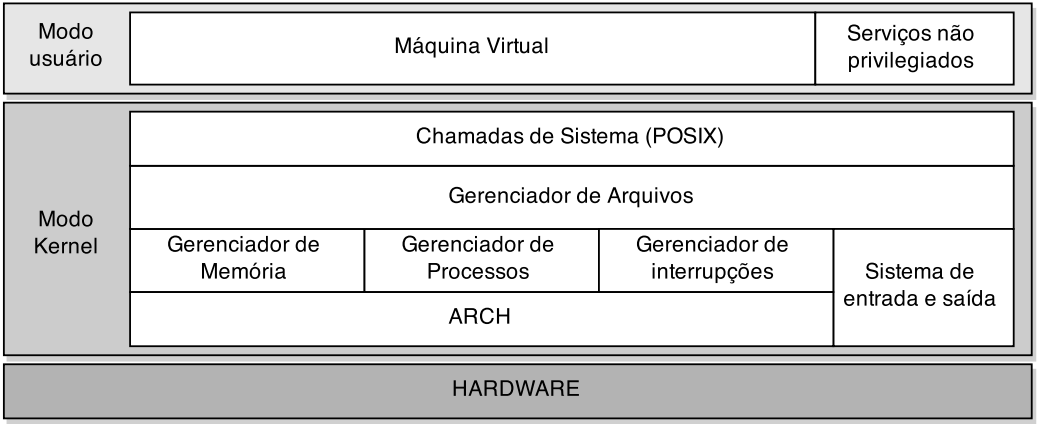
As máquinas virtuais são programas de computador que emulam uma interface para execução de outros programas.



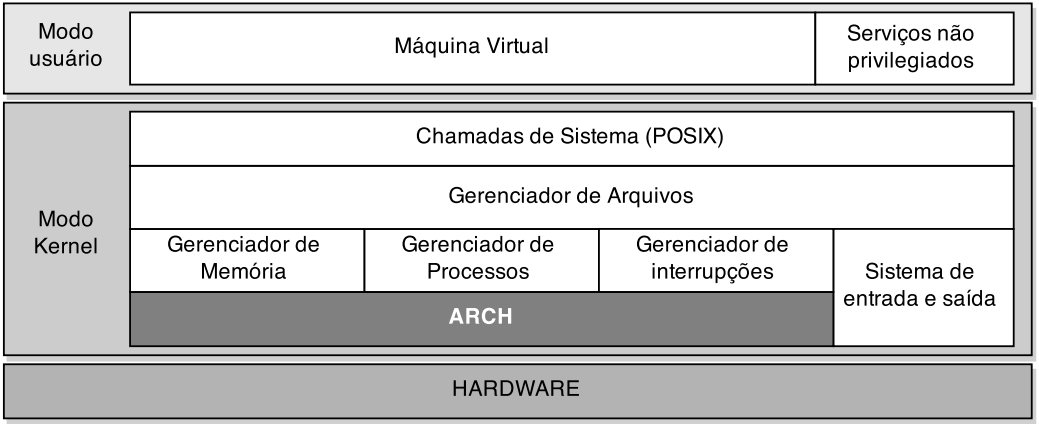
Máquinas virtuais de sistema



Native Kit


Modo usuário
Máquina Virtual
Serviços não privilegiados
Modo Kernel
Chamadas de Sistema (POSIX)
Gerenciador de Arquivos
Gerenciador de Memória
Gerenciador de Processos
Gerenciador de interrupções
Sistema de entrada e saída
ARCH
HARDWARE

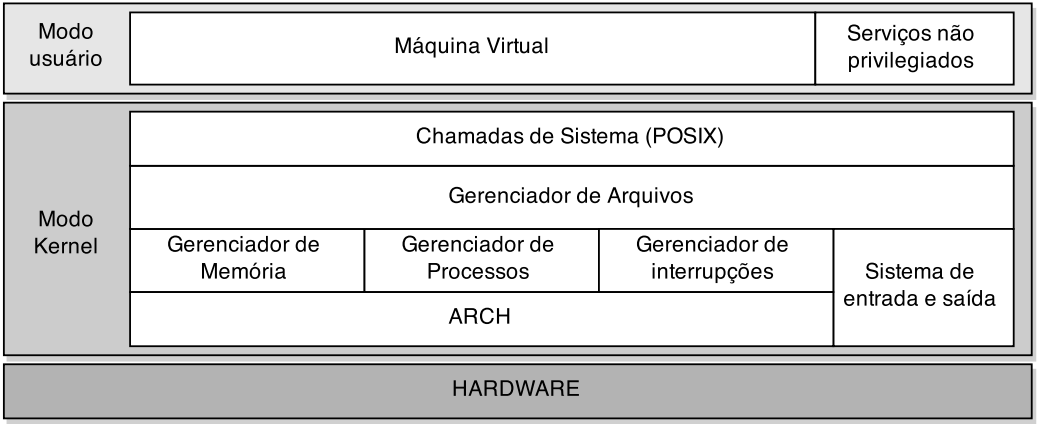
Camada ARCH


Modo usuário
Máquina Virtual
Serviços não privilegiados
Modo Kernel
Chamadas de Sistema (POSIX)
Gerenciador de Arquivos
Gerenciador de Memória
Gerenciador de Processos
Gerenciador de interrupções
Sistema de entrada e saída
ARCH
HARDWARE

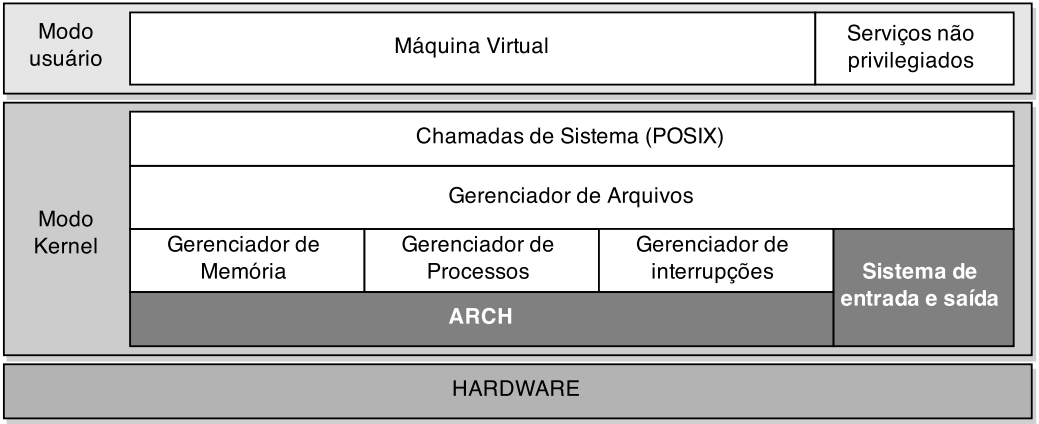
Camada ARCH

```
#include <stdint.h>

int main() {
    uint16_t color = 0xF0 << 8;
    uint16_t *vbuffer = ((uint16_t*) 0xb8000);

    vbuffer[1]  = 'H' | color;
    vbuffer[2]  = 'E' | color;
    vbuffer[3]  = 'L' | color;
    vbuffer[4]  = 'L' | color;
    vbuffer[5]  = 'O' | color;
    vbuffer[6]  = ' ' | color;
    vbuffer[7]  = 'W' | color;
    vbuffer[8]  = 'O' | color;
    vbuffer[9]  = 'R' | color;
    vbuffer[10] = 'L' | color;
    vbuffer[11] = 'D' | color;
    return 0;
}
```

Gerenciamento de entrada e saída



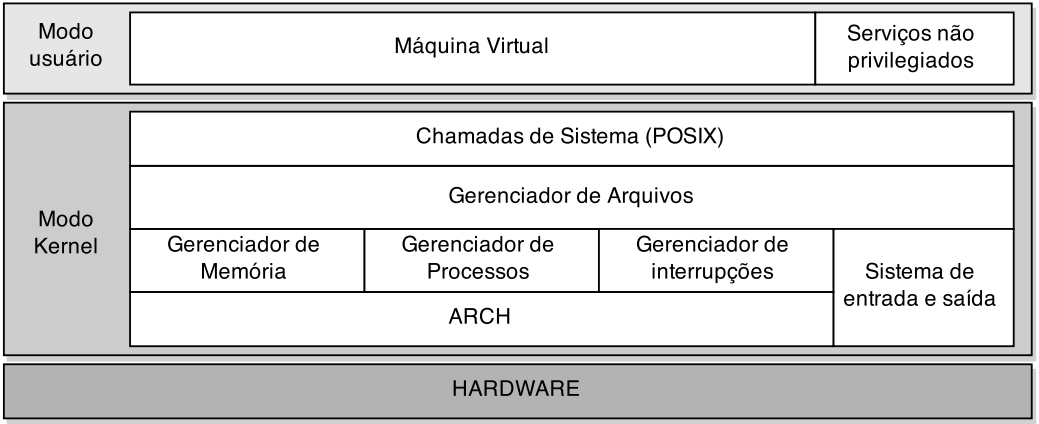
Gerenciamento de entrada e saída



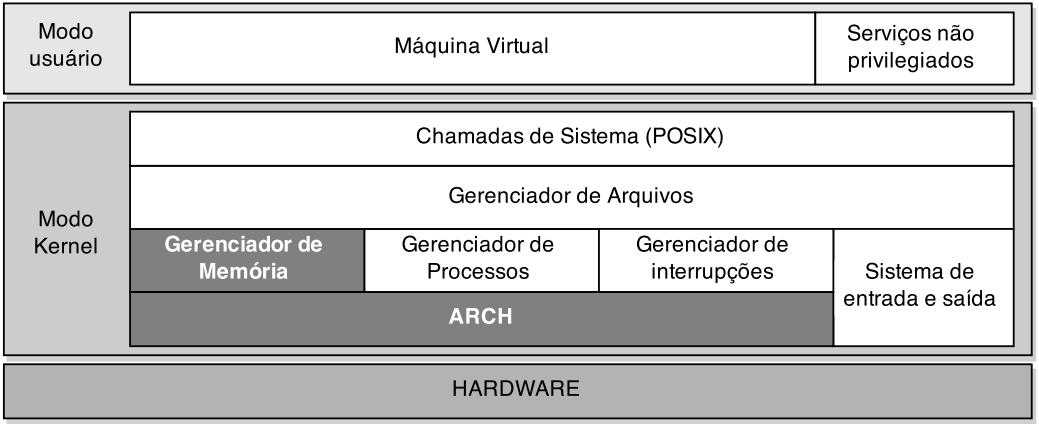
Gerenciamento de entrada e saída

```
#include <video.h>

int main() {
    Video video;
    video.write( "Hello Wolrd!" );
    return 0;
}
```

Gerente de memória



Gerente de memória

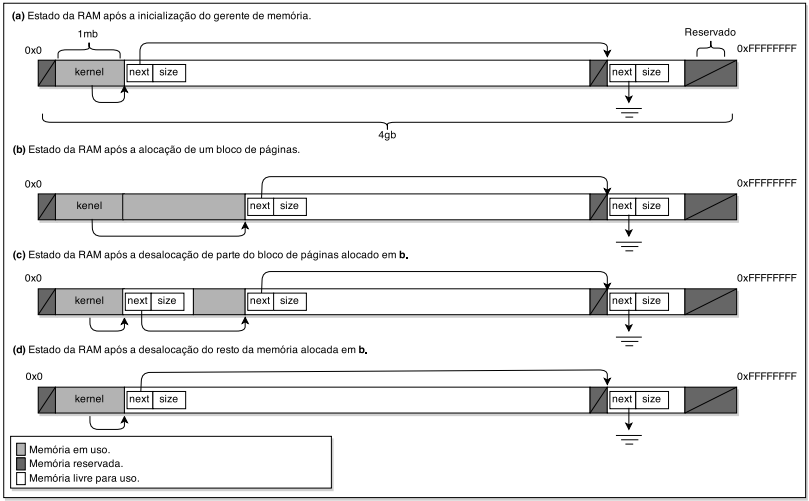


Gerente de memória: endereçamento física

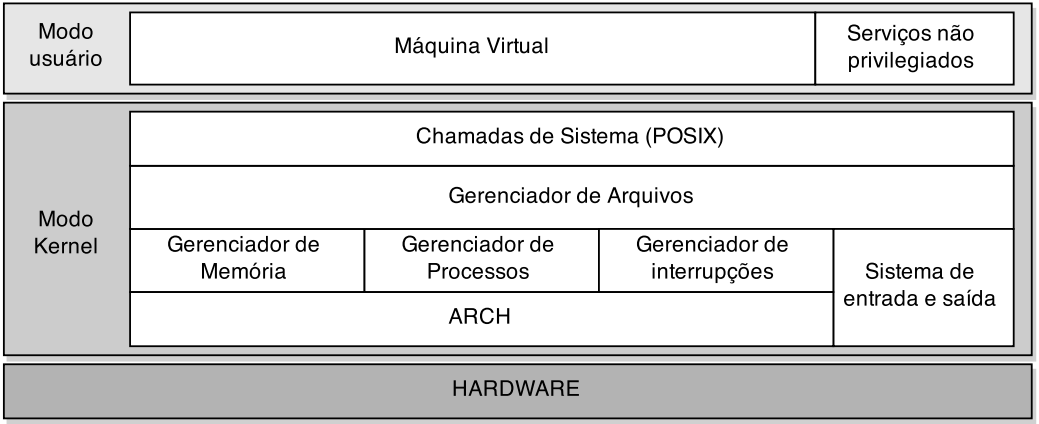
Mapa de memória:

Endereço base	Tamanho	Tipo
0x00000000	0x0009FC00	1
0x0009FC00	0x00000400	2
0x000F0000	0x00010000	2
0x00100000	0x1FEFE000	1
0x1FFFE000	0x00002000	2
0xFFFC0000	0x00040000	2

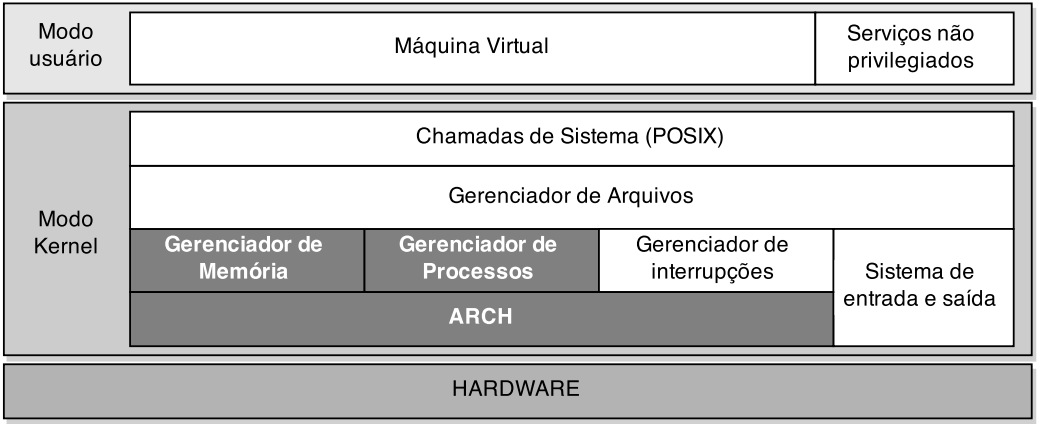
Gerente de memória: implementação



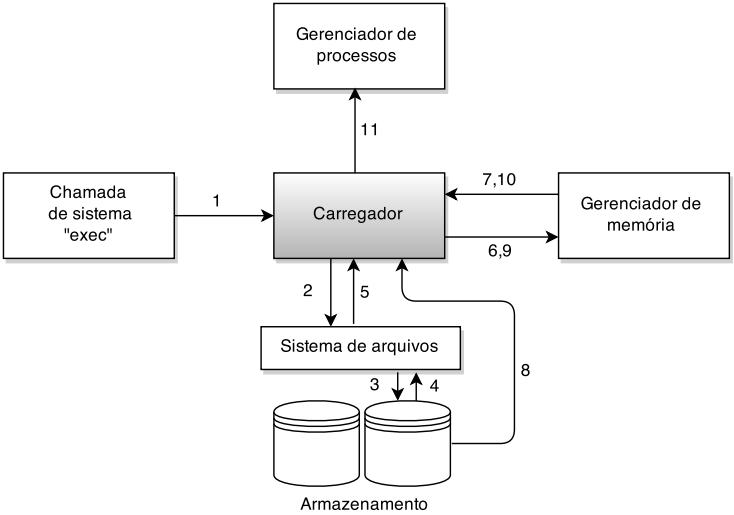
Gerente de processos



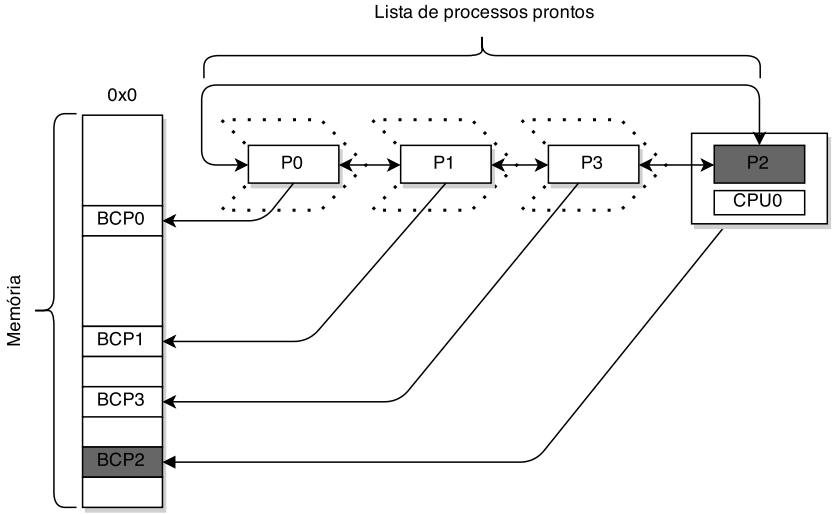
Gerente de processos



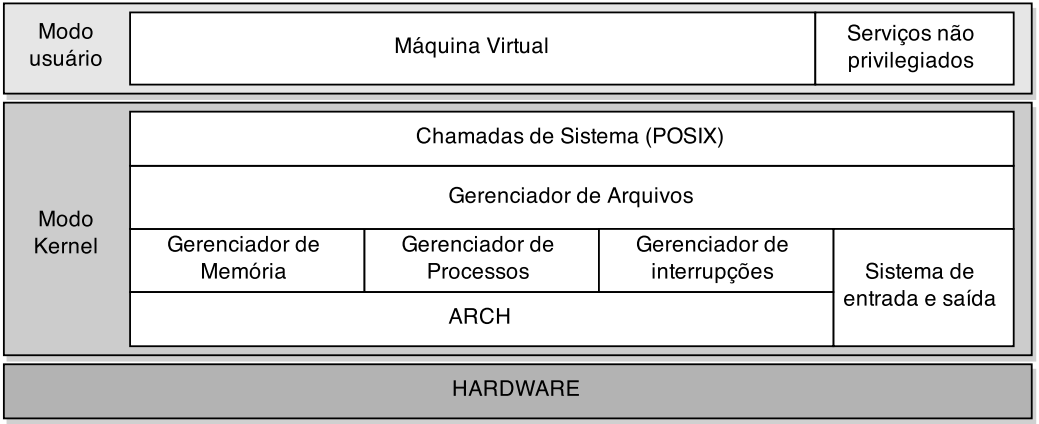
Gerente de processos: carregador



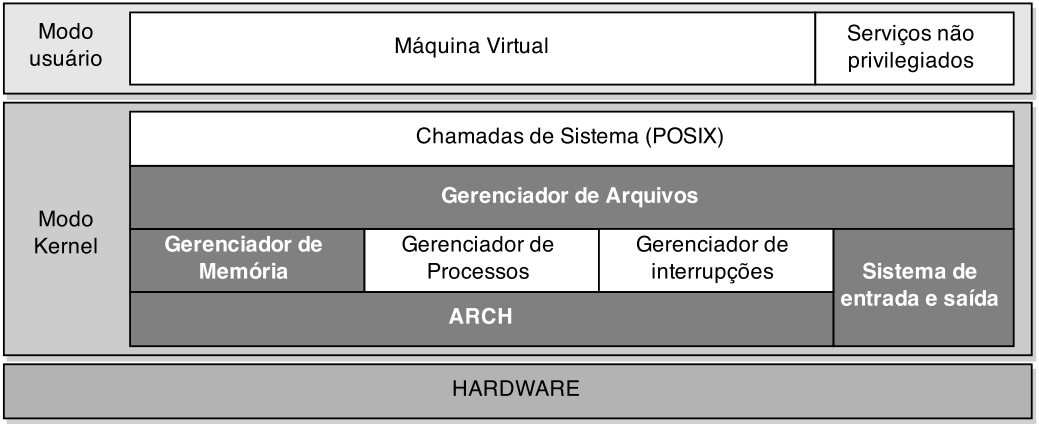
Gerente de processos: escalonador



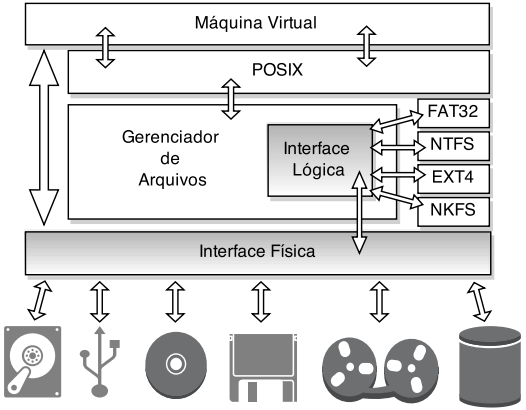
Gerente de arquivos



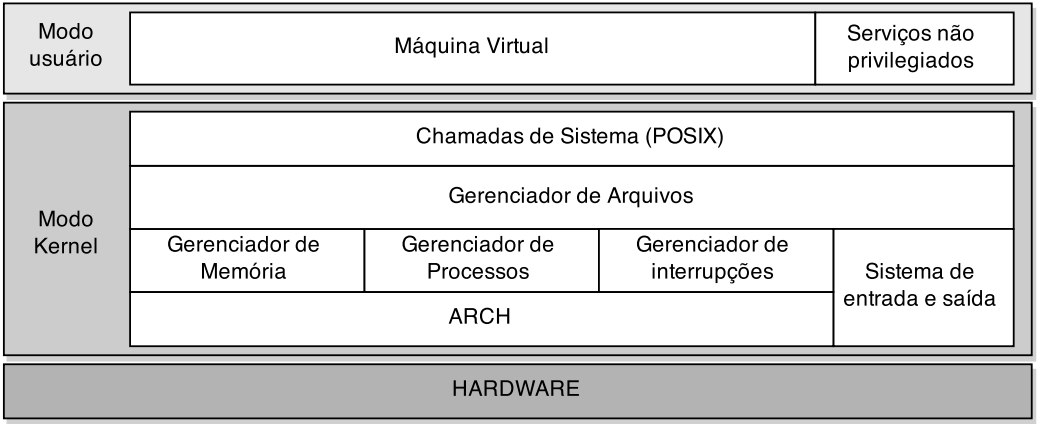
Gerente de arquivos



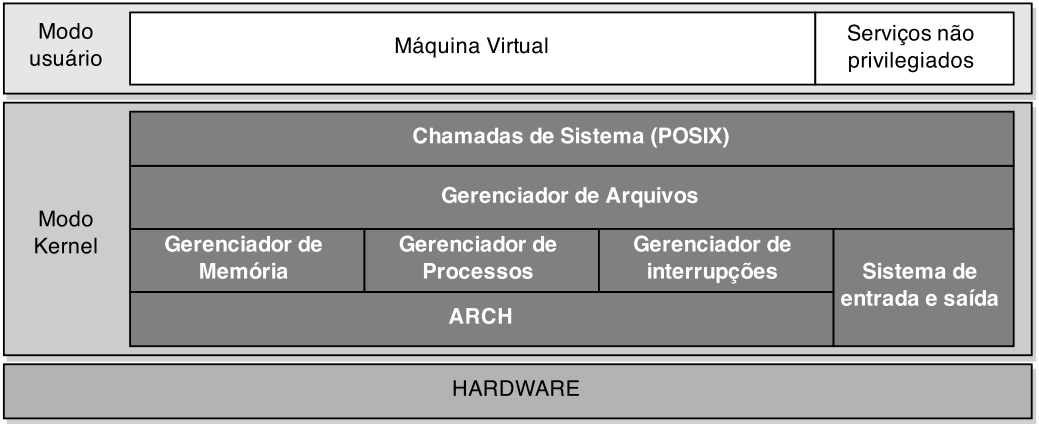
Gerente de arquivos



Chamadas de sistemas



Chamadas de sistemas



Chamadas de sistemas

```
#include <system.h>
#include <memory.h>
#include <video.h>
#include <syscalls.h>

class SimpleKernel : public System {
public:
    SimpleKernel() : System() {
        Memory memory;
        Video video;
        SysCalls syscalls;

        this->install( video );
        this->install( syscalls );
        this->install( memory );
        this->setDefaultOutput( video );
    }

    void start() {
        printf( "Hello World: usando a syscall write\n" );
    }
};
```

Chamadas de sistemas

```
#include <simplekernel.h>

int main() {
    SimpleKernel kernel;
    kernel.start();
    return 0;
}
```

Trabalhos relacionados

- OSv
- UML (User Mode Linux)
- JNodeOS
- XTrantum
- VMware Server
- Oracle VM
- Xen Server
- NewLib
- ...

OS Kit vs Native Kit

Características	OS Kit	Native Kit
Arquiteturas compatíveis	x86, alpha	x86
Formato dos binários	ELF	ELF
Suporte para bibliotecas dinâmicas	Sim	Não
Infraestrutura portável	Sim	Sim
Infraestrutura modular	Sim	Sim
Infraestrutura orientado a objeto	Não	Sim
Licença	GPLv2	LGPLv3
Ano da última versão	2002	2015
Número de componentes	34	5

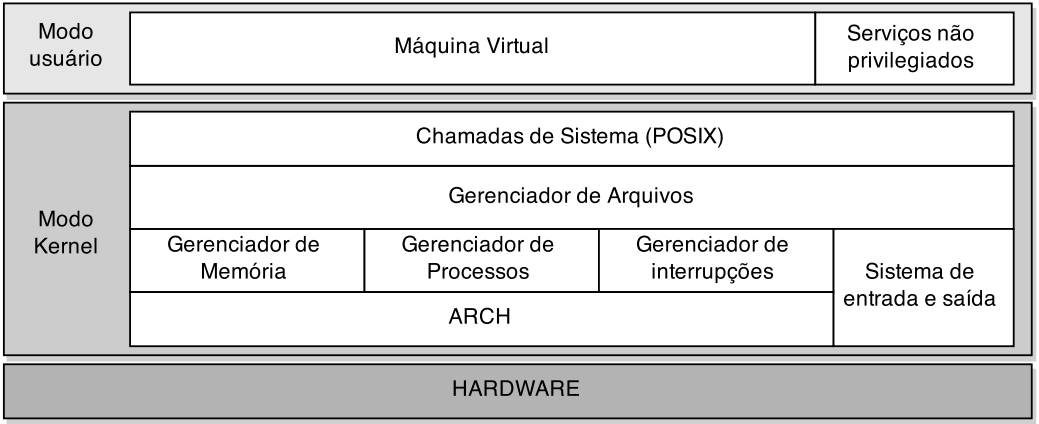
OS Kit vs Native Kit

Características	OS Kit	Native Kit
Suporte para segurança	Variável	Mínimo
Suporte para compatibilidade com POSIX	Variável	Mínimo
Suporte para gerenciamento de arquivos	Variável	Mínimo
Suporte para gerenciamento de memória	Variável	Mínimo
Suporte para gerenciamento de processos	Básico	Mínimo
Suporte para gerenciamento de *threads*	Completo	Não
Suporte para gerenciamento de rede	Variável	Não
Suporte para *drivers*	Completo	Mínimo
Suporte para depuração e testes	Básico	Básico
Complexidade	Média	Variável
Pilha de software	Média	Baixa

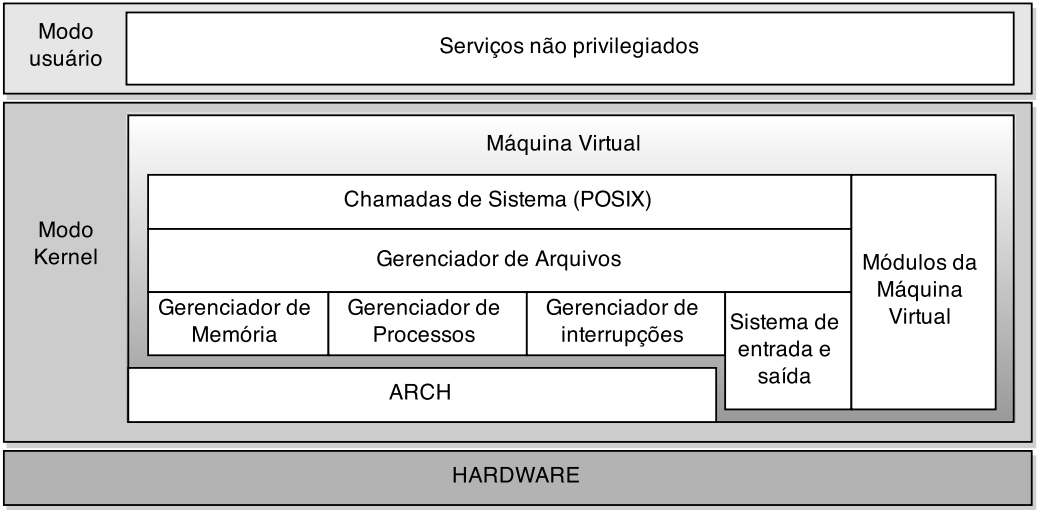
Organização do sistema

- A infraestrutura pode ser organizada de três formas diferentes.
  - Organização hospedada.
  - Organização supervisor.
  - Organização híbrida.

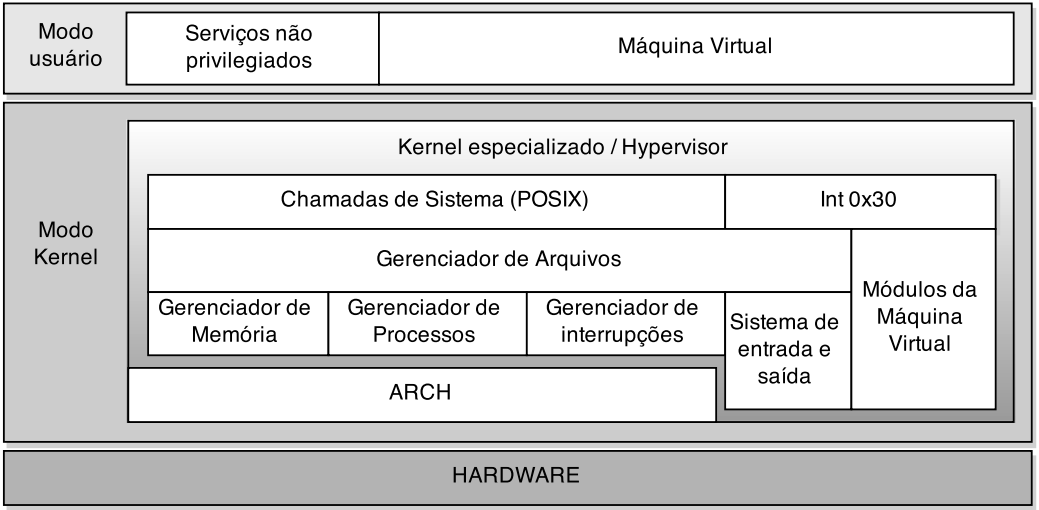
Organização do sistema: modo hospedado



Organização do sistema: modo supervisor



Organização do sistema: modo híbrido



Estatísticas gerais

Informações gerais.	
Número total de linhas	7570 linhas
Número de linhas de código independentes de arquitetura (C/C++)	5941 linhas
Número de linhas de código dependentes de arquitetura	1458 linhas
Número total de linhas de código de configuração	171 linhas
Número total de linhas de código de teste	857 linhas
Total de arquivos da infraestrutura	104 arquivos
Total de arquivos de todo o projeto	106254 arquivos
Tamanho total em disco	2.4 GB

Estatísticas gerais

Tempo de inicialização	
Carregamento	Menos de 1 segundo
Configuração	Menos de 1 segundo
Tela inicial do emulador	Menos de 2 segundos
Tempo total	Aproximadamente 3 segundo

Tamanho dos arquivos objeto

Tamanho dos objetos antes da ligação estática	
Máquina virtual	0.5 MB
Disco virtual	4.7 MB
libc.a	3.4 MB
libstdc++.a	8.6 MB
libm.a	1.3 MB

Tamanho da imagem final

Tamanho da imagem final	
Grub	8.0 MB
Máquina virtual (módulos + aplicação + libc.a + libm.a + libstdc++)	8.3 MB
Disco virtual	4.7 MB
Alocação estática da pilha	64.0 KB
Alocação estática da *heap*	64.0 KB
Tamanho da image de CD/DVD (disk.iso)	~21.0 MB

Consumo de memória

Tamanho da máquina virtual na RAM.	
Espaço reservado para modo o 8086	1 MB
Tamanho da pilha	64.0 KB
Tamanho da *heap*	64.0 KB
Espaço desperdiçado com alinhamento	$<$ 4.0 KB
Sistema de arquivos em memória	4.7 MB
Máquina virtual (módulos + aplicação + libc.a + libm.a + libstdc++)	˜7.1 MB
Tamanho total	13.0 MB

Agradecimentos

Contato

- skype: linharesalisson
- e-mail: arescarv@gmail.com