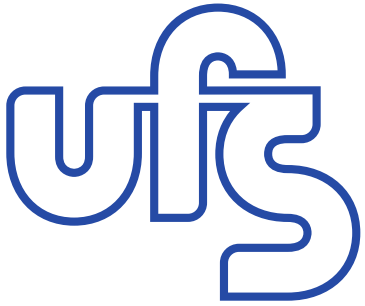
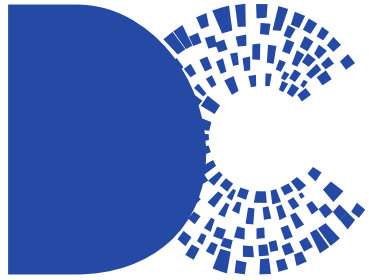


UNIVERSIDADE FEDERAL DE SERGIPE



DEPARTAMENTO DE COMPUTAÇÃO

Exceção e interrupção

Arquitetura de Computadores

Bruno Prado

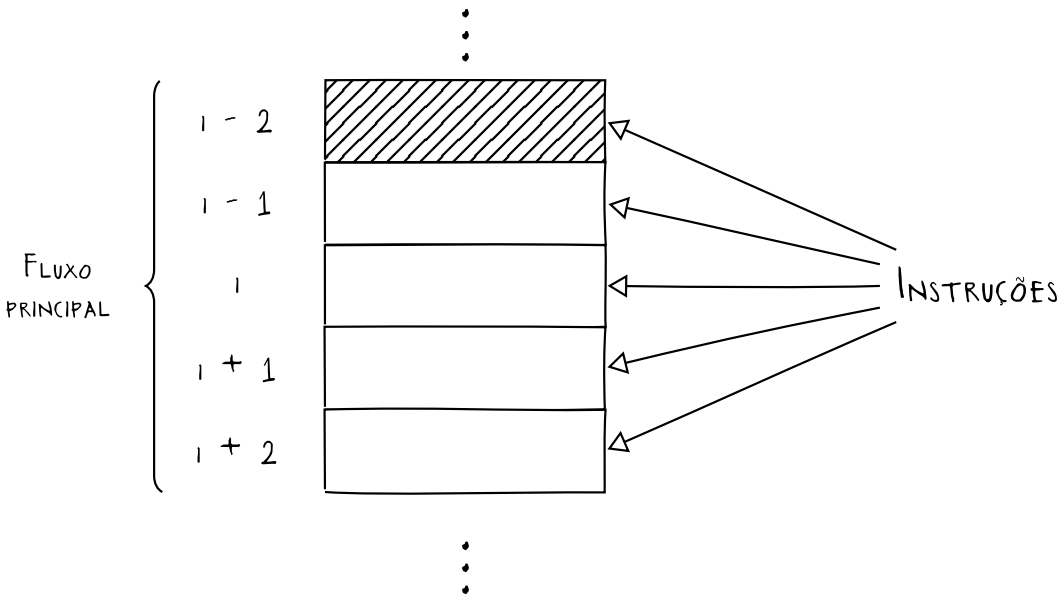
Departamento de Computação / UFS

Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma

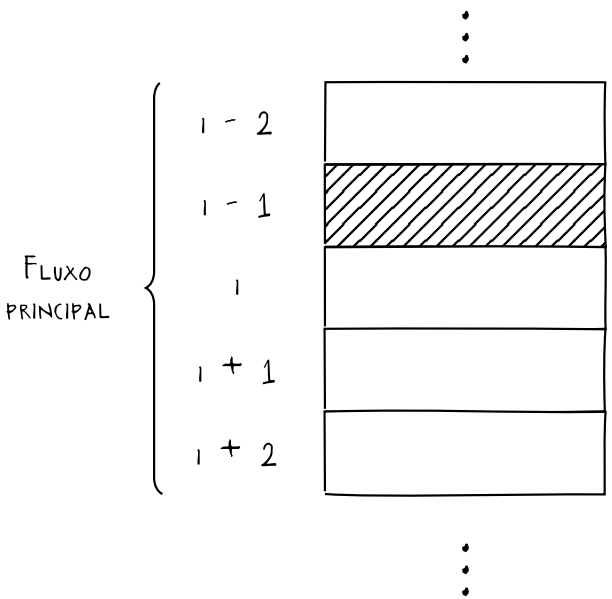
Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



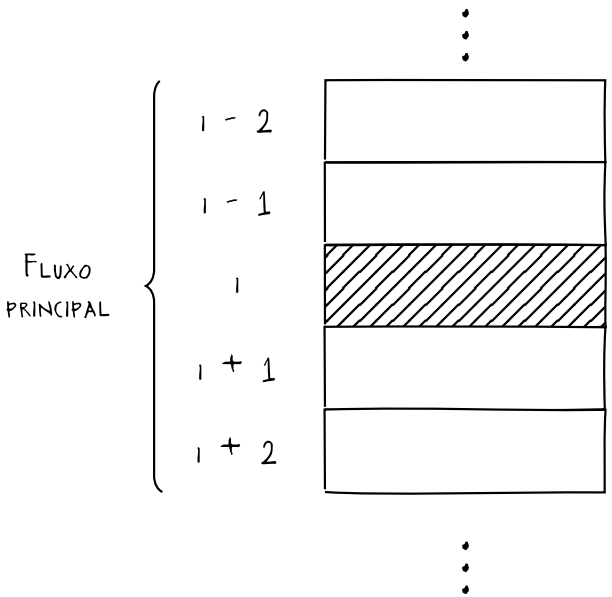
Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



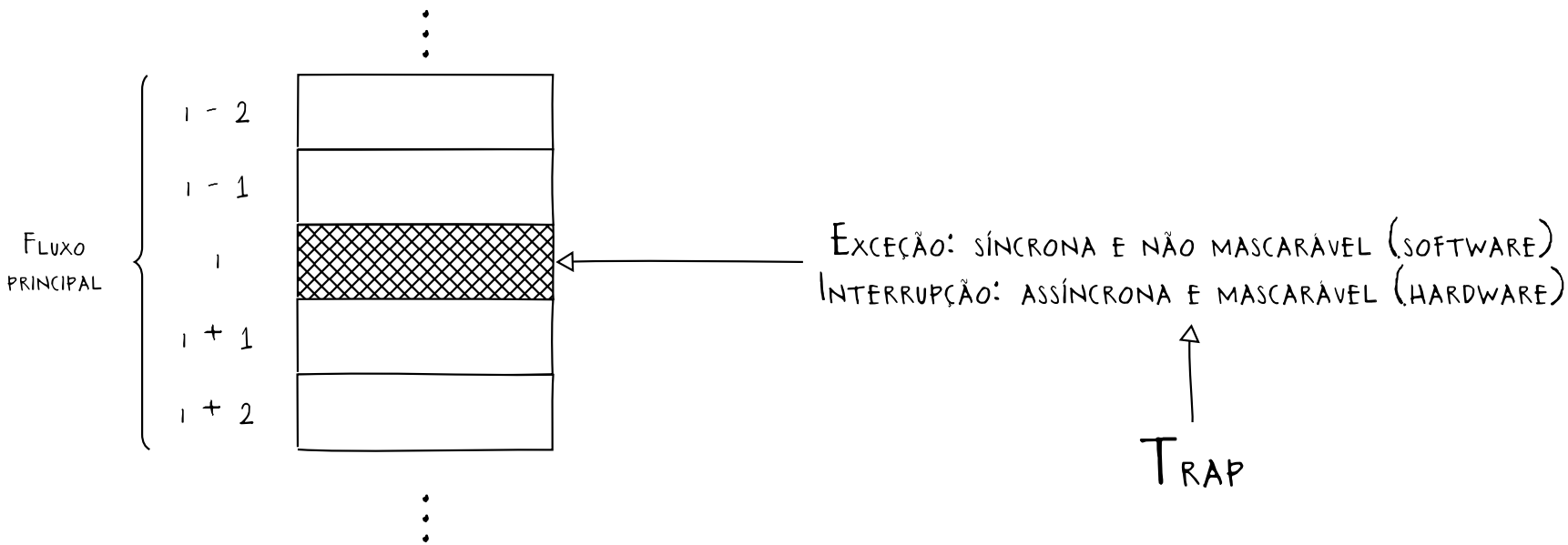
Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



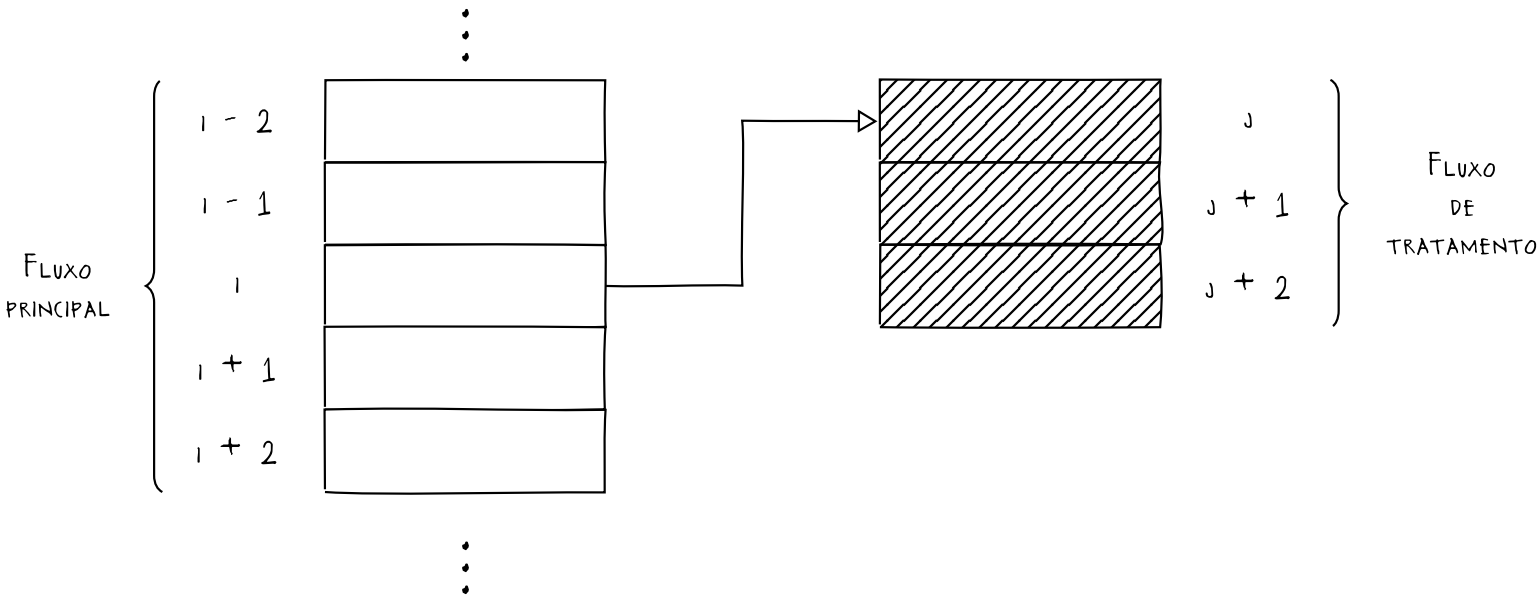
Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



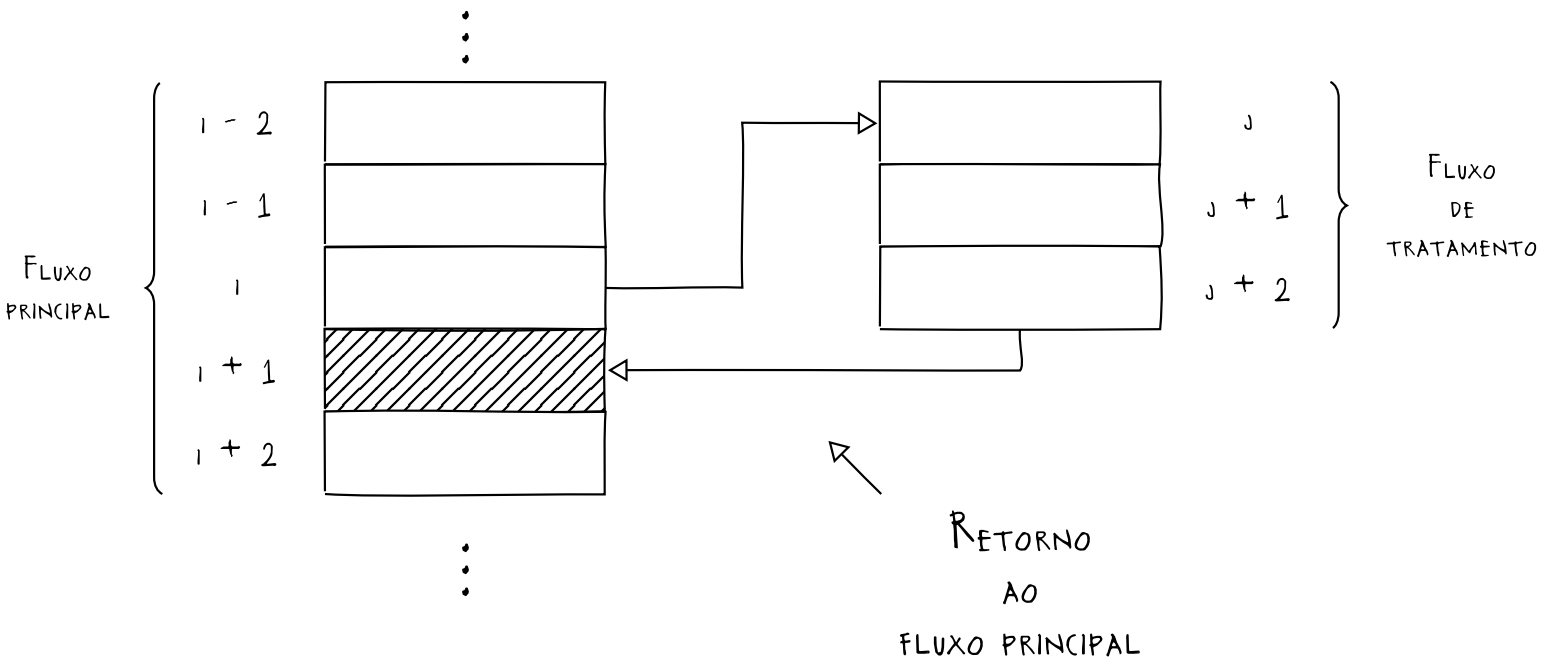
Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



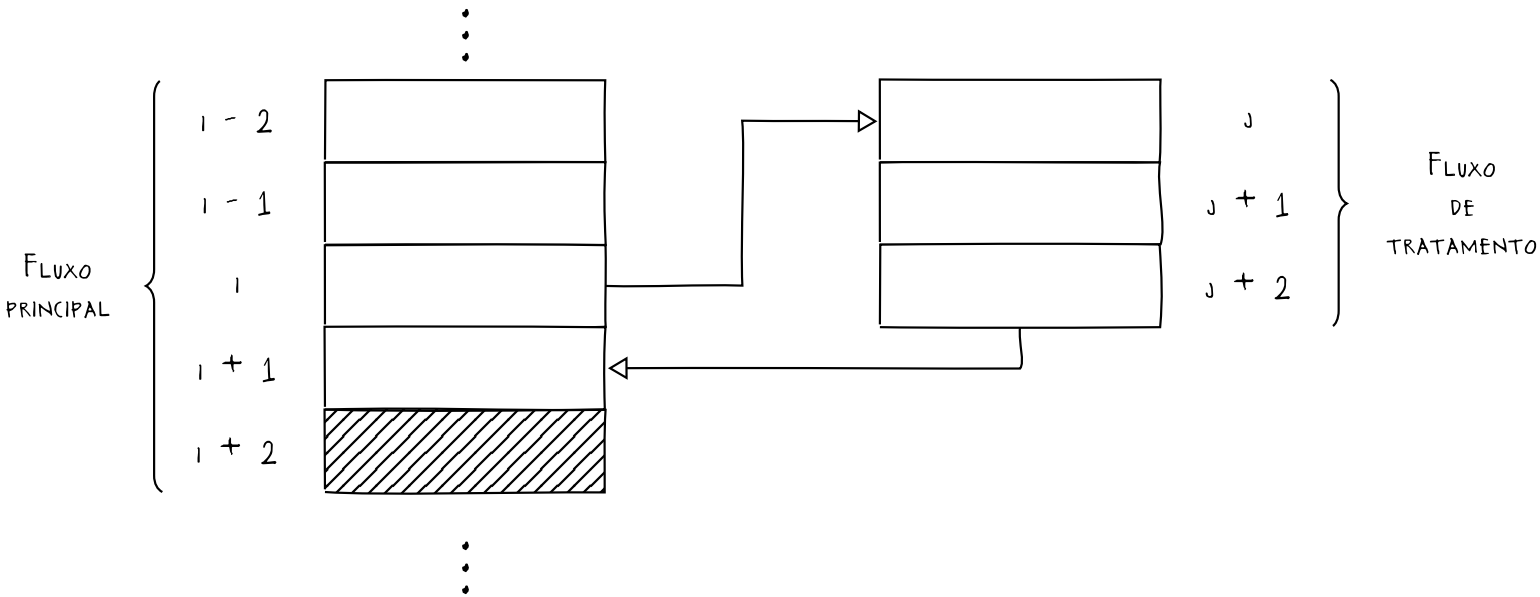
Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



Introdução

- O que é uma exceção/interrupção?
  - É um evento (*trap*) causado pela execução do software (exceção) no processador ou requisitado por um dispositivo de hardware (interrupção) da plataforma



Introdução

- Por que utilizar exceção/interrupção é necessário?

Introdução

- Por que utilizar exceção/interrupção é necessário?
  - Evita a espera do processador, sem reduzir a eficiência de execução das operações no software
  - Previne a utilização de *polling* em dispositivos de E/S

```
1  // Biblioteca de E/S padrão
2  #include <stdio.h>
3  // Função principal
4  int main() {
5      // Variável de nome
6      char nome[50] = { 0 };
7      // Mensagem de pergunta
8      printf("Qual␣é␣o␣seu␣nome?\n");
9      // Leitura do teclado
10     scanf("%s", nome);
11     // Mensagem de resposta
12     printf("Olá␣%s!\n", nome);
13     // Retorno sem erros
14     return 0;
15 }
```

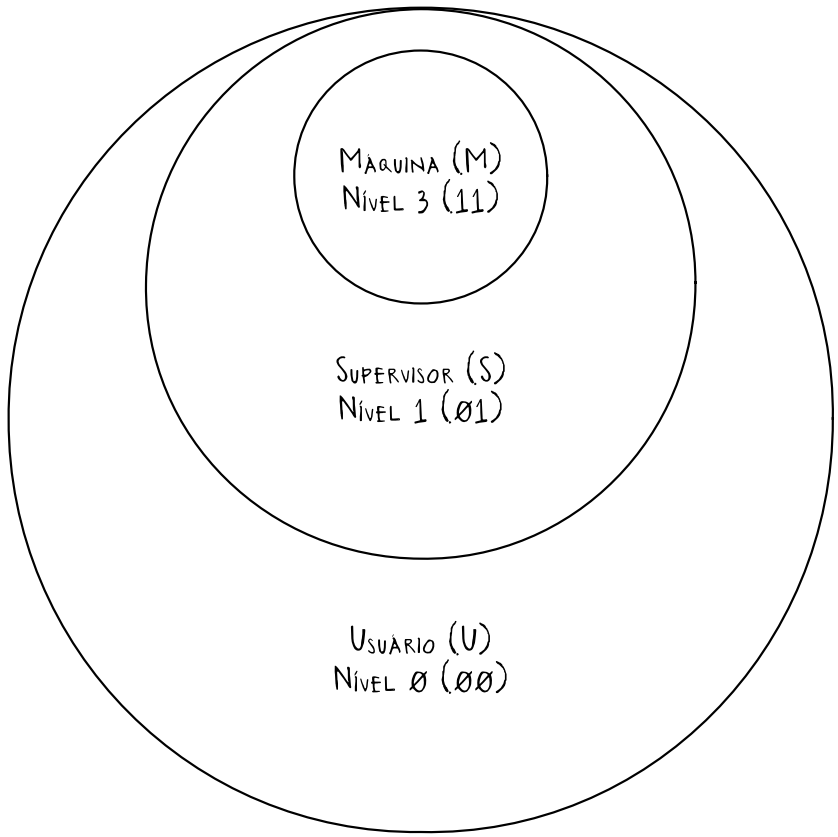
Introdução

- Por que utilizar exceção/interrupção é necessário?
  - Evita a espera do processador, sem reduzir a eficiência de execução das operações no software
  - Previne a utilização de *polling* em dispositivos de E/S

```
1  // Biblioteca de E/S padrão
2  #include <stdio.h>
3  // Função principal
4  int main() {
5      // Variável de nome
6      char nome[50] = { 0 };
7      // Mensagem de pergunta
8      printf("Qual␣é␣o␣seu␣nome?\n");
9      // Leitura do teclado
10     scanf("%s", nome);
11     // Mensagem de resposta
12     printf("Olá␣%s!\n", nome);
13     // Retorno sem erros
14     return 0;
15 }
```

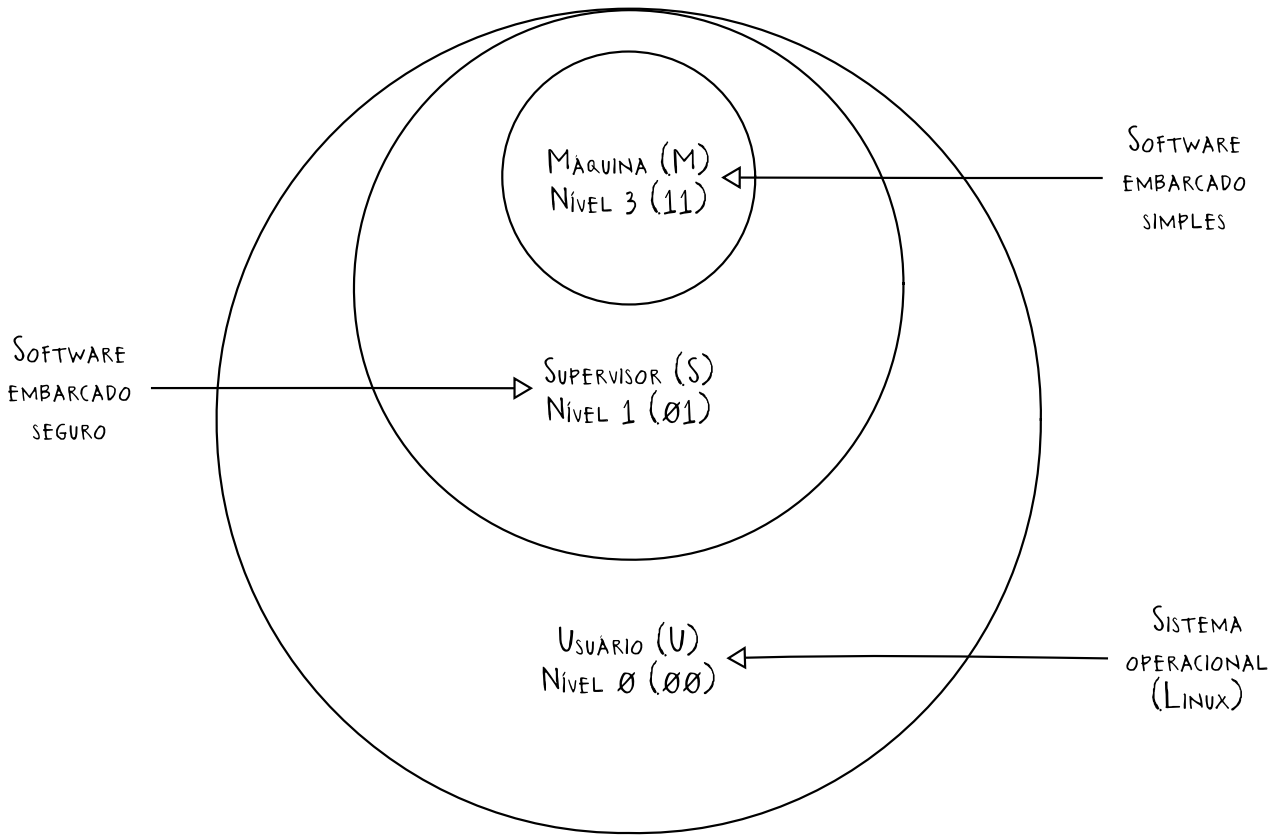
Níveis de privilégio

- A *thread* de hardware (*hart*) executa em algum nível de privilégio (somente o modo M será suportado), codificado nos registradores de controle e de status (CSRs)



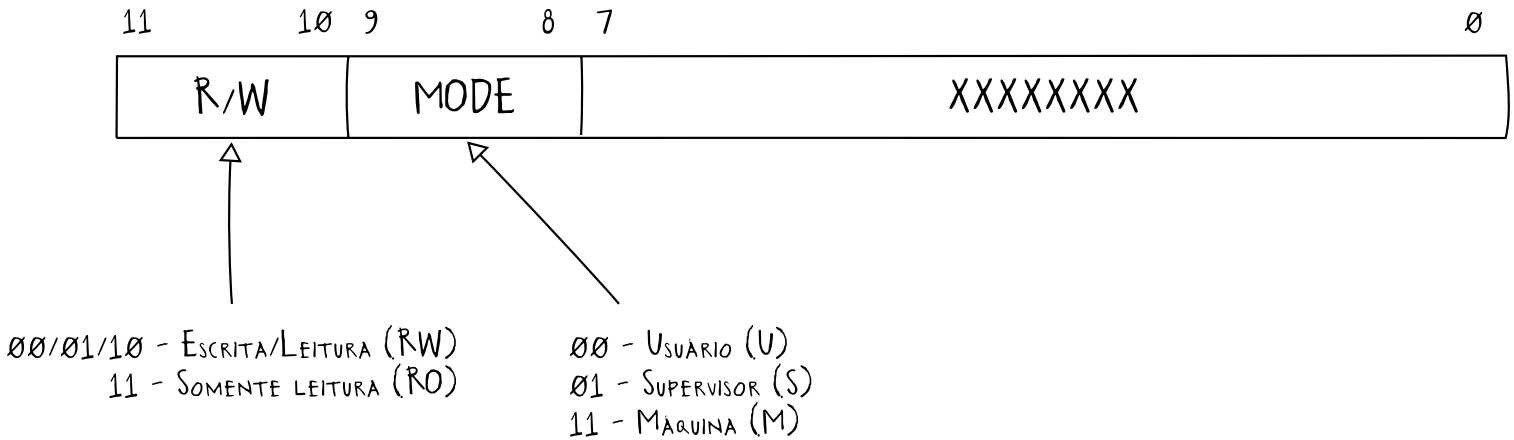
Níveis de privilégio

- A *thread* de hardware (*hart*) executa em algum nível de privilégio (somente o modo M será suportado), codificado nos registradores de controle e de status (CSRs)



Registradores de Controle e de Status (CSRs)

- É utilizado um espaço de 12 bits (4096 posições) para endereçamento, considerando a permissão de escrita ou leitura ($CSR[11:10]$) e o nível de privilégio mínimo necessário ($CSR[9:8]$)



Registradores de Controle e de Status (CSRs)

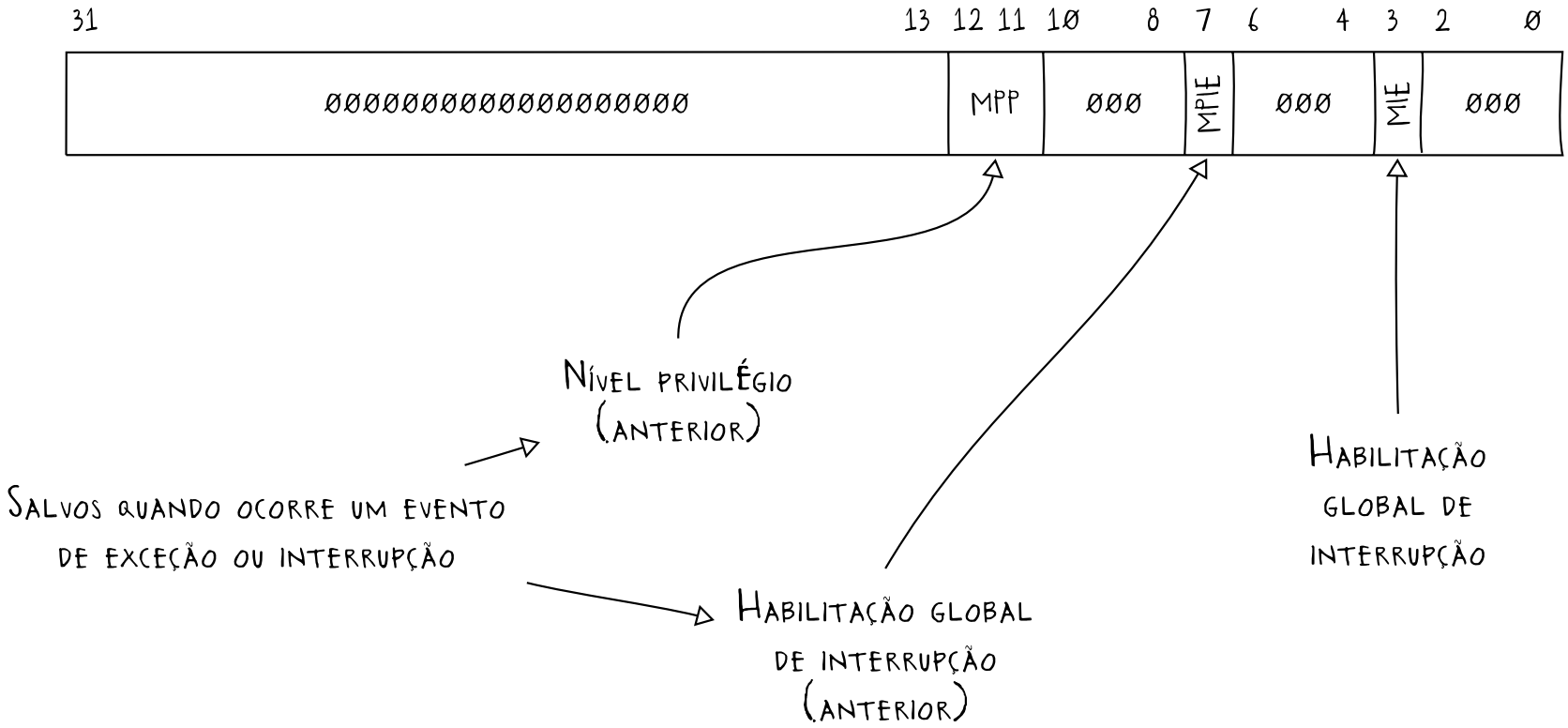
- Endereços dos registradores alocados

Número	Permissão	Nome	Descrição
0x300	MRW	mstatus	Registrador de status
0x304	MRW	mie	Habilitação de interrupção
0x305	MRW	mtvec	Endereço base do gerenciador
0x341	MRW	mepc	PC que gerou evento
0x342	MRW	mcause	Causa do evento
0x343	MRW	mtval	Endereço/instrução inválido
0x344	MRW	mip	Pendência de interrupção

Subconjunto suportado pelo Poxim-V

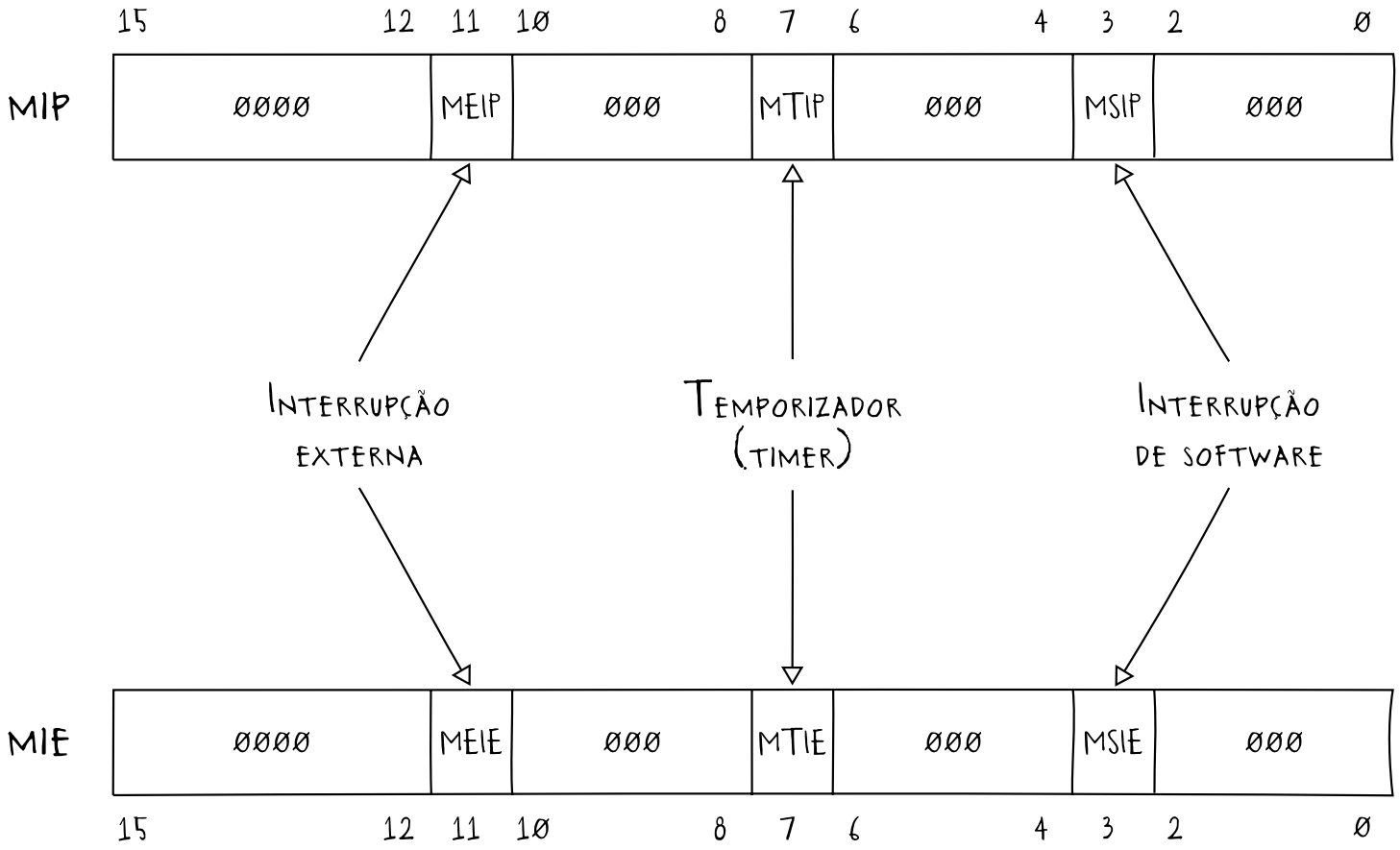
Registradores de Controle e de Status (CSRs)

- Registrador de status no nível de máquina (mstatus)



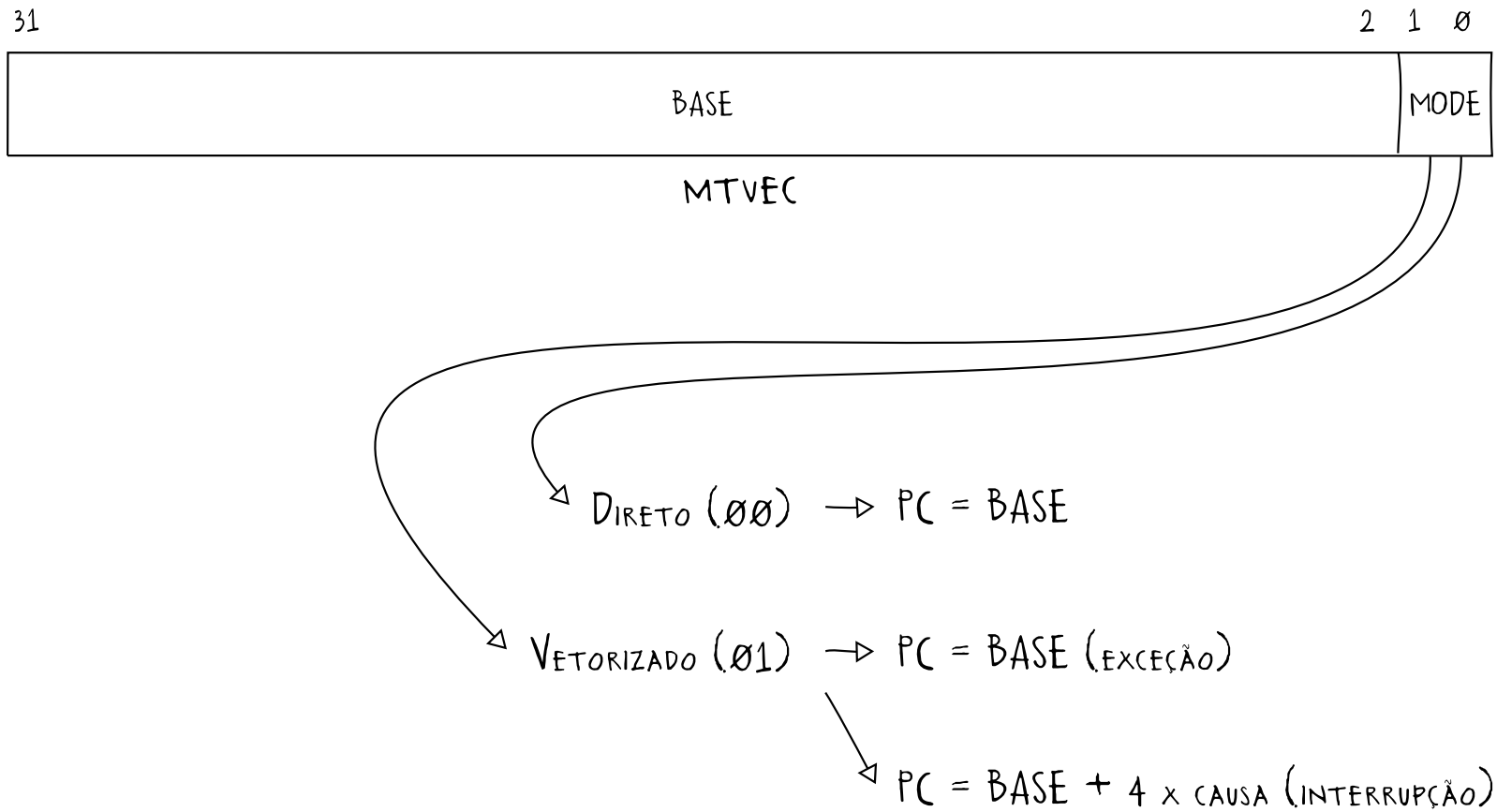
Registradores de Controle e de Status (CSRs)

- Registradores para habilitação (mie) e sinalização de pendências (mip) de interrupções



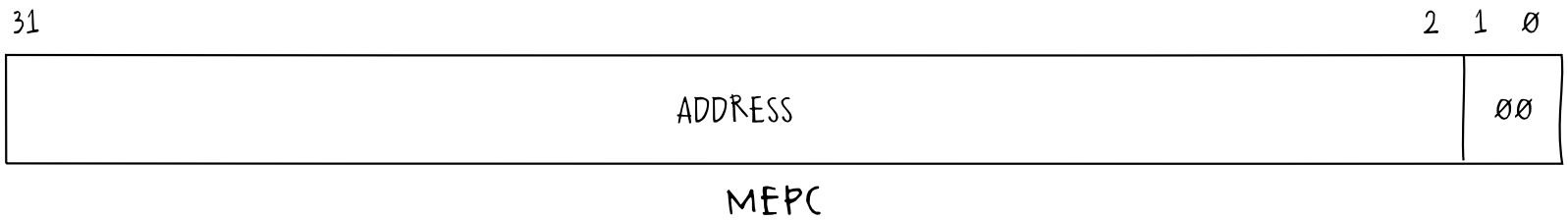
Registradores de Controle e de Status (CSRs)

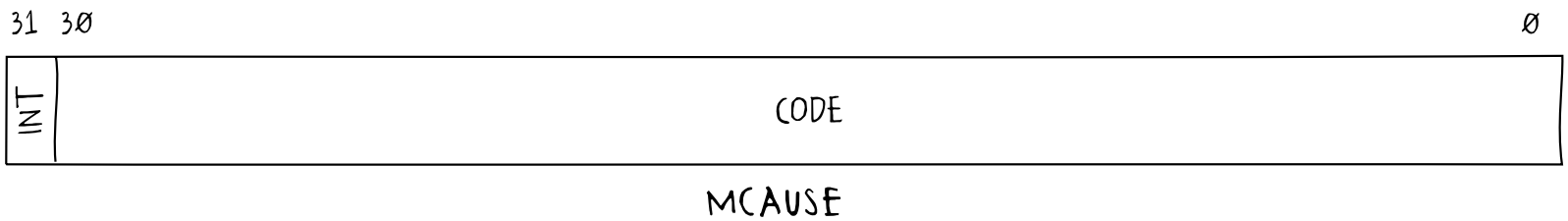
- Registrador de endereço da rotina de tratamento (mtvec)

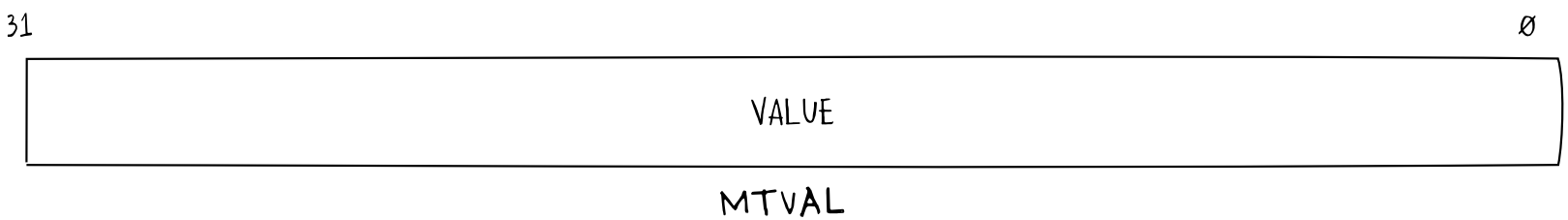


Registradores de Controle e de Status (CSRs)

- Registradores de valor do pc (mepc), causa (mcause) e endereço/instrução (mtval)

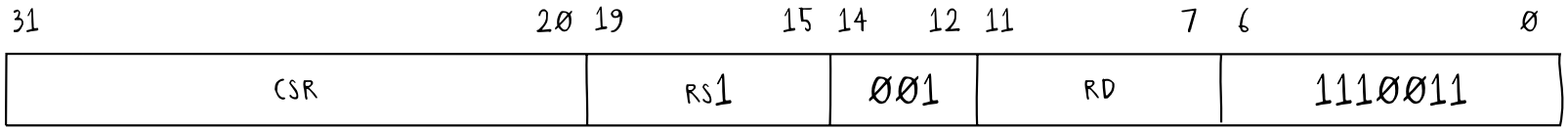






Instruções de manipulação dos CSRs

- Leitura e escrita de CSR (csrrw)

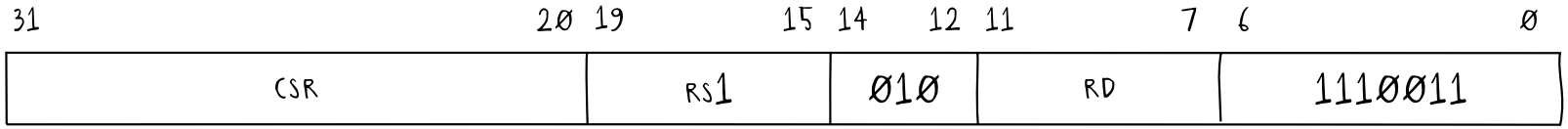



```
csrrw rd, csr, rs1:
    rd = csr
    csr = rs1

csrw csr, rs1:
    csrrw zero, csr, rs1
```

Instruções de manipulação dos CSRs

- Leitura e definição de CSR (csrrs)

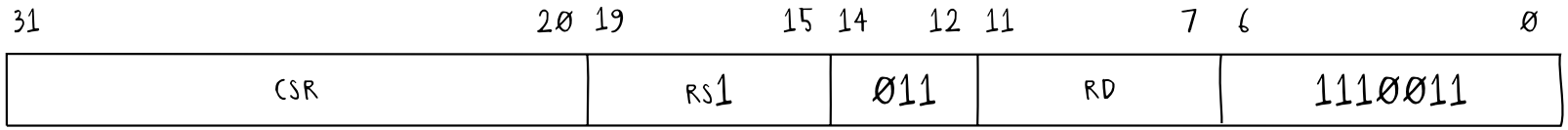



```
csrrs rd, csr, rs1:
    rd = csr
    csr = csr | rs1

csrr rd, csr:
    csrrs rd, csr, zero
```

Instruções de manipulação dos CSRs

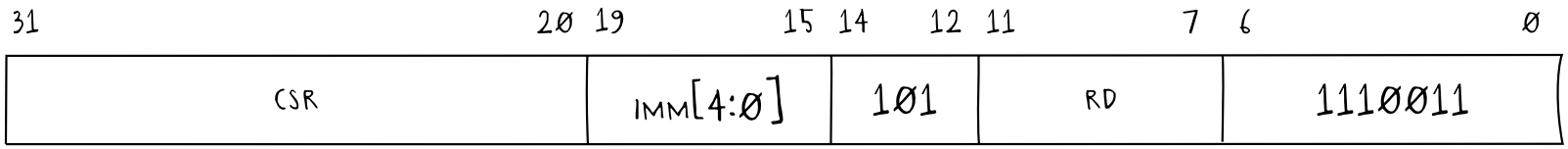
- Leitura e limpeza de CSR (csrrc)




```
csrrc rd, csr, rs1:
    rd = csr
    csr = csr & ~rs1
```

Instruções de manipulação dos CSRs

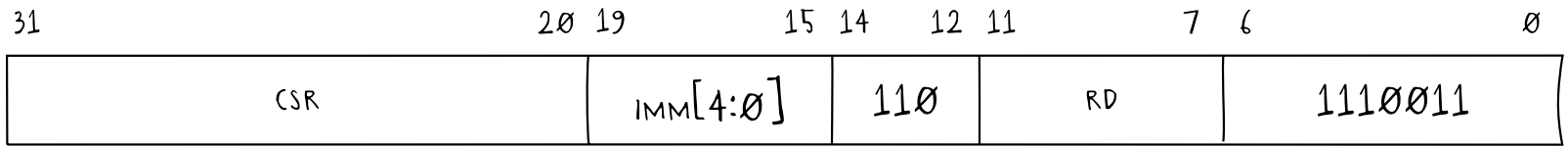
- Leitura e escrita imediata de CSR (csrrwi)




```
csrrwi rd, csr, imm:
    rd = csr
    csr = zero_extension(imm)
```

Instruções de manipulação dos CSRs

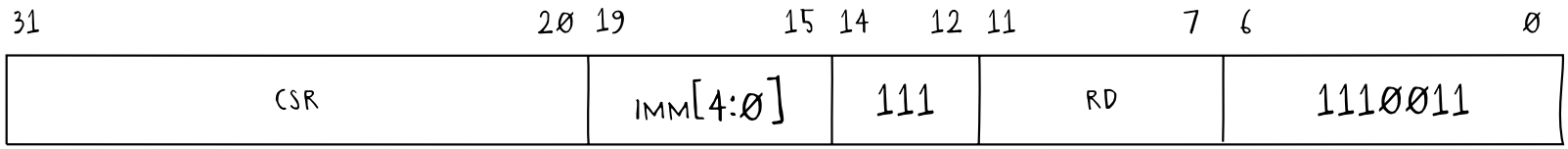
- Leitura e definição imediata de CSR (csrrsi)




```
csrrsi rd, csr, imm:
    rd = csr
    csr = csr | zero_extension(imm)
```

Instruções de manipulação dos CSRs

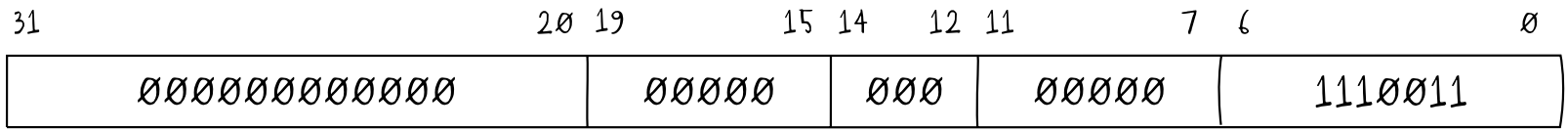
- Leitura e limpeza imediata de CSR (csrrci)




```
csrrci rd, csr, imm:
    rd = csr
    csr = csr & ~zero_extension(imm)
```

Controle do fluxo de tratamento

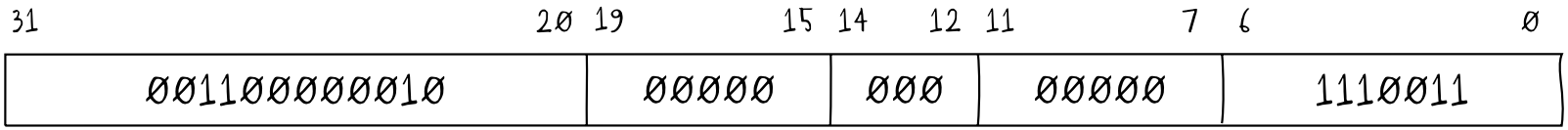
- Chamada para ambiente (*ecall*)



Realiza chamada para o ambiente
(gera exceção de código 11)

Controle do fluxo de tratamento

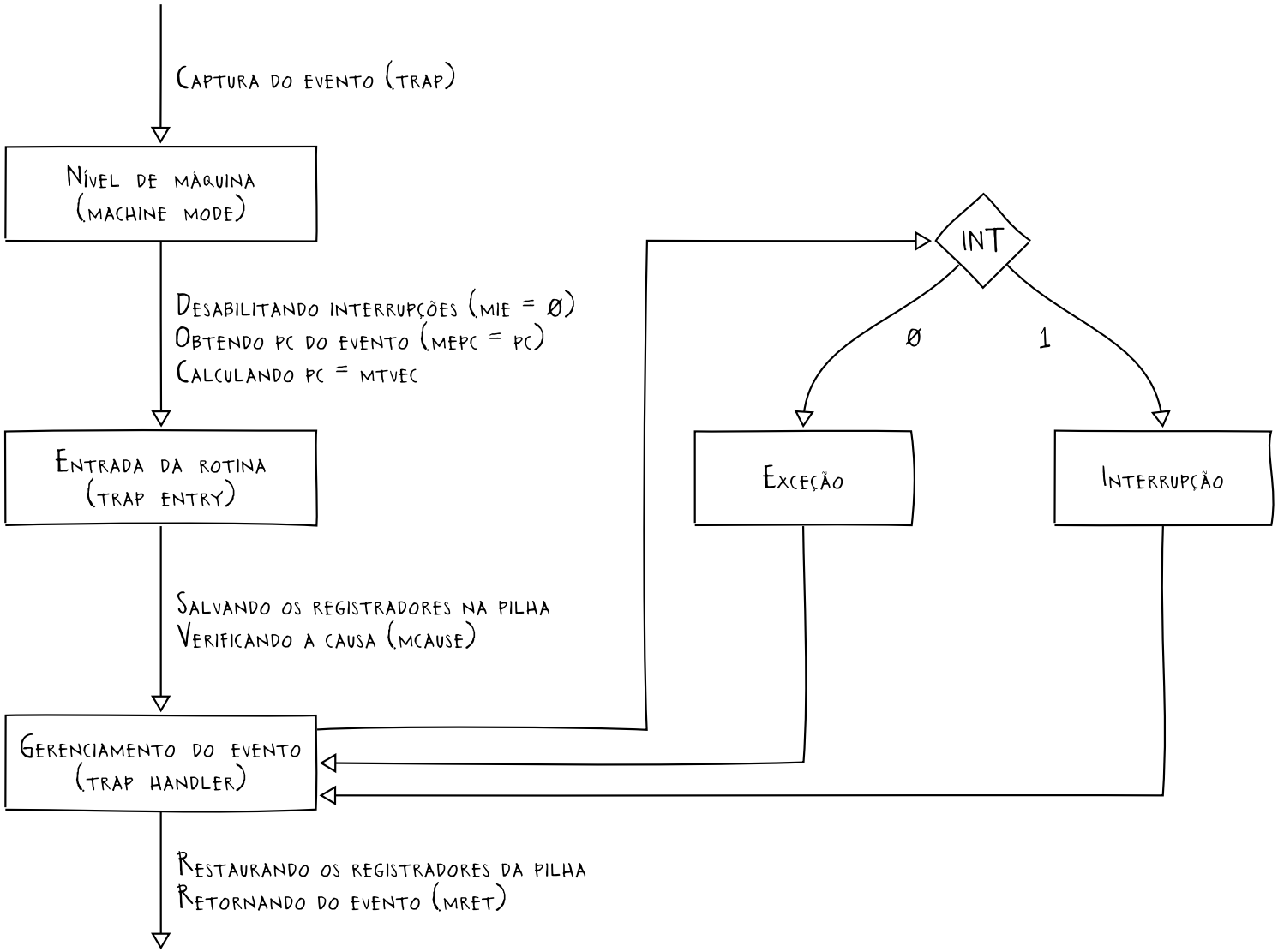
- Retorno do nível de máquina (*mret*)




```
mret:
    pc = mepc
```

Controle do fluxo de tratamento

- Etapas do tratamento da exceção/interrupção



Controle do fluxo de tratamento

- Função principal

```
63  # Função principal
64  main:
…       ...
69      # Ajuste do endereço da rotina de tratamento
70      call trap_configuration
71      # Exceções
72      exceptions:
73          # Instruction access fault (1)
74          jr zero
75          # Illegal instruction (2)
76          .word 0xf0f0f0f0
77          # Load access fault (5)
78          lw zero, 1(zero)
79          # Store access fault (7)
80          sw zero, 3(zero)
81          # Environment call
82          ecall
…       ...
```

Controle do fluxo de tratamento

- Ajustando o endereço da rotina de tratamento

```
55 # Configuração da rotina de tratamento
56 trap_configuration:
57     # Obtendo endereço da rotina de entrada
58     la t0, _trap_entry
59     csrw mtvec, t0
60     # Retornando da chamada
61     ret
```

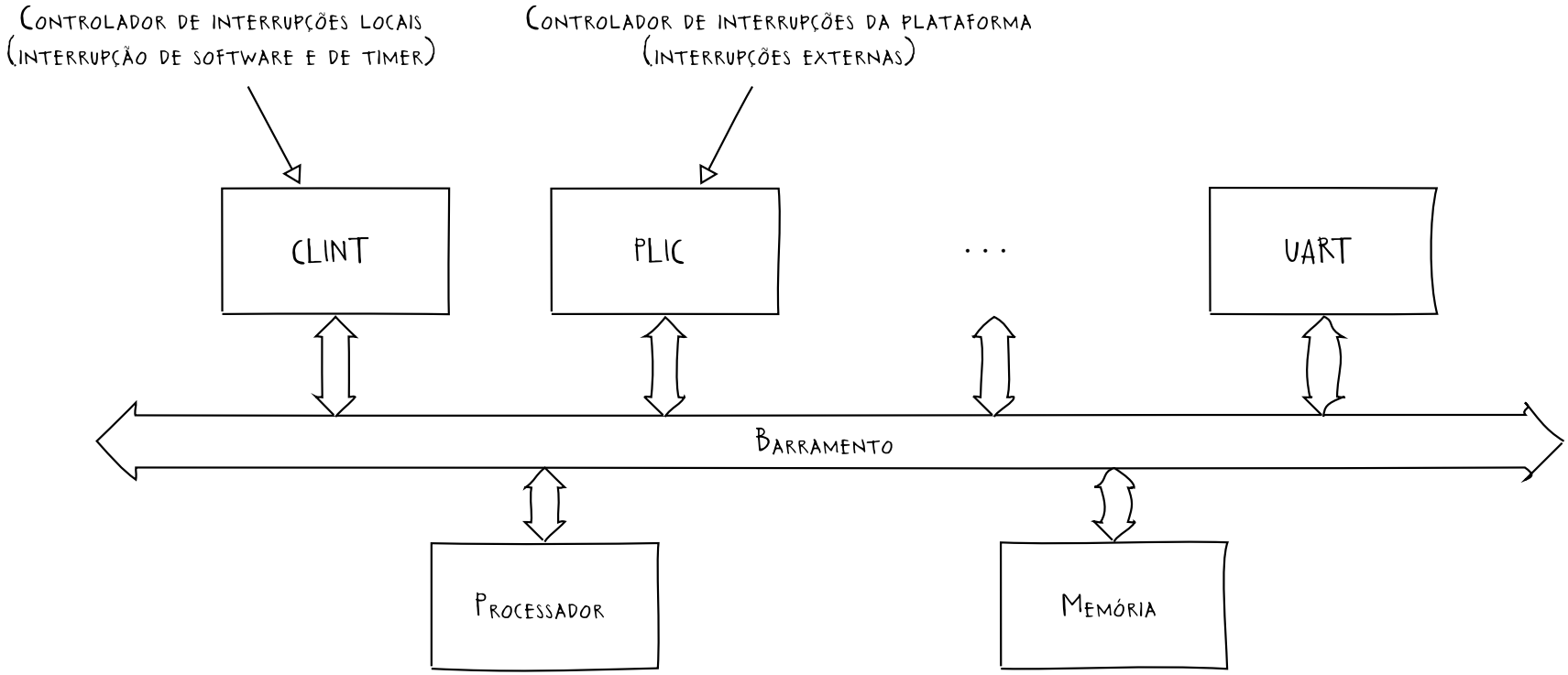
Controle do fluxo de tratamento

- Rotina de tratamento da exceção

```
39  # Tratamento da exceção
40  .global exception_handler
41  exception_handler:
…       ...
33      # Obtendo mcause e mepc
34      csrr a0, mcause
35      csrr t0, mepc
36      # Checando por instruction access fault (1)
37      li t1, 1
38      bne a0, t1, increment_mepc
39      la t1, exceptions
40      add t0, t0, t1
41      # Incrementando mepc por 4
42      increment_mepc:
43          addi t0, t0, 4
44          csrw mepc, t0
45      ...
51      # Retornando do tratamento
52      mret
```

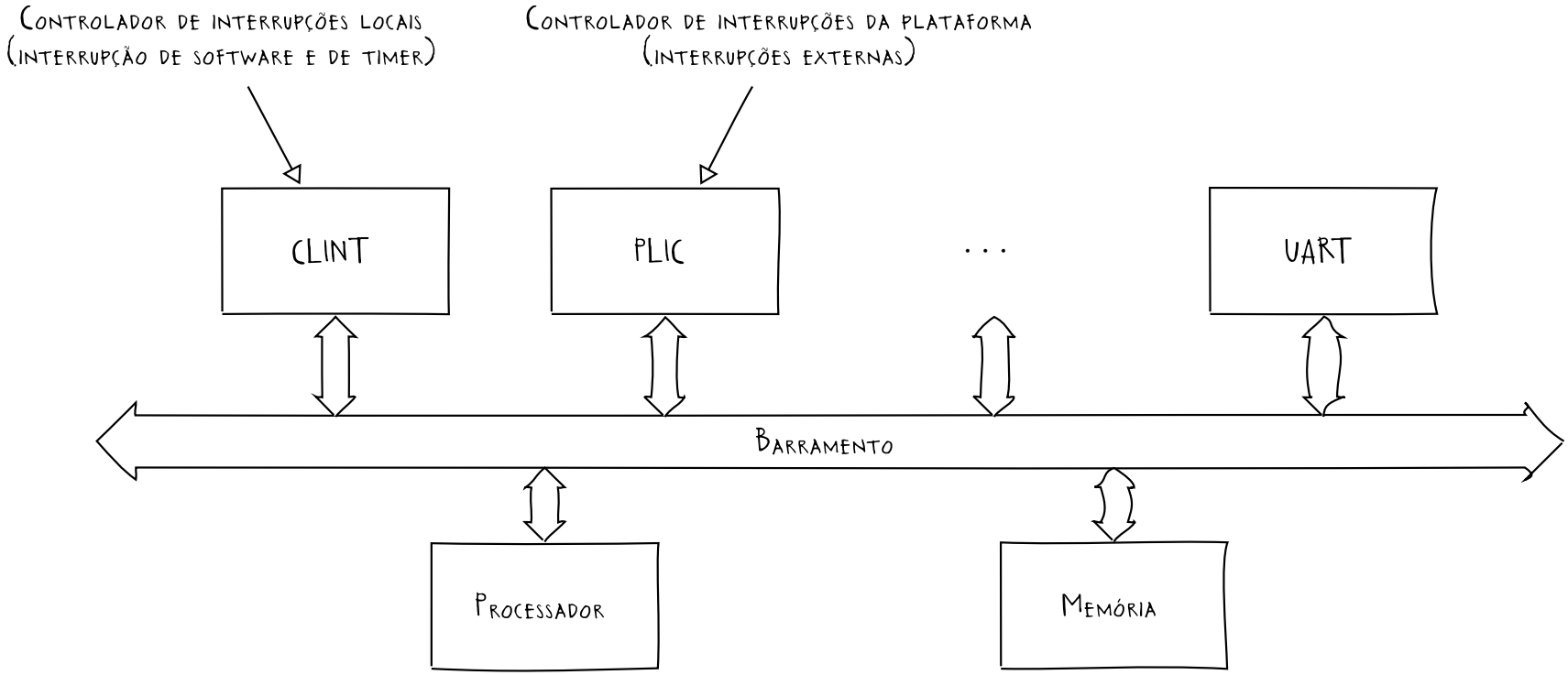
Eventos de interrupção do hardware

- São requisições assíncronas da plataforma (mascaráveis) para execução de rotinas de tratamento



Eventos de interrupção do hardware

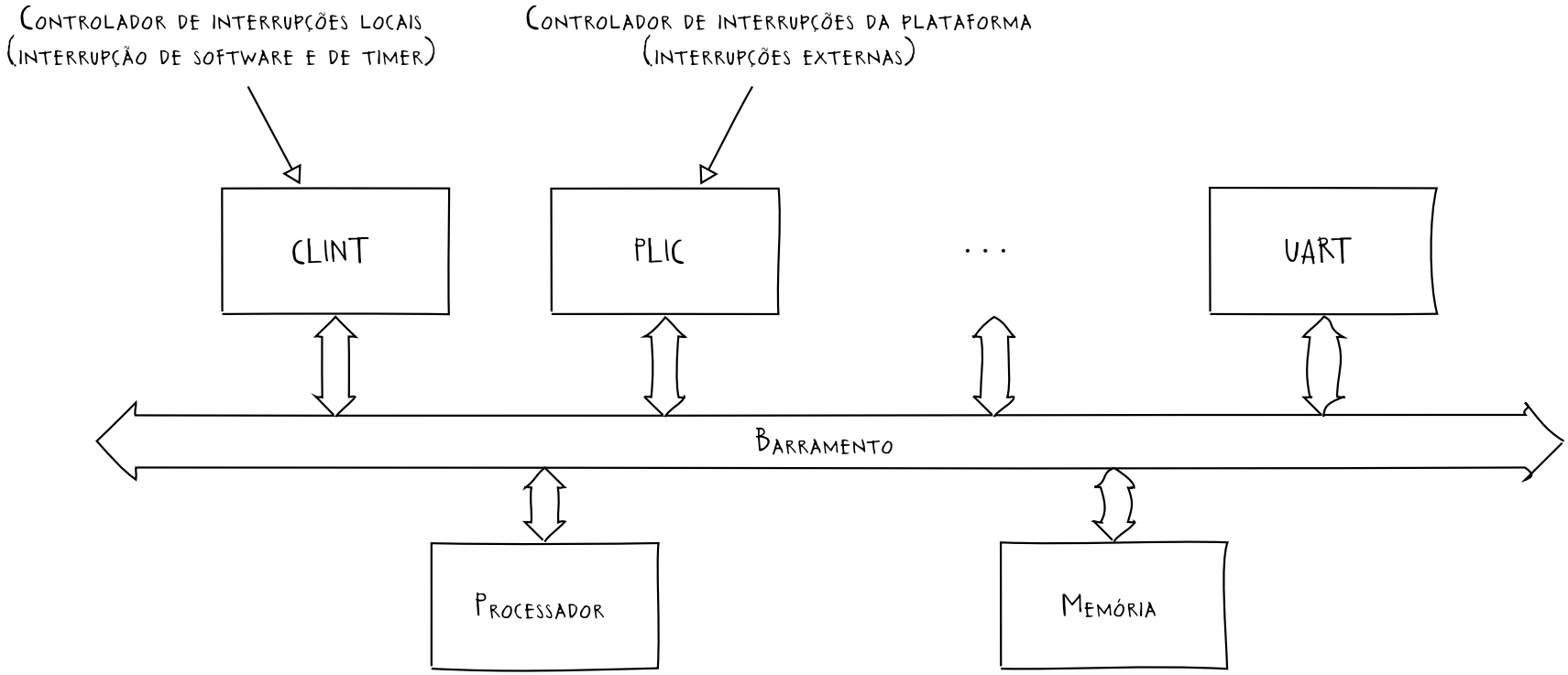
- São requisições assíncronas da plataforma (mascaráveis) para execução de rotinas de tratamento



Priorização das interrupções
no nível de máquina:
Externa > Software > Timer

Eventos de interrupção do hardware

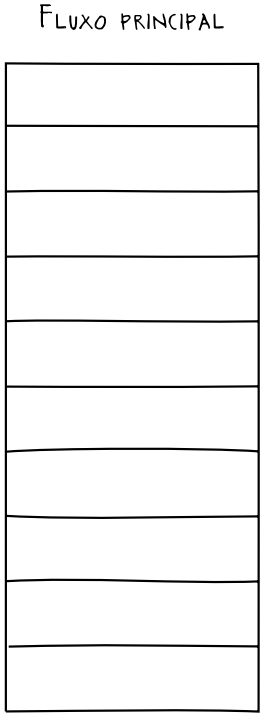
- São requisições assíncronas da plataforma (mascaráveis) para execução de rotinas de tratamento



As interrupções não mascaráveis (NMI) são usadas exclusivamente para condições de erro do hardware, mesmo que as interrupções não estejam habilitadas (MIE = 0)

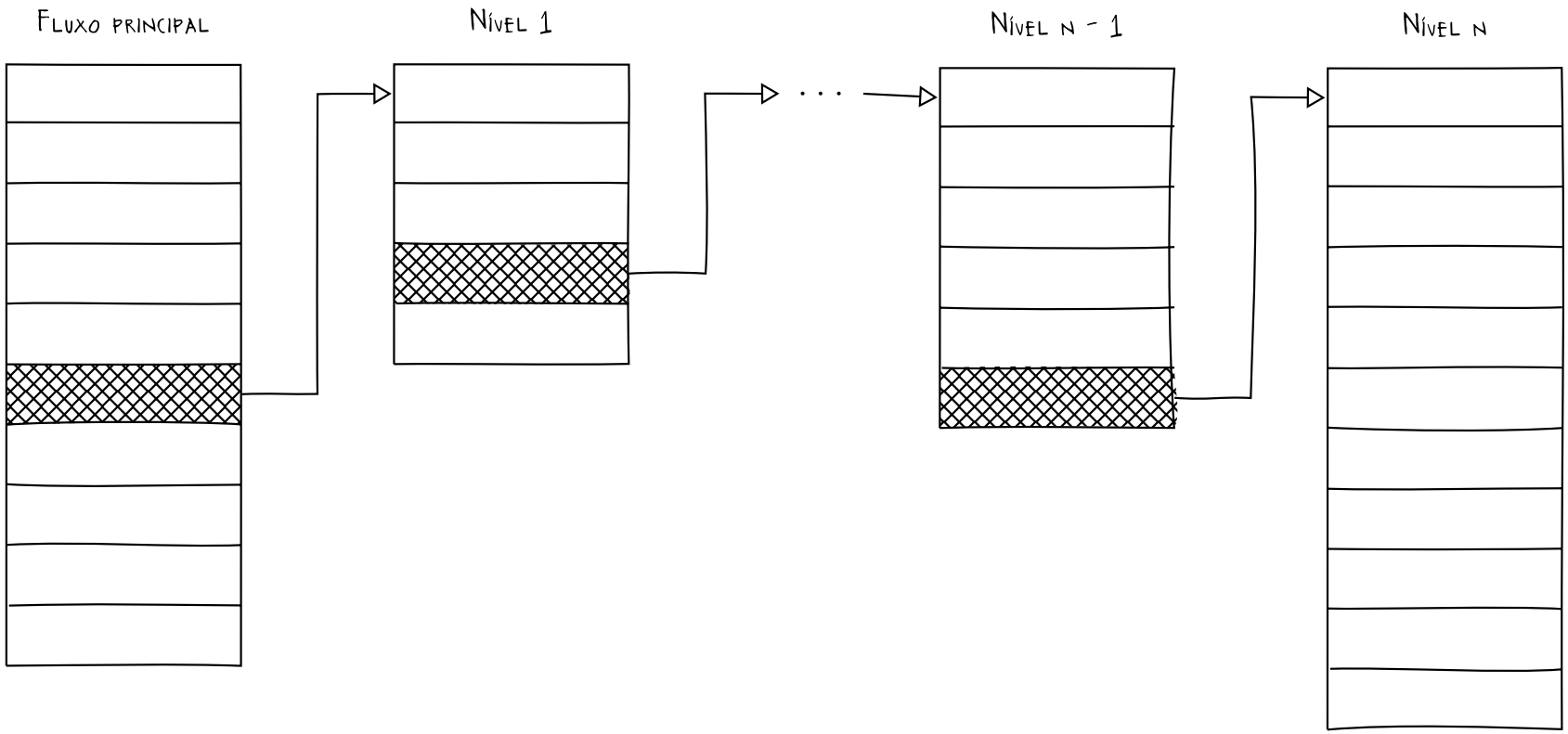
Priorização das interrupções

- Controlador de interrupções da plataforma (PLIC)



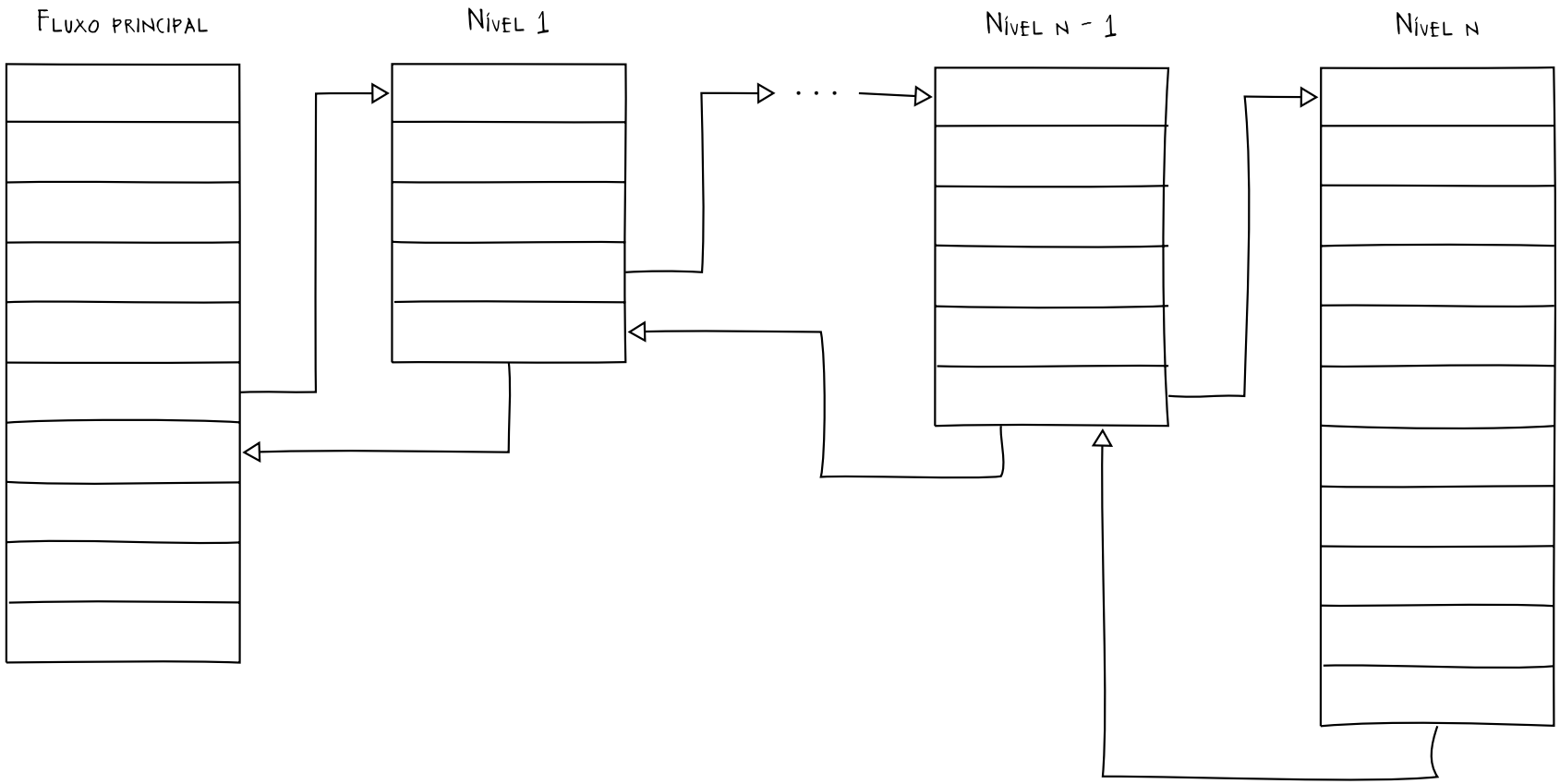
Priorização das interrupções

- Controlador de interrupções da plataforma (PLIC)



Priorização das interrupções

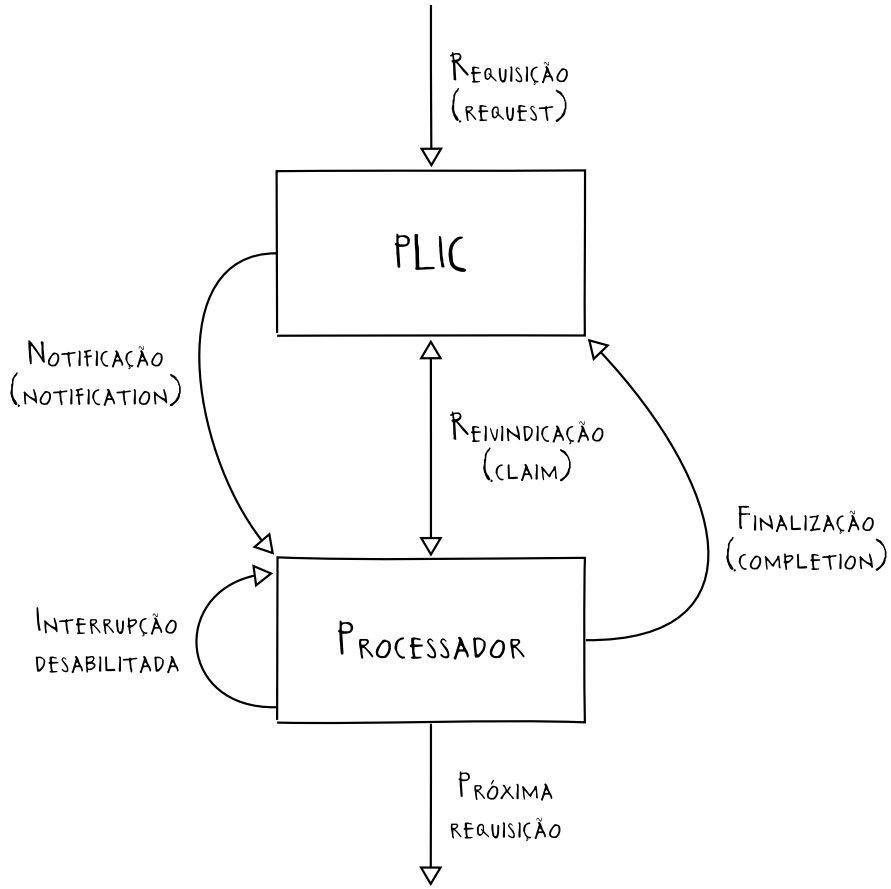
- Controlador de interrupções da plataforma (PLIC)



A interrupção nunca é gerada no nível 0

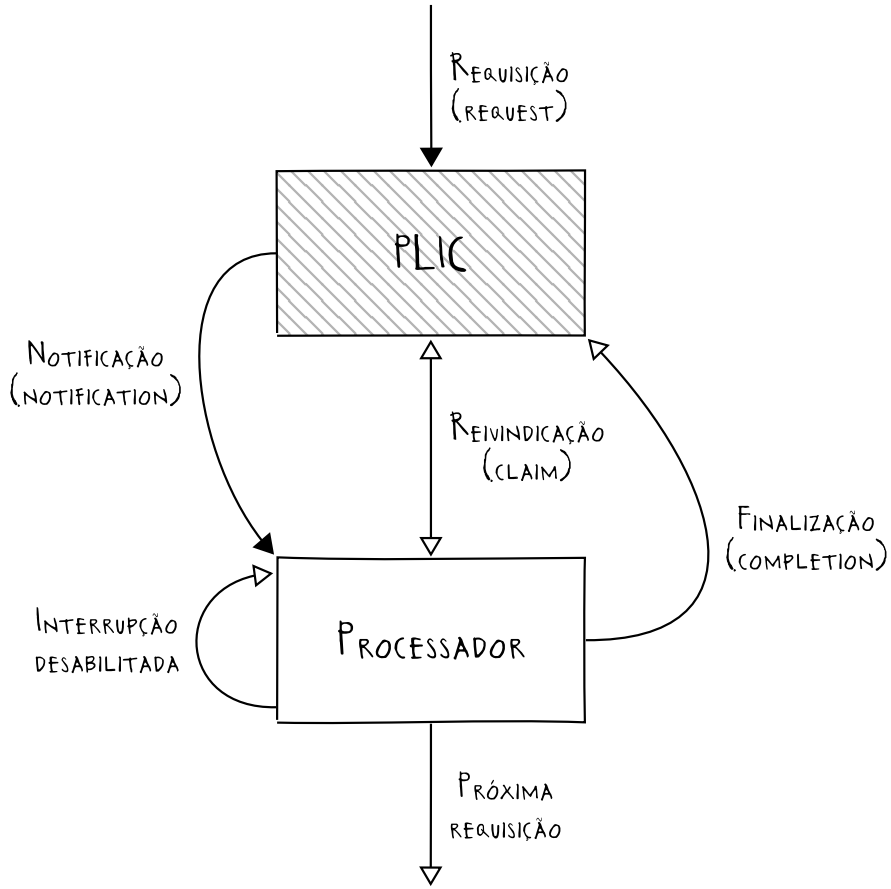
Processo de reinvidicação/finalização

- Controlador de interrupções da plataforma (PLIC)



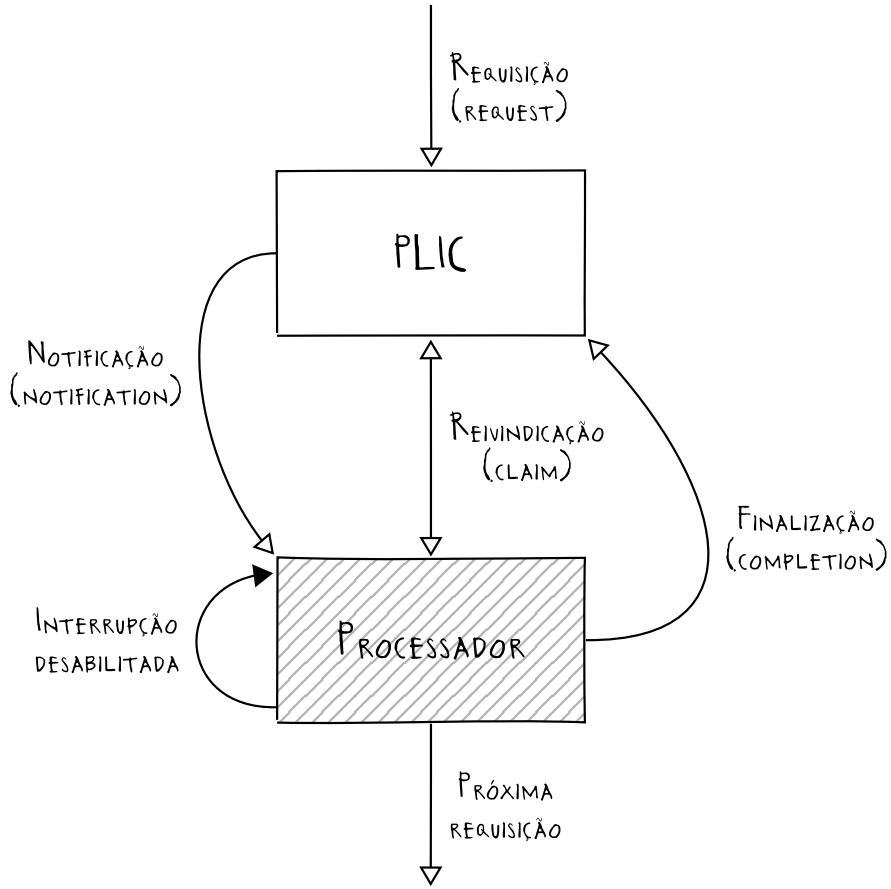
Processo de reinvidicação/finalização

- Controlador de interrupções da plataforma (PLIC)



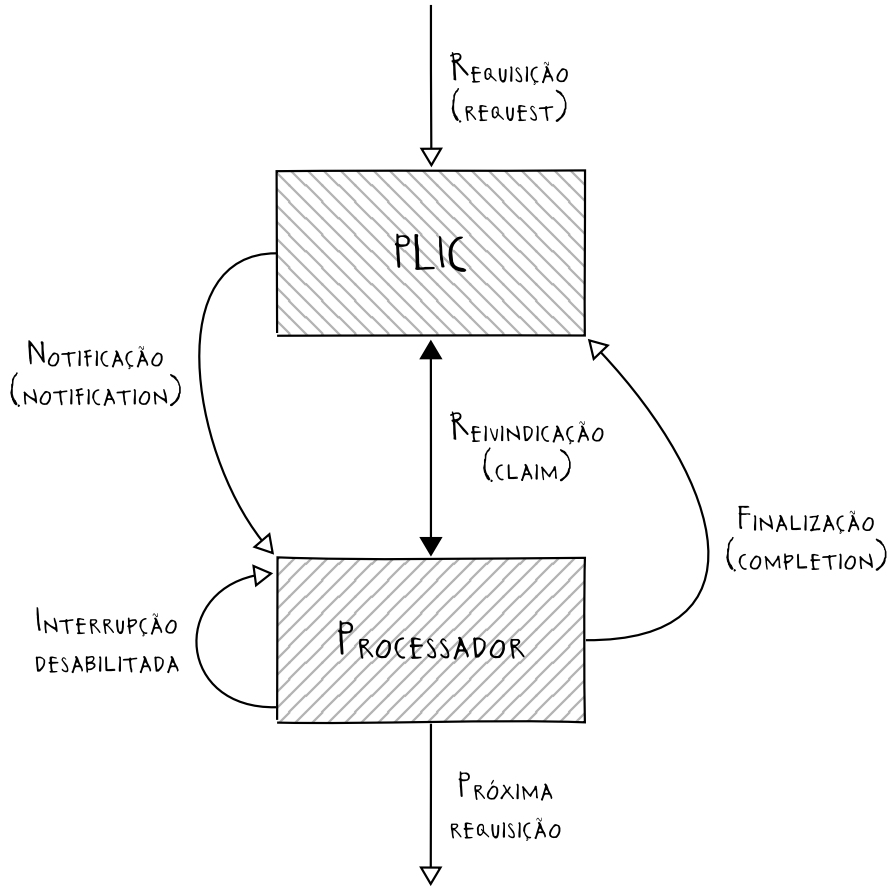
Processo de reinvidicação/finalização

- Controlador de interrupções da plataforma (PLIC)



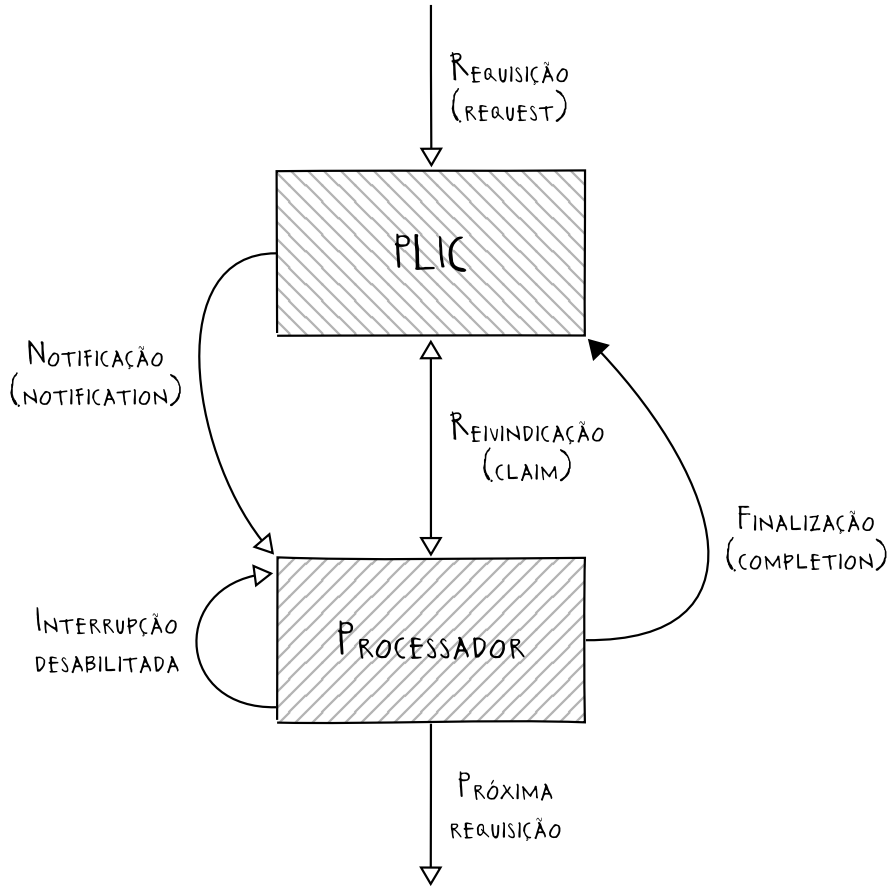
Processo de reinvidicação/finalização

- Controlador de interrupções da plataforma (PLIC)



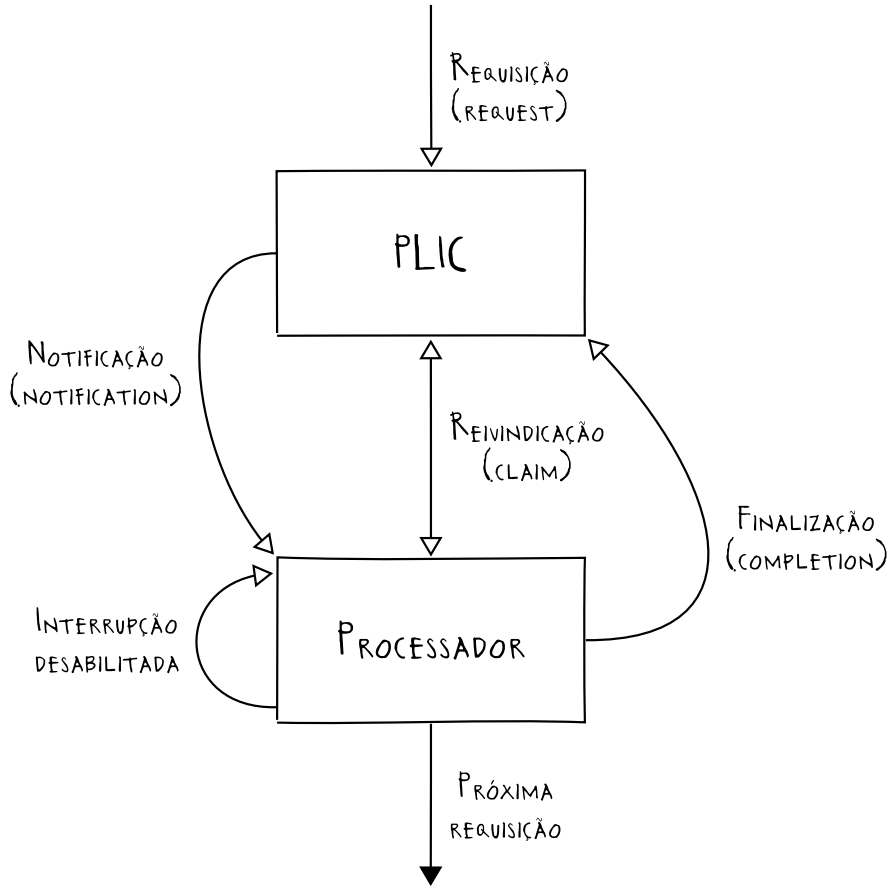
Processo de reinvidicação/finalização

- Controlador de interrupções da plataforma (PLIC)



Processo de reinvidicação/finalização

- Controlador de interrupções da plataforma (PLIC)



Exercício

- Implemente as instruções privilegiadas para manipulação do subconjunto de registradores de controle e de status (CSR), além da chamada para ambiente (*ecall*) e de retorno do nível de máquina (*mret*), com o objetivo de suportar o tratamento das seguintes exceções
  - *Instruction access fault* (1)
  - *Illegal instruction* (2)
  - *Load access fault* (5)
  - *Store access fault* (7)
  - *Environment call from M-mode* (11)