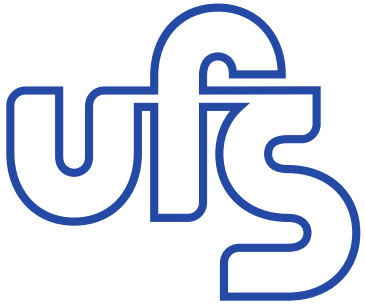
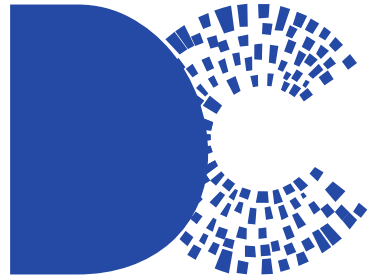




Software básico

Arquitetura de Computadores

Bruno Prado

Departamento de Computação / UFS

Introdução

- O que é software básico?

Introdução

- O que é software básico?
  - É um conjunto de componentes de software projetados para gerenciar os recursos do sistema

Introdução

- O que é software básico?
  - É um conjunto de componentes de software projetados para gerenciar os recursos do sistema
  - Este software cria a infraestrutura necessária para execução das aplicações do usuário

Introdução

- O que é software básico?
  - É um conjunto de componentes de software projetados para gerenciar os recursos do sistema
  - Este software cria a infraestrutura necessária para execução das aplicações do usuário
  - Também é definido como software de sistema

Introdução

- Tipos de software básico

Introdução

- Tipos de software básico
  - Sistema Operacional (SO)
    - Harmony OS, Linux, Tizen, ...
    - Android, MacOS, Windows, ...

Introdução

- Tipos de software básico
  - Sistema Operacional (SO)
    - Harmony OS, Linux, Tizen, ...
    - Android, MacOS, Windows, ...
  - Software dependente do Hardware (HdS)
    - Gerenciadores de dispositivos (*device drivers*)
    - Camada de abstração de hardware (HAL)

Introdução

- Tipos de software básico
  - Sistema Operacional (SO)
    - Harmony OS, Linux, Tizen, ...
    - Android, MacOS, Windows, ...
  - Software dependente do Hardware (HdS)
    - Gerenciadores de dispositivos (*device drivers*)
    - Camada de abstração de hardware (HAL)
  - Ferramentas de desenvolvimento (*toolchain*)
    - Compilador/montador
    - Depurador/simulador

Sistema Operacional

- Interface gráfica
  - Consiste na representação visual de aplicações (janelas) que utilizam um cursor controlado por mouse
  - Maior intuitividade e popularização de computadores

Apple ⟷ Microsoft ⟷ Xerox PARC

Sistema Operacional

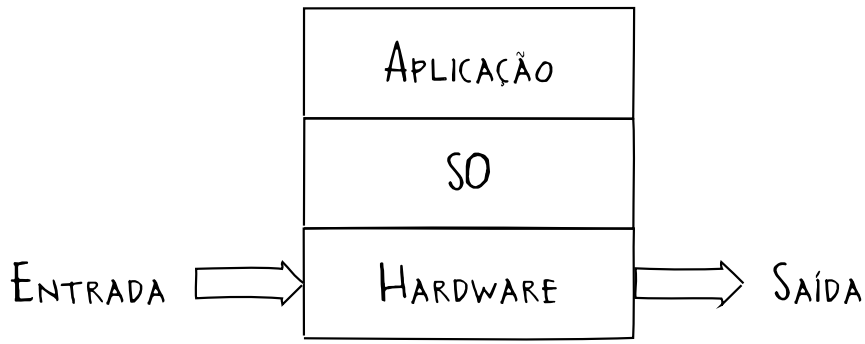
- Interface gráfica
  - Consiste na representação visual de aplicações (janelas) que utilizam um cursor controlado por mouse
  - Maior intuitividade e popularização de computadores

Apple ⟷ Microsoft ⟷ Xerox PARC

- Interface de texto
  - Utiliza um console ou terminal baseado em linha de comando que interpreta comandos e *scripts*
  - Ainda é muito utilizado em servidores (SSH)

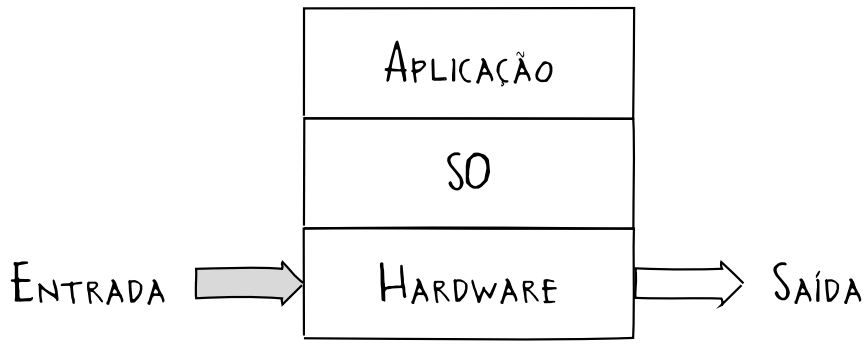
Sistema Operacional

- Operações de entrada e saída (E/S)
  - Perspectiva do computador: pelas interfaces de programação do sistema operacional (SO), a aplicação do usuário consegue realizar operações de entrada e saída



Sistema Operacional

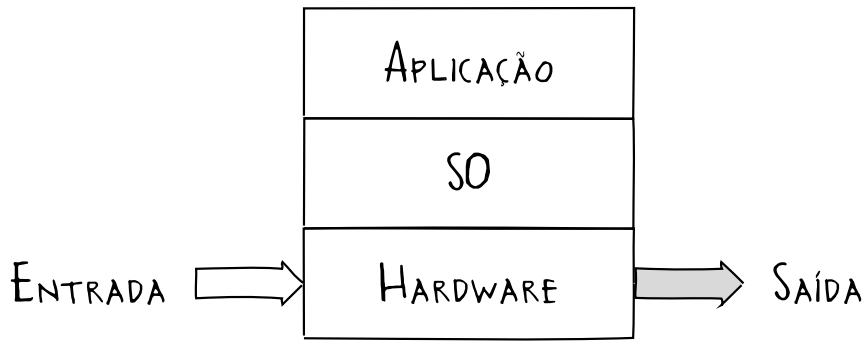
- Operações de entrada e saída (E/S)
  - Perspectiva do computador: pelas interfaces de programação do sistema operacional (SO), a aplicação do usuário consegue realizar operações de entrada e saída



  - Os dispositivos de entrada, como teclado e mouse, tem o propósito de fornecer os dados para o sistema

Sistema Operacional

- Operações de entrada e saída (E/S)
  - Perspectiva do computador: pelas interfaces de programação do sistema operacional (SO), a aplicação do usuário consegue realizar operações de entrada e saída



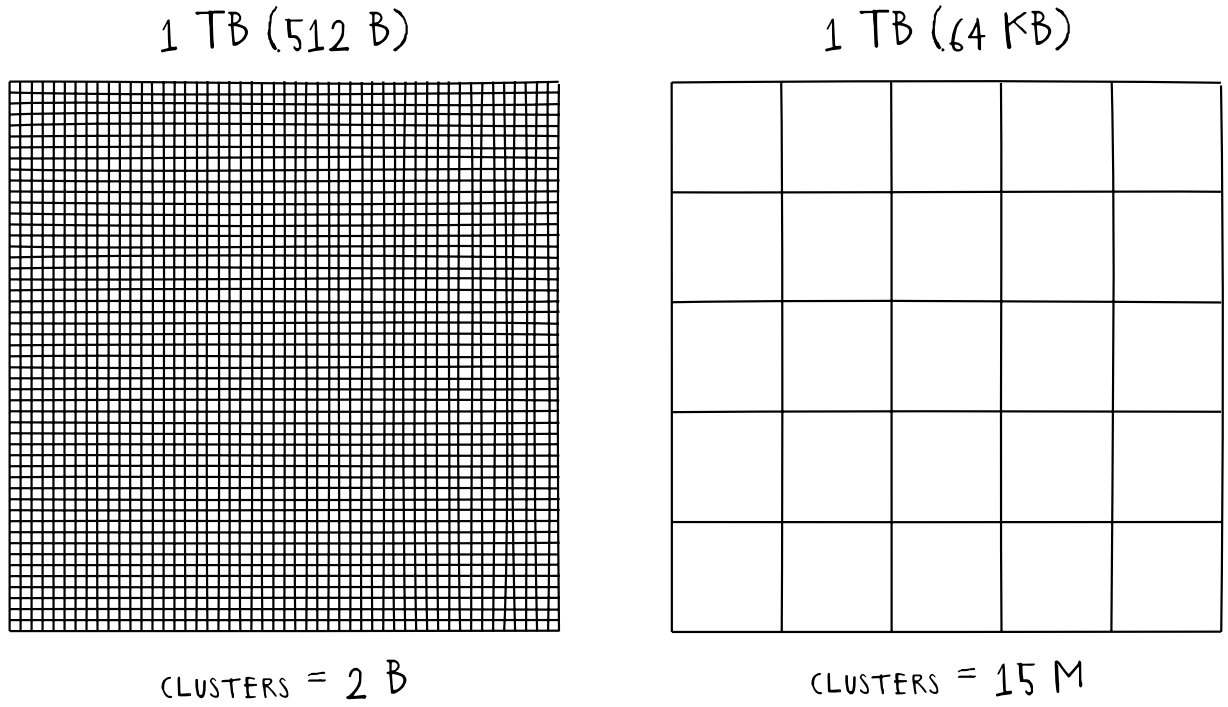
  - Os dispositivos de entrada, como teclado e mouse, tem o propósito de fornecer os dados para o sistema
  - Quando dados são produzidos pelo computador, são enviados ou exibidos por dispositivos de saída (alto-falantes, impressora, tela, etc)

Sistema Operacional

- Suporte para armazenamento
  - Como a memória principal do computador é volátil e limitada em capacidade, é necessário utilizar unidades de armazenamento em disco de alta capacidade para retenção dos dados com suporte de sistemas de arquivo
    - **DOS/Windows**: *File Allocation Table* (FAT) e *New Technology File System* (NTFS)
    - **Linux**: *B-tree File System* (BtrFS) e *Extended File System* (Ext, Ext2, Ext3 e Ext4)
    - **MacOS**: *Apple File System* (APFS) e *Hierarchical File System* (HFS e HFS+)

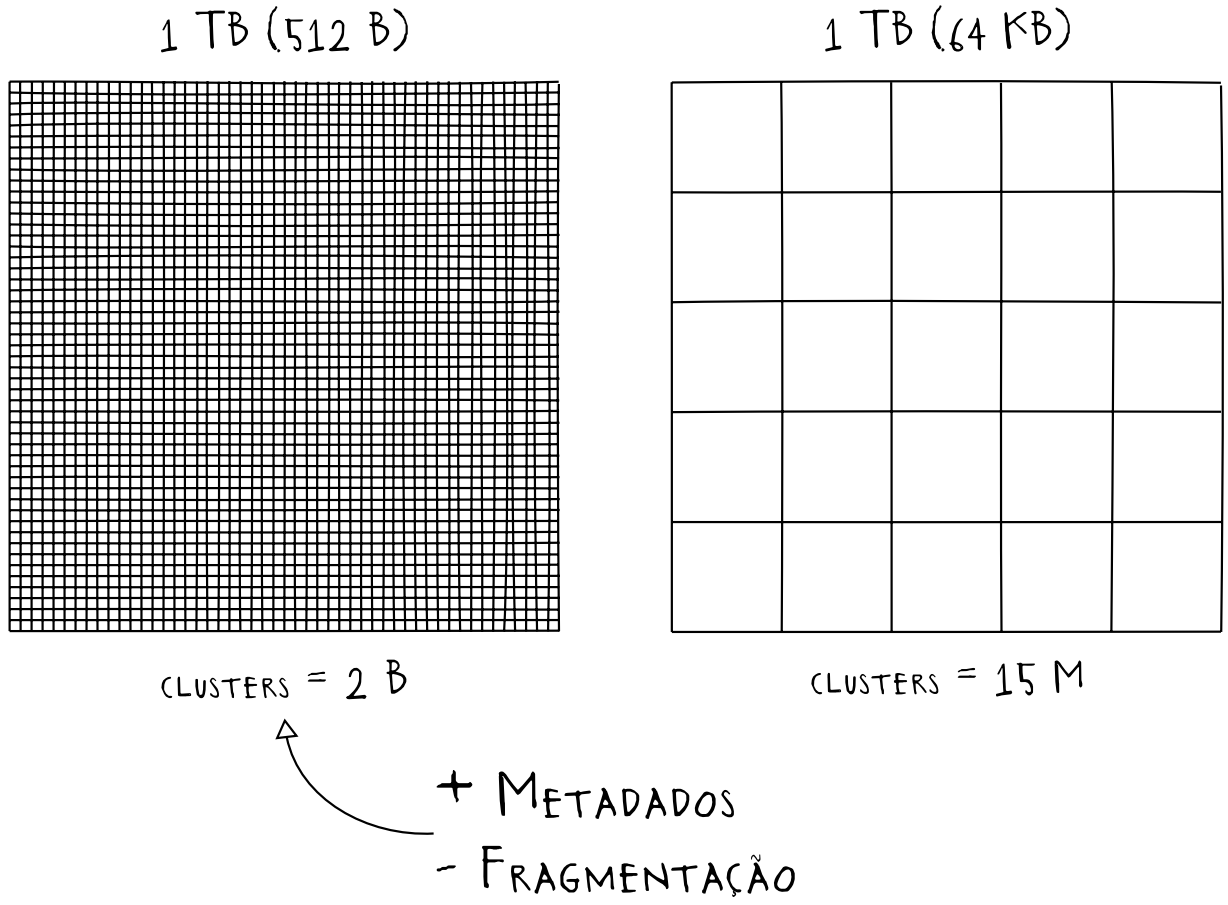
Sistema Operacional

- Suporte para armazenamento
  - São utilizadas estruturas de árvore e de tabelas nas implementações dos sistemas de arquivo, agrupando os dados em unidades de alocação (*clusters*) com tamanho variando entre 512 bytes até 64 KB



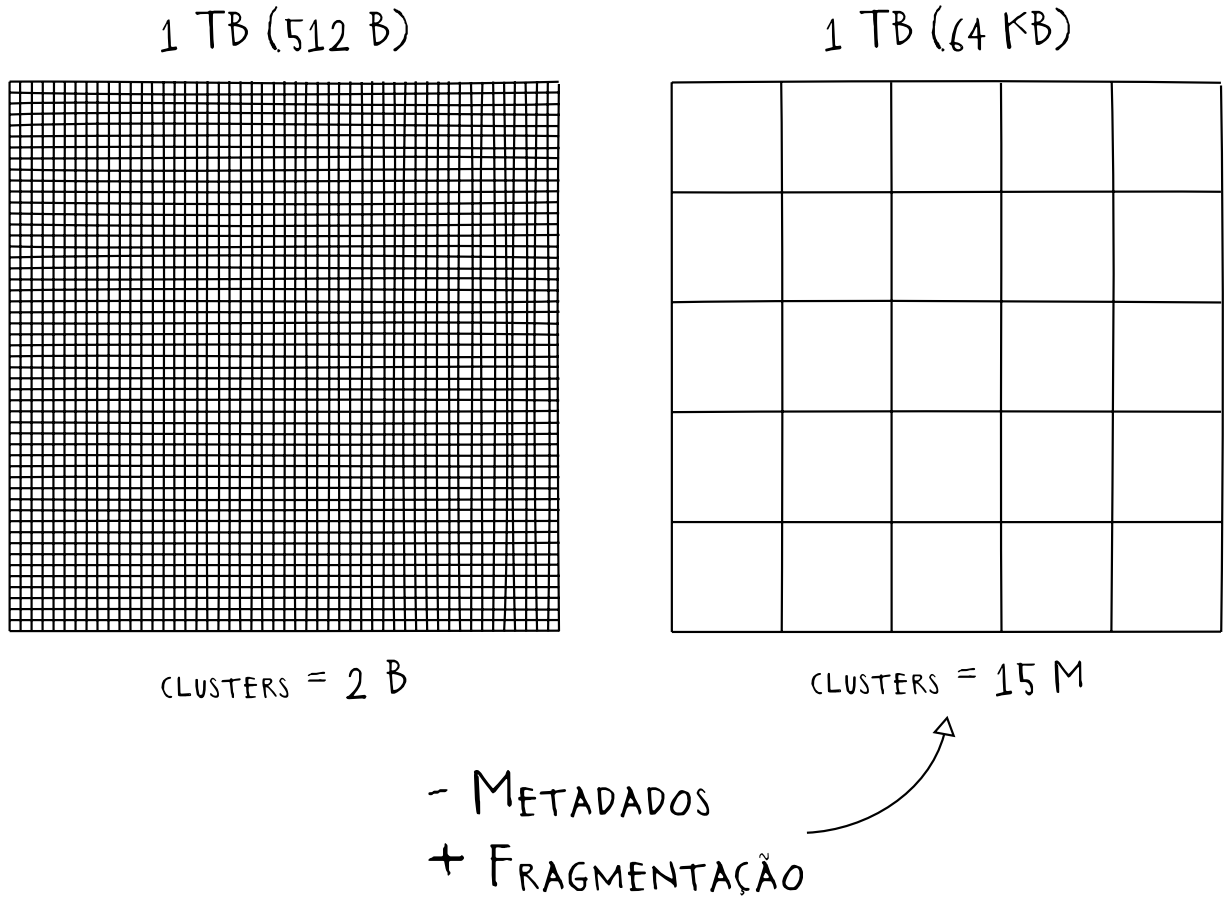
Sistema Operacional

- Suporte para armazenamento
  - São utilizadas estruturas de árvore e de tabelas nas implementações dos sistemas de arquivo, agrupando os dados em unidades de alocação (*clusters*) com tamanho variando entre 512 bytes até 64 KB



Sistema Operacional

- Suporte para armazenamento
  - São utilizadas estruturas de árvore e de tabelas nas implementações dos sistemas de arquivo, agrupando os dados em unidades de alocação (*clusters*) com tamanho variando entre 512 bytes até 64 KB



Sistema Operacional

- Serviços de rede e segurança
  - Para transferência de dados através de uma rede de computadores, como a Internet, é preciso um conjunto de bibliotecas de programação (*sockets*) para realizar as operações de transferência de dados

Sistema Operacional

- Serviços de rede e segurança
  - Para transferência de dados através de uma rede de computadores, como a Internet, é preciso um conjunto de bibliotecas de programação (*sockets*) para realizar as operações de transferência de dados
    - *Transfer Control Protocol* (TCP): orientado a conexão, garantindo a entrega e a ordenação dos dados, além da checagem de integridade e retransmissão em caso de perdas (HTTP ou HTTPS)

Sistema Operacional

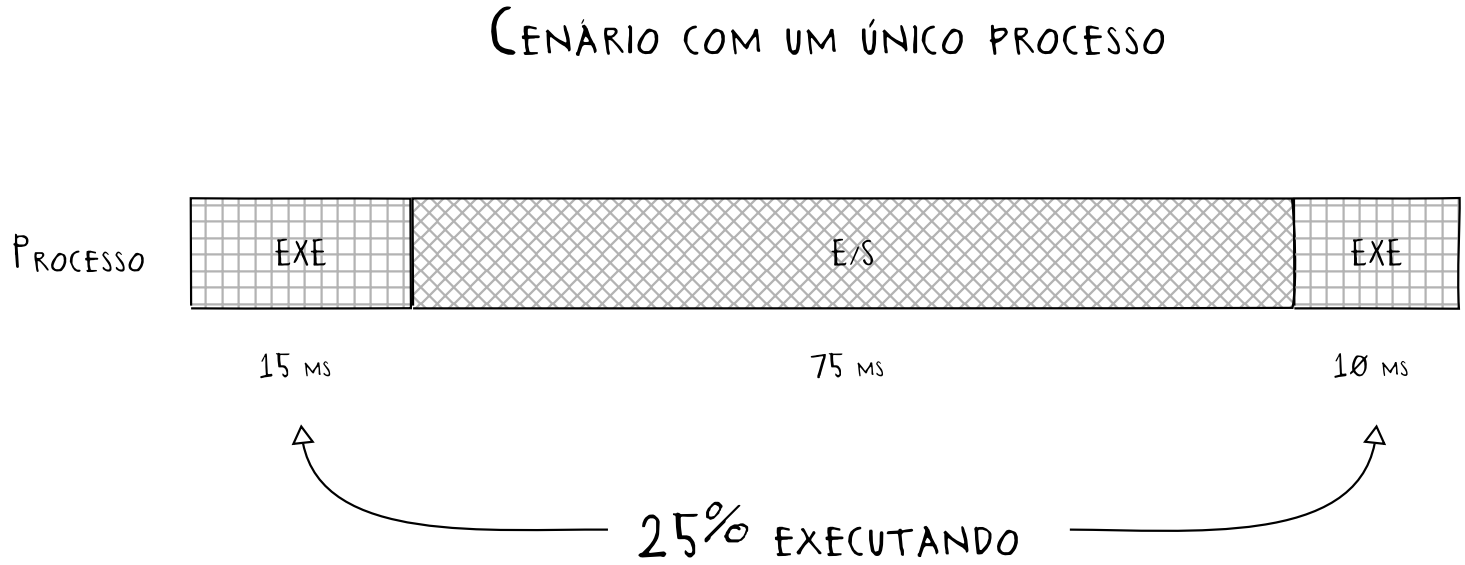
- Serviços de rede e segurança
  - Para transferência de dados através de uma rede de computadores, como a Internet, é preciso um conjunto de bibliotecas de programação (*sockets*) para realizar as operações de transferência de dados
    - *Transfer Control Protocol* (TCP): orientado a conexão, garantindo a entrega e a ordenação dos dados, além da checagem de integridade e retransmissão em caso de perdas (HTTP ou HTTPS)
    - *User Datagram Protocol* (UDP): opera através do envio e recebimento de pacotes individuais sem conexão ou estado, sendo de responsabilidade da aplicação a ordenação, a integridade e a retransmissão dos dados (*streaming*)

Sistema Operacional

- Serviços de rede e segurança
  - Criptografia dos dados armazenados
    - Assinatura digital de software
    - Autenticação dos usuários (*hash*)
    - Sistema de arquivo criptografado
    - ...
  - Protocolos seguros de comunicação
    - ***HTTP Secure* (HTTPS)**: protege com certificados e criptografia o acesso a páginas via protocolo HTTP
    - ***Secure SHell* (SSH)**: permite conexões remotas seguras através da interface de linha de comando
    - ***Virtual Private Network* (VPN)**: cria uma rede privada criptografa entre dois nós conectados pela Internet
    - ...

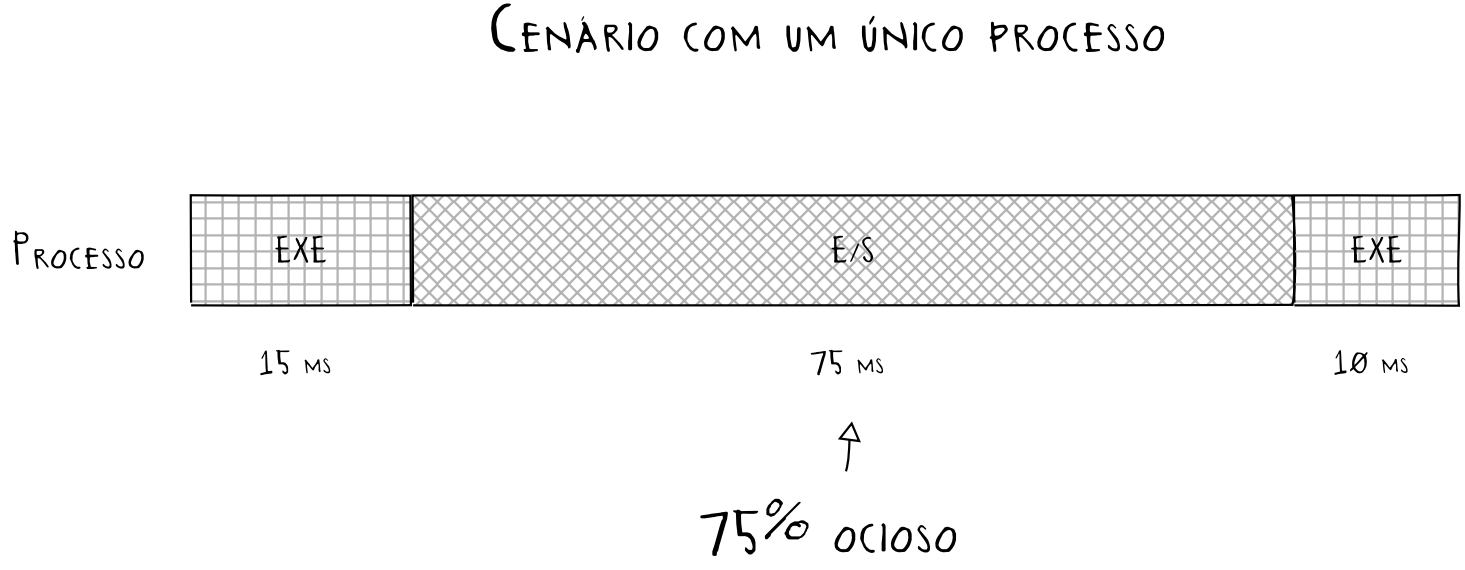
Sistema Operacional

- Escalonamento e multiprogramação
  - Para ter um propósito geral, um computador deve ser executar múltiplas aplicações concorrentemente



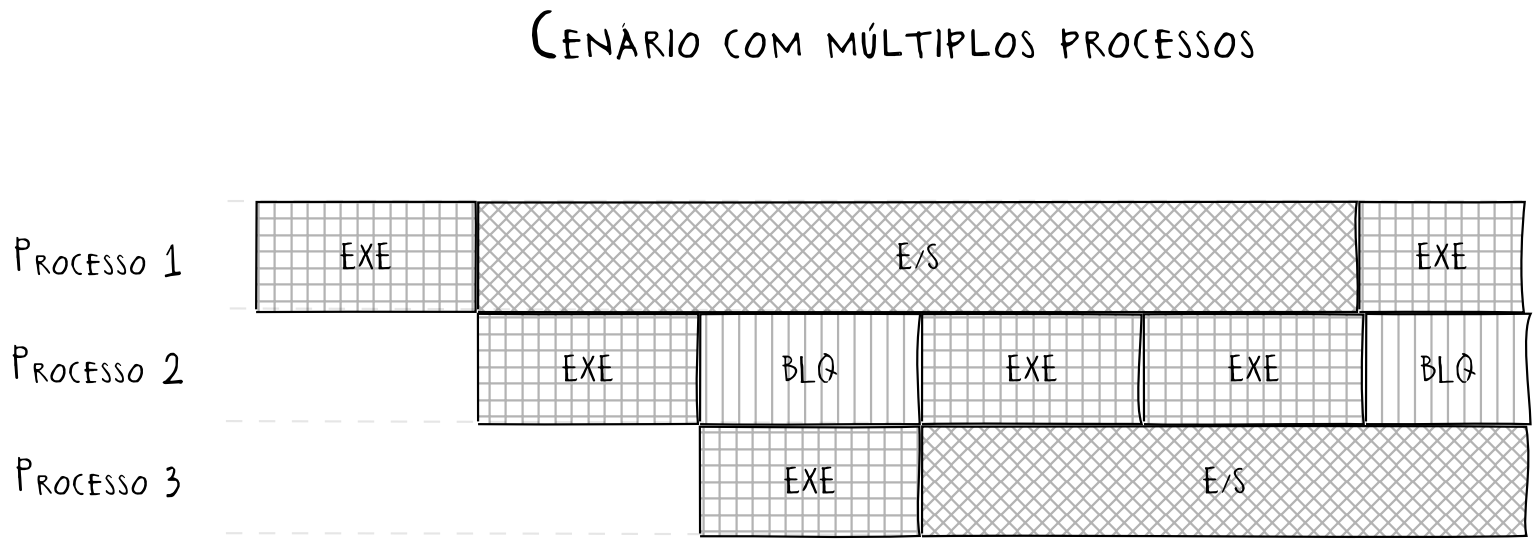
Sistema Operacional

- Escalonamento e multiprogramação
  - Para ter um propósito geral, um computador deve ser executar múltiplas aplicações concorrentemente



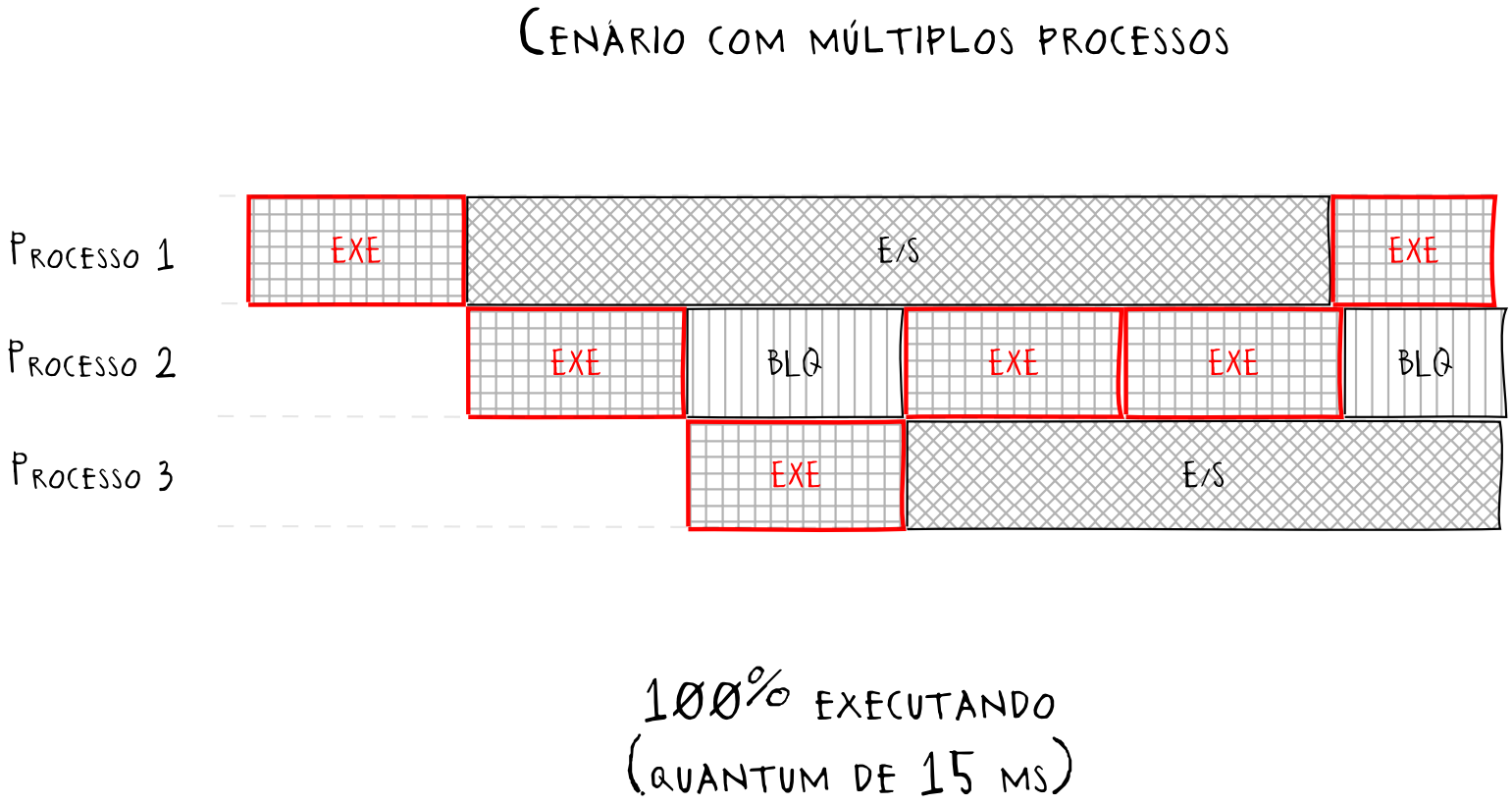
Sistema Operacional

- Escalonamento e multiprogramação
  - Em um ambiente de execução pseudo paralelo, cada processo possui uma determinada quantidade máxima de tempo de execução (*quantum*)



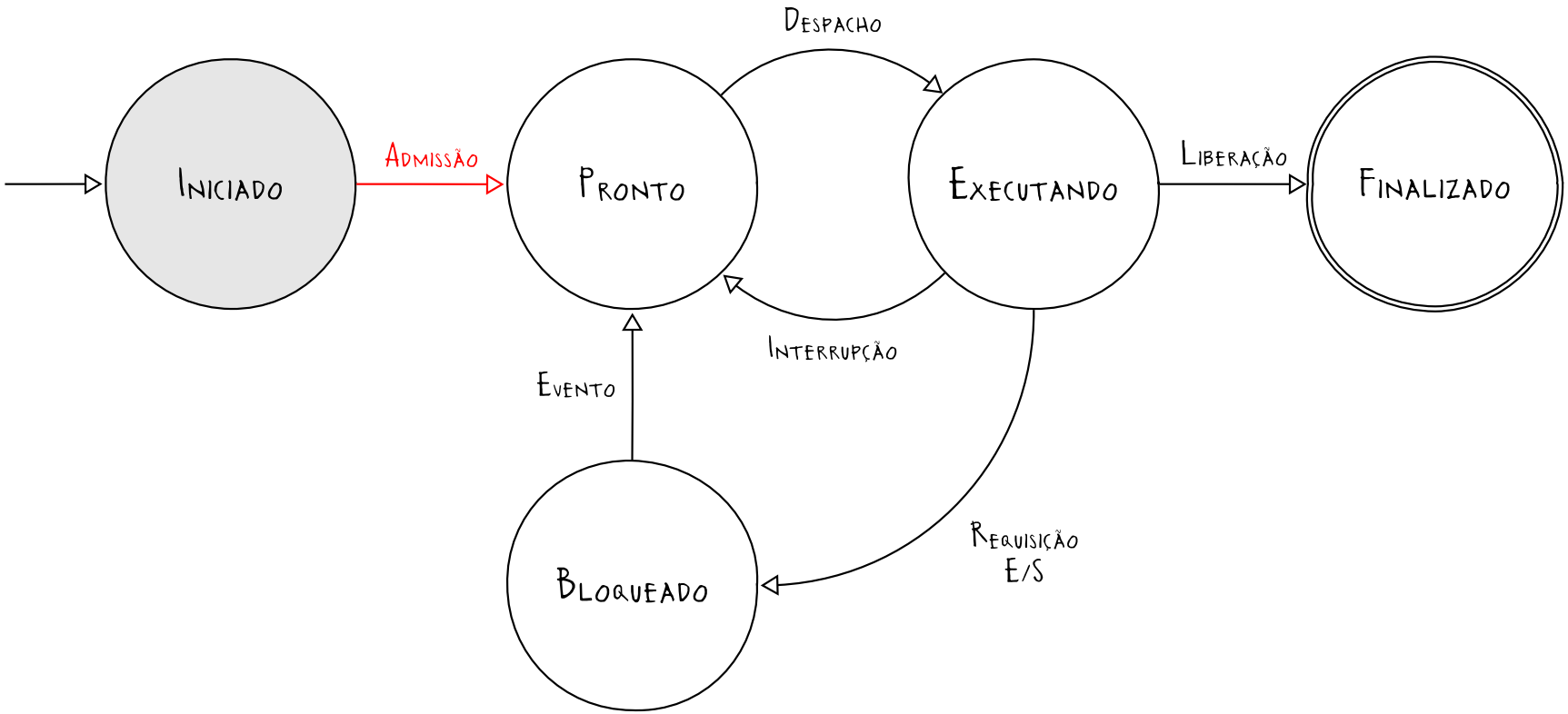
Sistema Operacional

- Escalonamento e multiprogramação
  - Em um ambiente de execução pseudo paralelo, cada processo possui uma determinada quantidade máxima de tempo de execução (*quantum*)



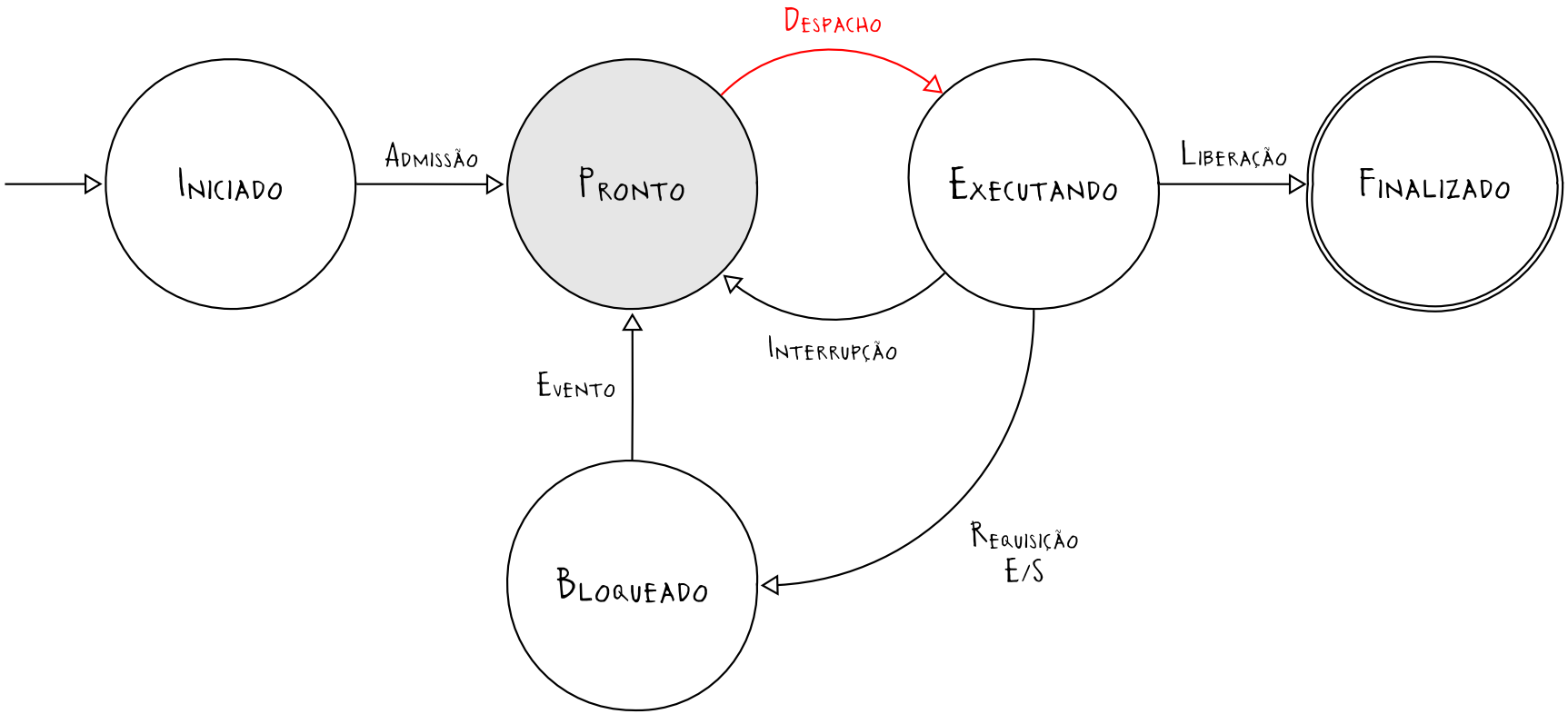
Sistema Operacional

- Escalonamento e multiprogramação
  - Máquina de estados do processo



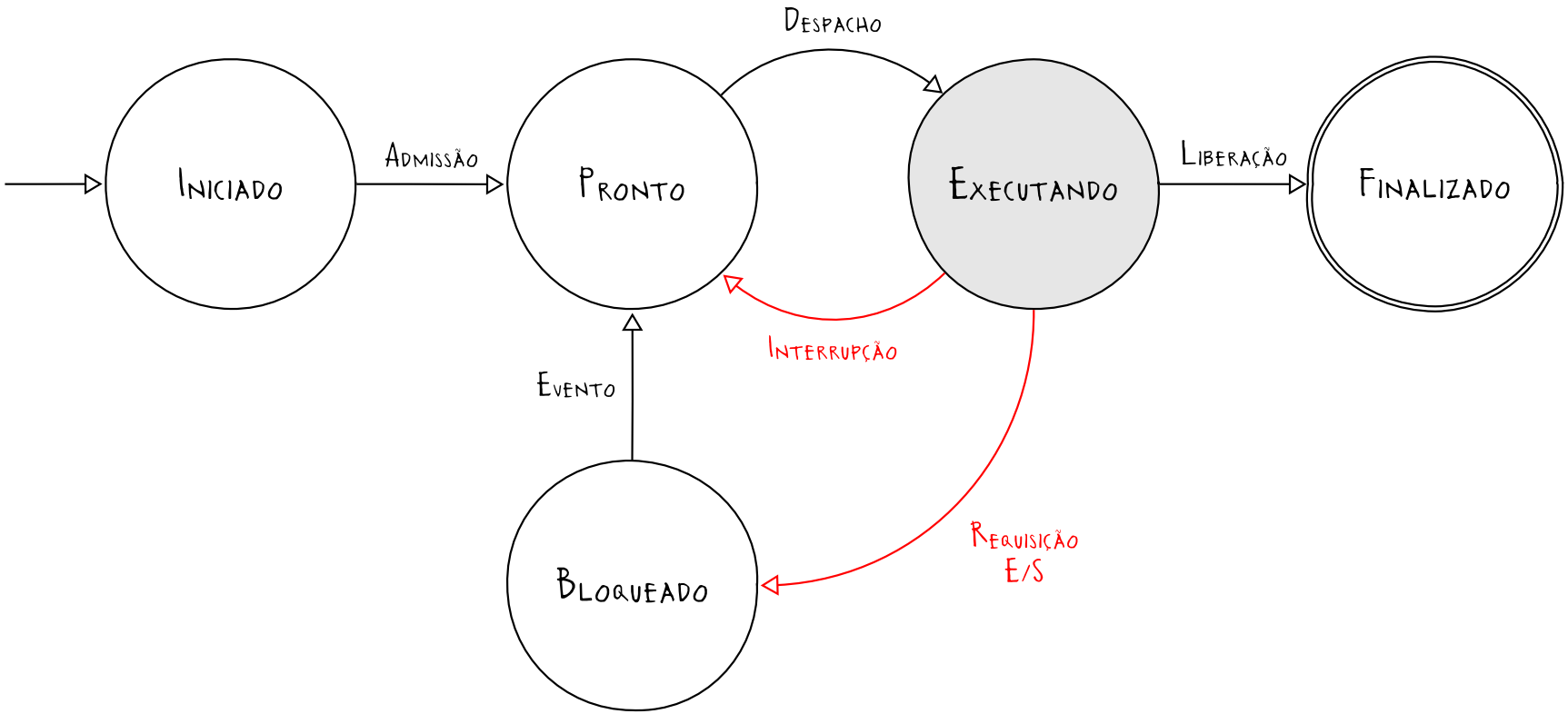
Sistema Operacional

- Escalonamento e multiprogramação
  - Máquina de estados do processo



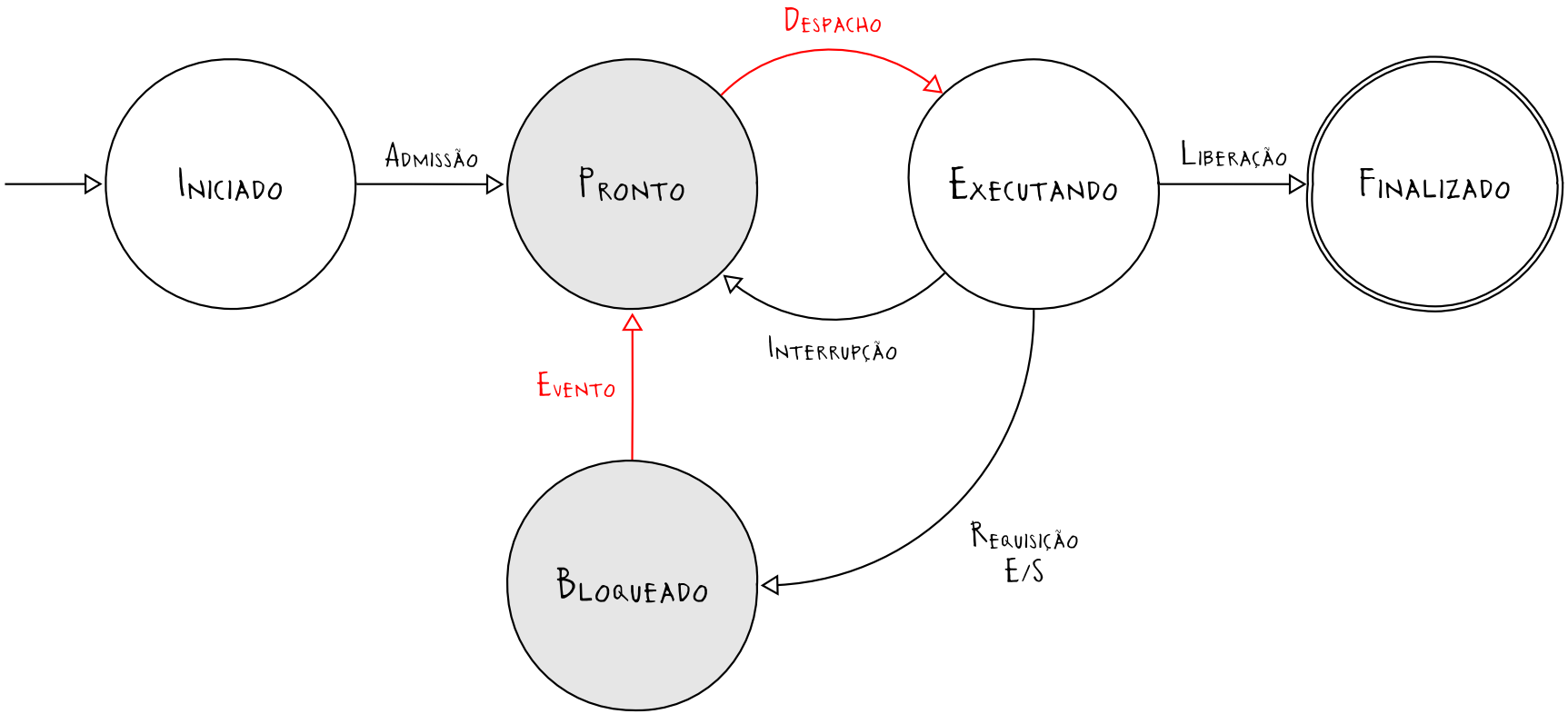
Sistema Operacional

- Escalonamento e multiprogramação
  - Máquina de estados do processo



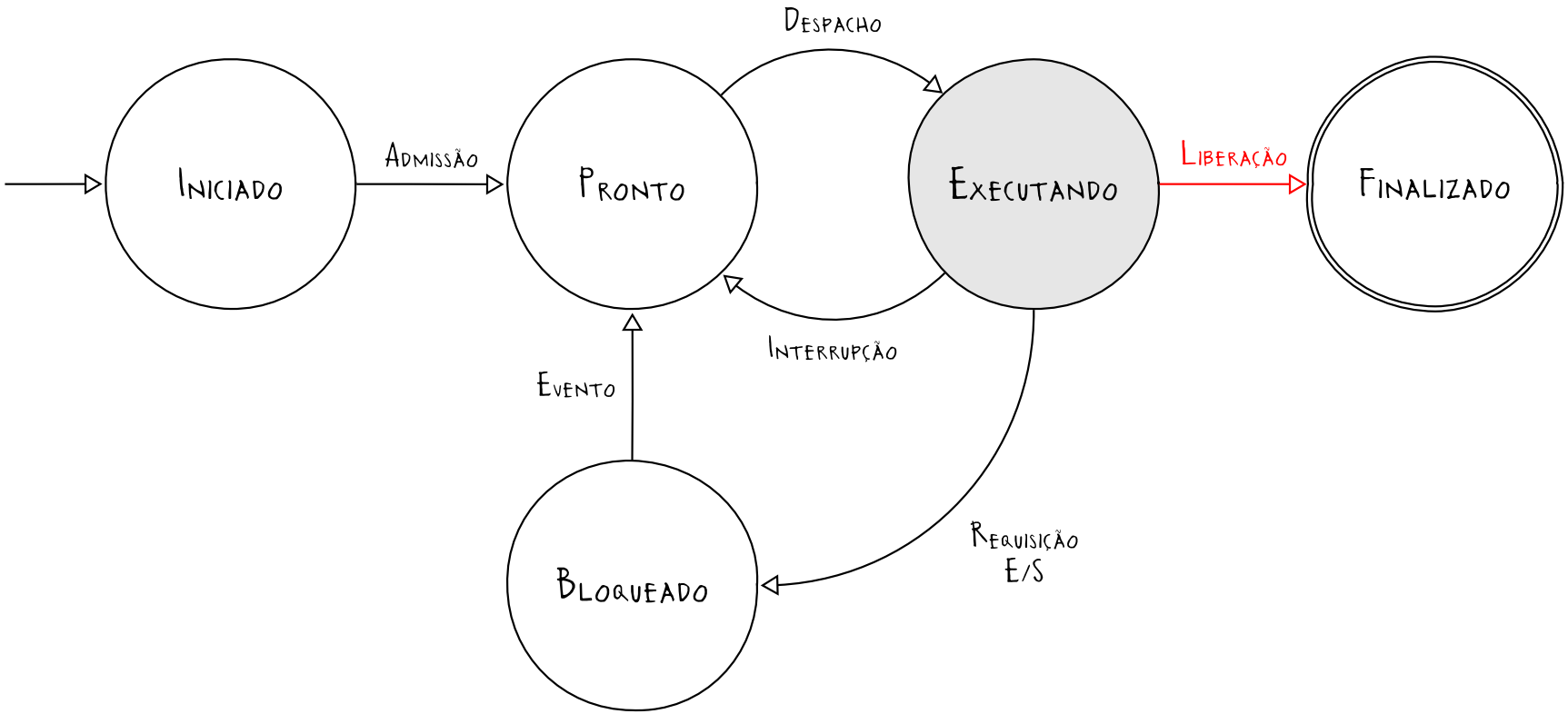
Sistema Operacional

- Escalonamento e multiprogramação
  - Máquina de estados do processo



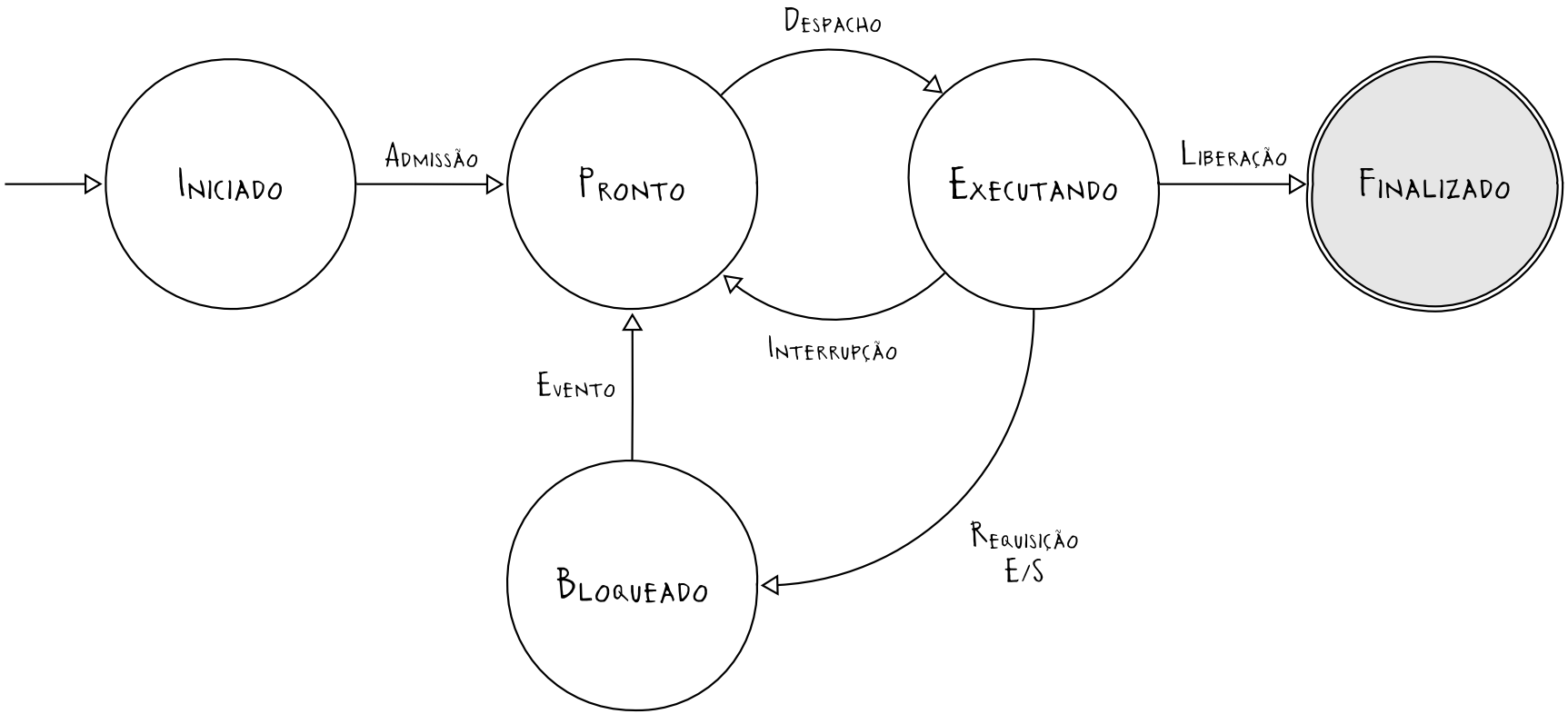
Sistema Operacional

- Escalonamento e multiprogramação
  - Máquina de estados do processo



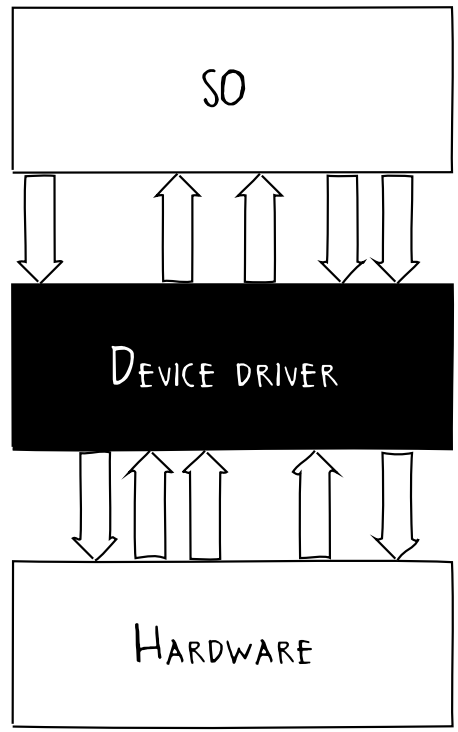
Sistema Operacional

- Escalonamento e multiprogramação
  - Máquina de estados do processo



Software dependente do Hardware

- Gerenciamento de dispositivos (*device driver*)
  - É o código fonte utilizado para um hardware específico (*device driver*) através de uma API
  - Abstrai do resto do sistema os detalhes de como o dispositivo funciona (caixa preta)



Modularidade <-> Plug and play

Software dependente do Hardware

- Quais são os tipos de gerenciadores de dispositivo?

Software dependente do Hardware

- Quais são os tipos de gerenciadores de dispositivo?
  - **Caractere**: neste tipo de dispositivo o acesso é feito por fluxo de bytes, assim como ler ou escrever em um arquivo, utilizando as chamadas de sistema

Software dependente do Hardware

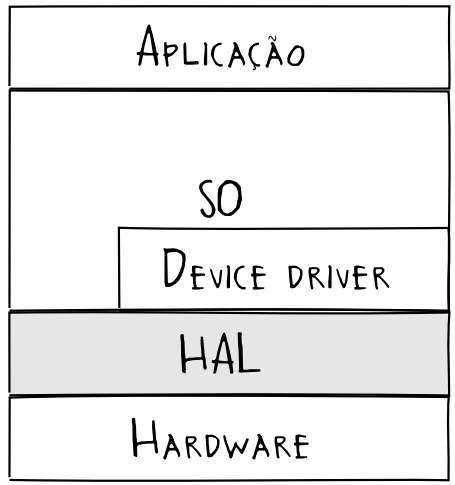
- Quais são os tipos de gerenciadores de dispositivo?
  - **Caractere**: neste tipo de dispositivo o acesso é feito por fluxo de bytes, assim como ler ou escrever em um arquivo, utilizando as chamadas de sistema
  - **Bloco**: permite que sistemas de arquivo sejam gerenciados através de blocos com tamanho múltiplo de 2 e com acesso transparente ao usuário

Software dependente do Hardware

- Quais são os tipos de gerenciadores de dispositivo?
  - **Caractere**: neste tipo de dispositivo o acesso é feito por fluxo de bytes, assim como ler ou escrever em um arquivo, utilizando as chamadas de sistema
  - **Bloco**: permite que sistemas de arquivo sejam gerenciados através de blocos com tamanho múltiplo de 2 e com acesso transparente ao usuário
  - **Rede**: proporciona a troca de dados com outros sistemas através da transmissão e no recebimento de pacotes de dados, com interface distinta da utilizada pelos dispositivos de caractere ou de bloco (*socket*)

Software dependente do Hardware

- Camada de abstração de hardware (HAL)
  - Proporciona uma camada de software para abstrair os detalhes específicos da plataforma através de uma interface de programação de software (API)
  - Cada plataforma possui suas próprias configurações de componentes (*board support package*)



Portabilidade do device driver e do SO

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
22 OUTPUT_ARCH(riscv)
23 ENTRY(_start)
24
25 _RAM_origin = 0x80000000;
26 _RAM_length = 32K;
27
28 MEMORY {
29     RAM(rwx) : ORIGIN = _RAM_origin, LENGTH =
           _RAM_length
30 }
31 PHDRS {
32     ro PT_LOAD FLAGS(4);
33     rx PT_LOAD FLAGS(5);
34     rw PT_LOAD FLAGS(6);
35 }
36 SECTIONS {
...     ...
66 }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
37      .text : ALIGN(4) {
38          _start_text = .;
39          *(.text.vector_table)
40          *(.text.init)
41          *(.text*)
42          _end_text = .;
43      } > RAM : rx
44      .rodata : ALIGN(4) {
45          _start_rodata = .;
46          *(.rodata*)
47          _end_rodata = .;
48      } > RAM : ro
...     ...
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
37      .text : ALIGN(4) {
38          _start_text = .;
39          *(.text.vector_table)
40          *(.text.init)
41          *(.text*)
42          _end_text = .;
43      } > RAM : rx
44      .rodata : ALIGN(4) {
45          _start_rodata = .;
46          *(.rodata*)
47          _end_rodata = .;
48      } > RAM : ro
...     ...
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
37      .text : ALIGN(4) {
38          _start_text = .;
39          *(.text.vector_table)
40          *(.text.init)
41          *(.text*)
42          _end_text = .;
43      } > RAM : rx
44      .rodata : ALIGN(4) {
45          _start_rodata = .;
46          *(.rodata*)
47          _end_rodata = .;
48      } > RAM : ro
...     ...
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
…       ...
49      .data : ALIGN(4) {
50          _start_data = .;
51          *(.data*)
52          _end_data = .;
53      } > RAM : rw
54      .bss : ALIGN(4) {
55          _start_bss = .;
56          *(.bss*)
57          *(COMMON)
58          _end_bss = .;
59      } > RAM : rw
…       ...
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
…       ...
49      .data : ALIGN(4) {
50          _start_data = .;
51          *(.data*)
52          _end_data = .;
53      } > RAM : rw
54      .bss : ALIGN(4) {
55          _start_bss = .;
56          *(.bss*)
57          *(COMMON)
58          _end_bss = .;
59      } > RAM : rw
…       ...
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
…       ...
49      .data : ALIGN(4) {
50          _start_data = .;
51          *(.data*)
52          _end_data = .;
53      } > RAM : rw
54      .bss : ALIGN(4) {
55          _start_bss = .;
56          *(.bss*)
57          *(COMMON)
58          _end_bss = .;
59      } > RAM : rw
…       ...
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
…       ...
60      .heap (NOLOAD) : ALIGN(4) {
61          _heap_pointer = .;
62      } > RAM : rw
63      .stack ORIGIN(RAM) + LENGTH(RAM) (NOLOAD) :
            ALIGN(16) {
64          _stack_pointer = .;
65      } > RAM : rw
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
…       ...
60      .heap (NOLOAD) : ALIGN(4) {
61          _heap_pointer = .;
62      } > RAM : rw
63      .stack ORIGIN(RAM) + LENGTH(RAM) (NOLOAD) :
            ALIGN(16) {
64          _stack_pointer = .;
65      } > RAM : rw
66  }
```

Software dependente do Hardware

- Script de vinculação (*linker script*)
  - Organiza os dados e as instruções na memória

```
36  SECTIONS {
…       ...
60      .heap (NOLOAD) : ALIGN(4) {
61          _heap_pointer = .;
62      } > RAM : rw
63      .stack ORIGIN(RAM) + LENGTH(RAM) (NOLOAD) :
            ALIGN(16) {
64          _stack_pointer = .;
65      } > RAM : rw
66  }
```

Software dependente do Hardware

- Código de montagem para inicialização

```
25  # Seção da tabela de vetor de interrupção
26  .section .text.vector_table
…   ...
55  # Seção de código de inicialização
56  .section .text.init
…   ...
112 # Função de inicialização
113 .global _start
114 _start:
115     # Ajustando registrador de topo da pilha
116     la sp, _stack_pointer
117     # Inicializando a seção bss
118     call _init_bss
119     # Chamando a função principal
120     call main
121     # Finalizando a simulação
122     j _exit
```

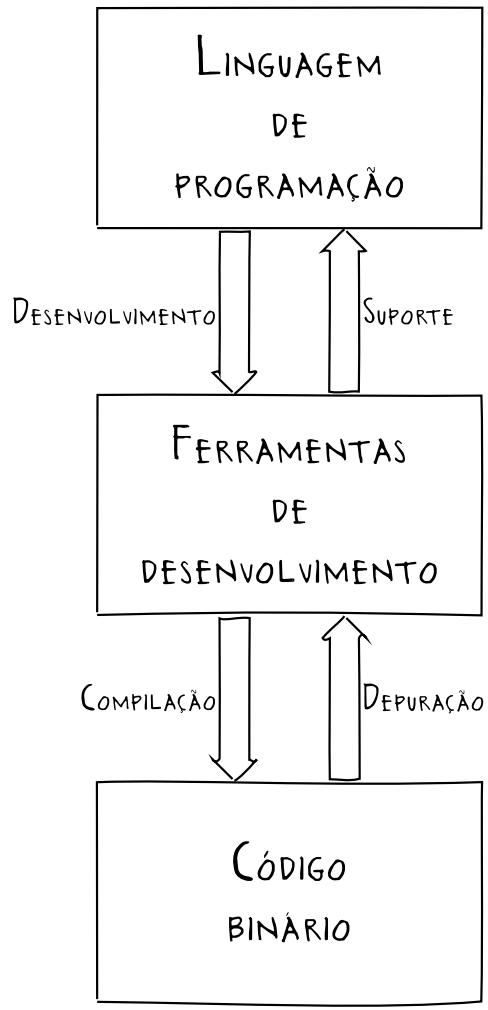
Software dependente do Hardware

- Código de montagem para inicialização

```
55  # Seção de código de inicialização
56  .section .text.init
57  # Função de inicialização da seção bss
58  _init_bss:
59      # Obtendo limites de bss
60      la a0, _start_bss
61      la a1, _end_bss
62      # Inicializando com zero
63      _init_bss_loop:
64          # Escrevendo zero na memória
65          sw zero, 0(a0)
66          # Incrementando ponteiro de 4
67          addi a0, a0, 4
68          # Checando se index < _end_bss
69          blt a0, a1, _init_bss_loop
70      # Returnando da chamada
71      ret
... ...
```

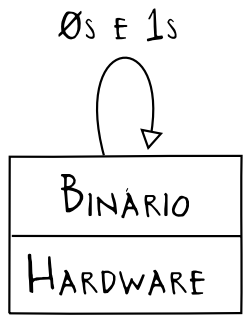
Ferramentas de desenvolvimento

- As ferramentas de desenvolvimento criam a infraestrutura básica para compilação e depuração



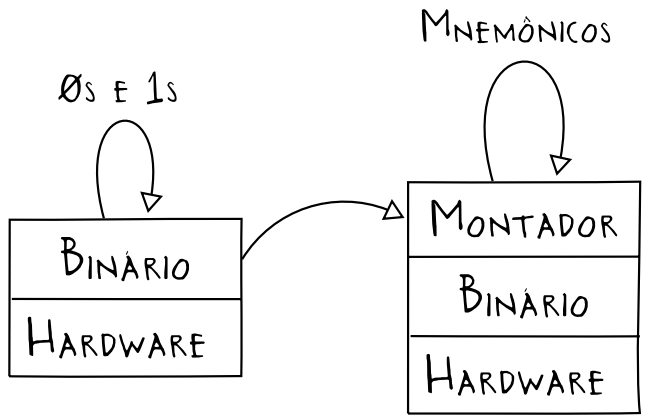
Ferramentas de desenvolvimento

- Etapas de *bootstrapping*



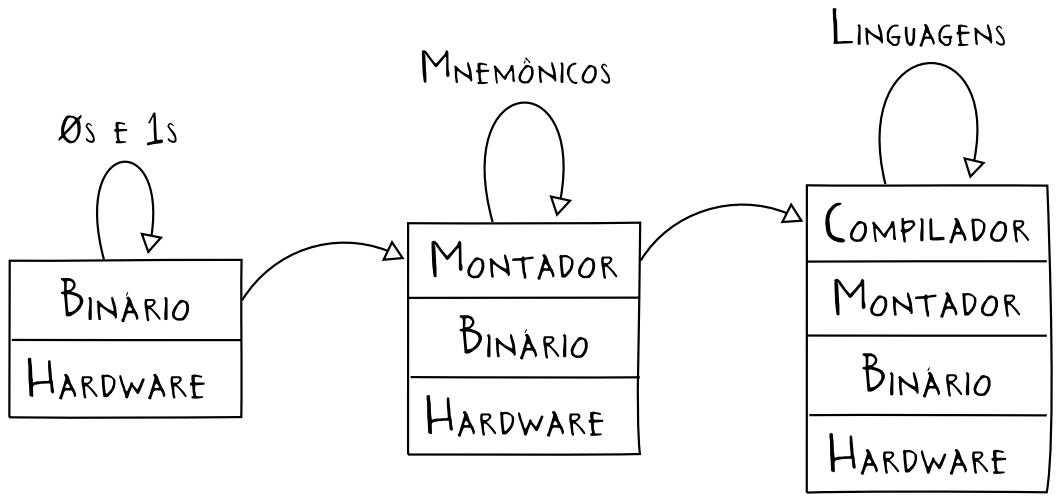
Ferramentas de desenvolvimento

- Etapas de *bootstrapping*



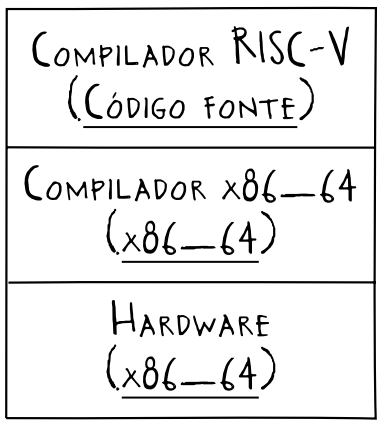
Ferramentas de desenvolvimento

- Etapas de *bootstrapping*



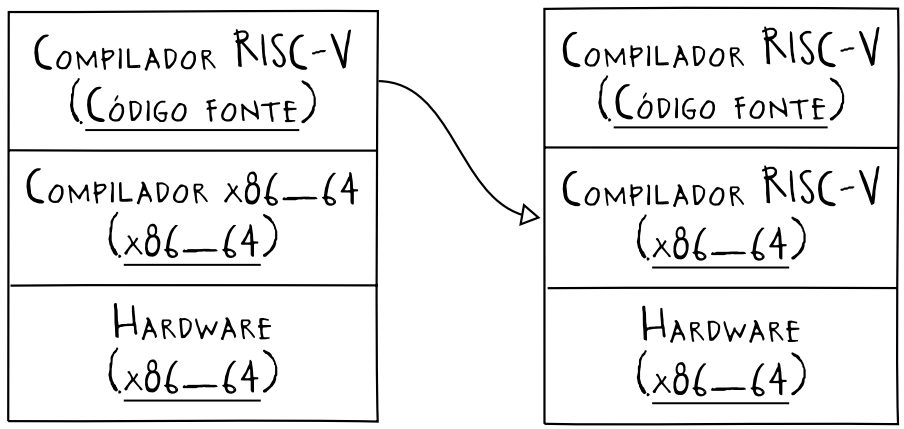
Ferramentas de desenvolvimento

- Etapas de compilação cruzada



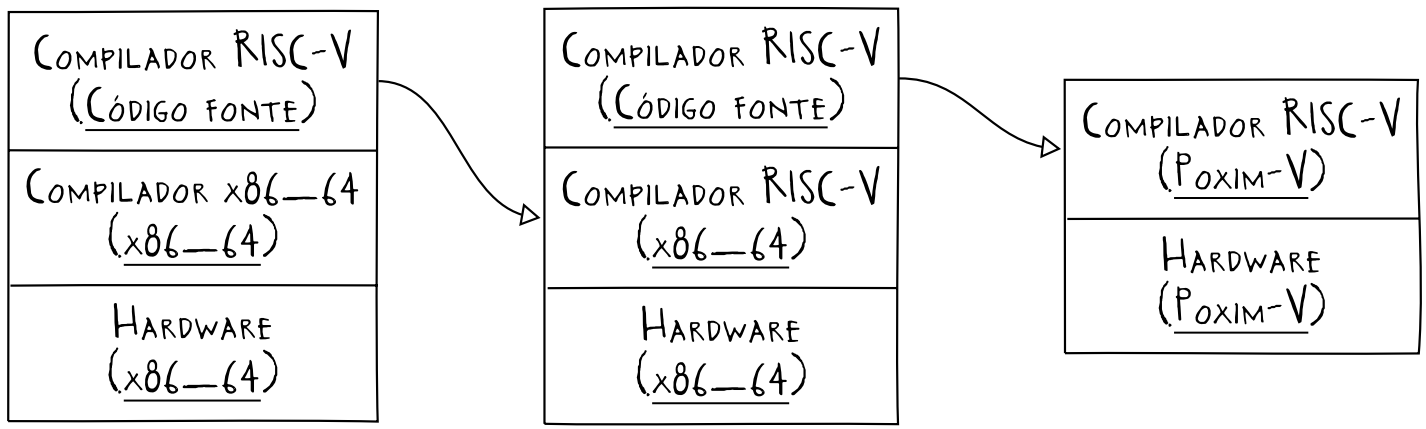
Ferramentas de desenvolvimento

- Etapas de compilação cruzada



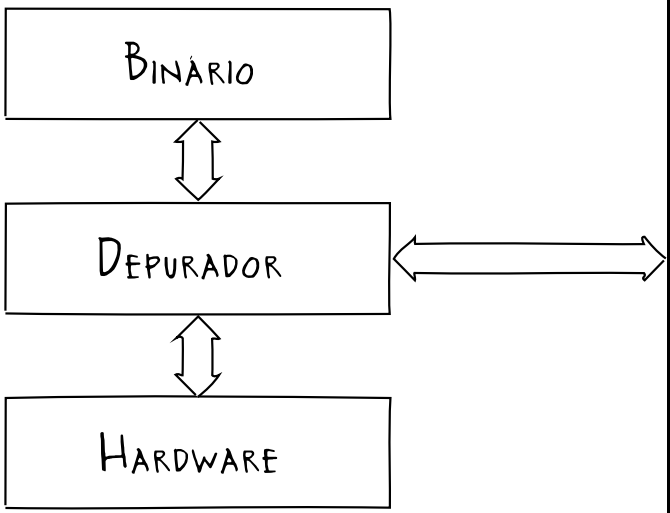
Ferramentas de desenvolvimento

- Etapas de compilação cruzada



Ferramentas de desenvolvimento

- Depuração do comportamento (*debugging*)

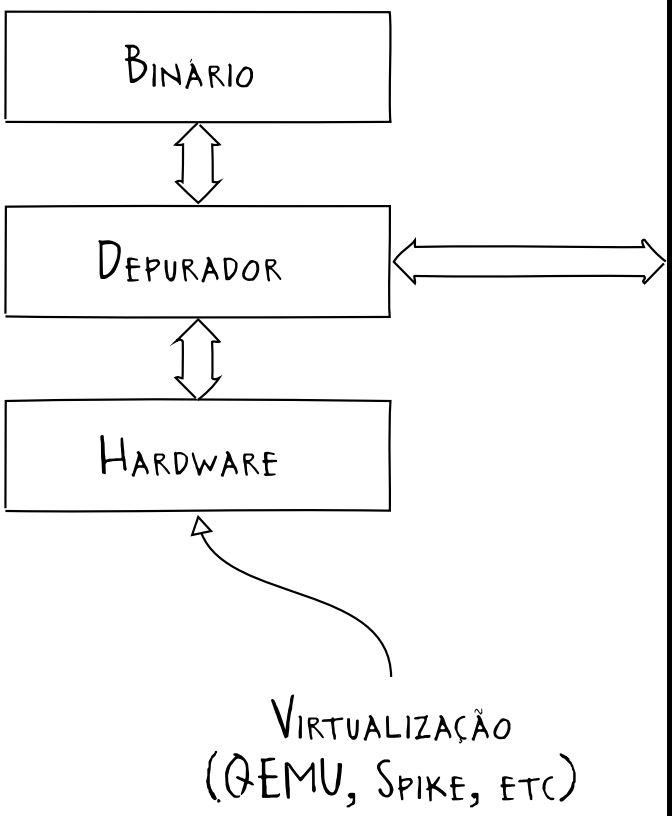



```
GNU gdb (Poxim-V) 14.2
Copyright (C) 2023 Free Software Foundation, Inc.
.
.
.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) file 1_factorial.elf
Reading symbols from 1_factorial.elf...
(gdb) target remote :1234
Remote debugging using :1234
0x00001000 in ?? ()
(gdb) break main
Breakpoint 1 at 0x800000d8: file 1_factorial.s, line 63.
(gdb) continue
Continuing.

Breakpoint 1, main () at 1_factorial.s:63
63          addi sp, sp, -16
(gdb) disassemble
Dump of assembler code for function main:
=> 0x800000d8 <+0>:     addi    sp,sp,-16
   0x800000dc <+4>:     sw      ra,0(sp)
   0x800000e0 <+8>:     li      a0,5
   0x800000e4 <+12>:    jal     0x800000a0 <factorial>
   0x800000e8 <+16>:    lw      ra,0(sp)
   0x800000ec <+20>:    addi    sp,sp,16
   0x800000f0 <+24>:    li      a0,0
   0x800000f4 <+28>:    ret
End of assembler dump.
(gdb)
```

Ferramentas de desenvolvimento

- Depuração do comportamento (*debugging*)




```
GNU gdb (Poxim-V) 14.2
Copyright (C) 2023 Free Software Foundation, Inc.
.
.
.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) file 1_factorial.elf
Reading symbols from 1_factorial.elf...
(gdb) target remote :1234
Remote debugging using :1234
0x00001000 in ?? ()
(gdb) break main
Breakpoint 1 at 0x800000d8: file 1_factorial.s, line 63.
(gdb) continue
Continuing.

Breakpoint 1, main () at 1_factorial.s:63
63          addi sp, sp, -16
(gdb) disassemble
Dump of assembler code for function main:
=> 0x800000d8 <+0>:     addi    sp,sp,-16
   0x800000dc <+4>:     sw      ra,0(sp)
   0x800000e0 <+8>:     li      a0,5
   0x800000e4 <+12>:    jal     0x800000a0 <factorial>
   0x800000e8 <+16>:    lw      ra,0(sp)
   0x800000ec <+20>:    addi    sp,sp,16
   0x800000f0 <+24>:    li      a0,0
   0x800000f4 <+28>:    ret
End of assembler dump.
(gdb)
```

Exercício

- A infraestrutura de software básico é fundamental para utilização da arquitetura RISC-V, principalmente as rotinas de inicialização, os scripts de vinculação e as ferramentas de desenvolvimento
  - Estude as diretivas e sintaxe suportadas pelo montador, incluindo as dependências para arquitetura alvo
  - Analise a documentação de comandos do GDB
  - Verifique as opções de depuração do QEMU