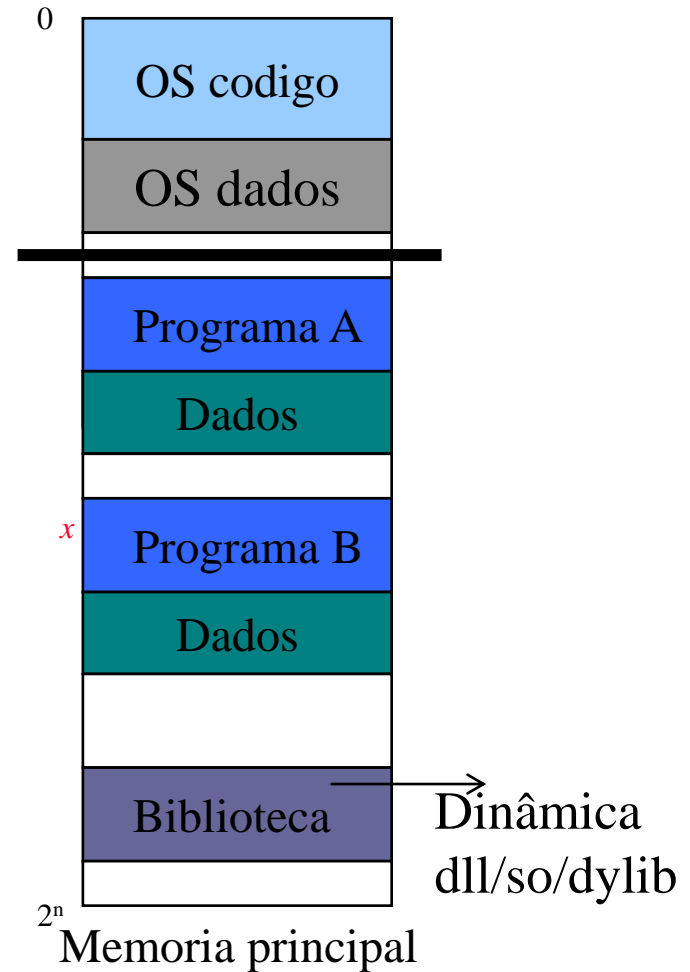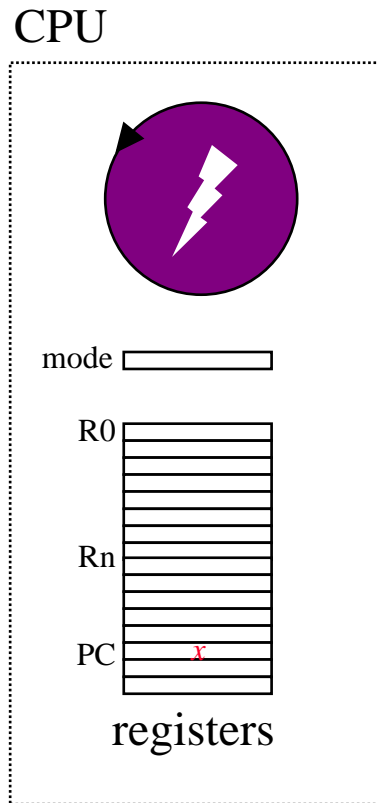# Um Kernel Protegido

Modo (register bit) indique se o CPU está a executar em modo *User* ou modo *Protegido*

User mode /Kernel Mode
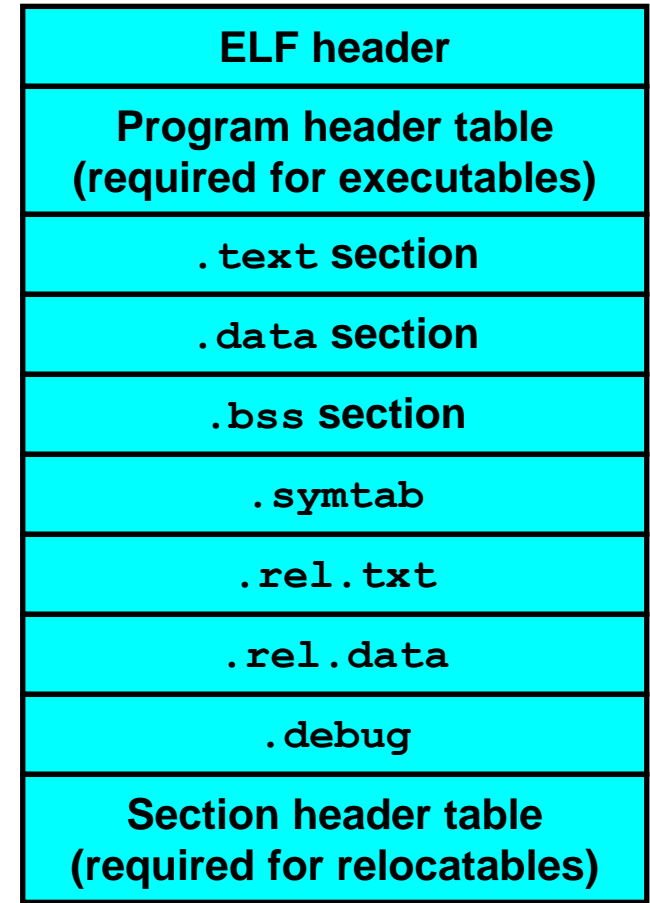
Alguns instruções e acessos a memoria são permitido apenas quando o CPU está em modo protegido

CPU

mode

R0

Rn

PC x

registers

0

OS codigo

OS dados

Programa A

Dados

x Programa B

Dados

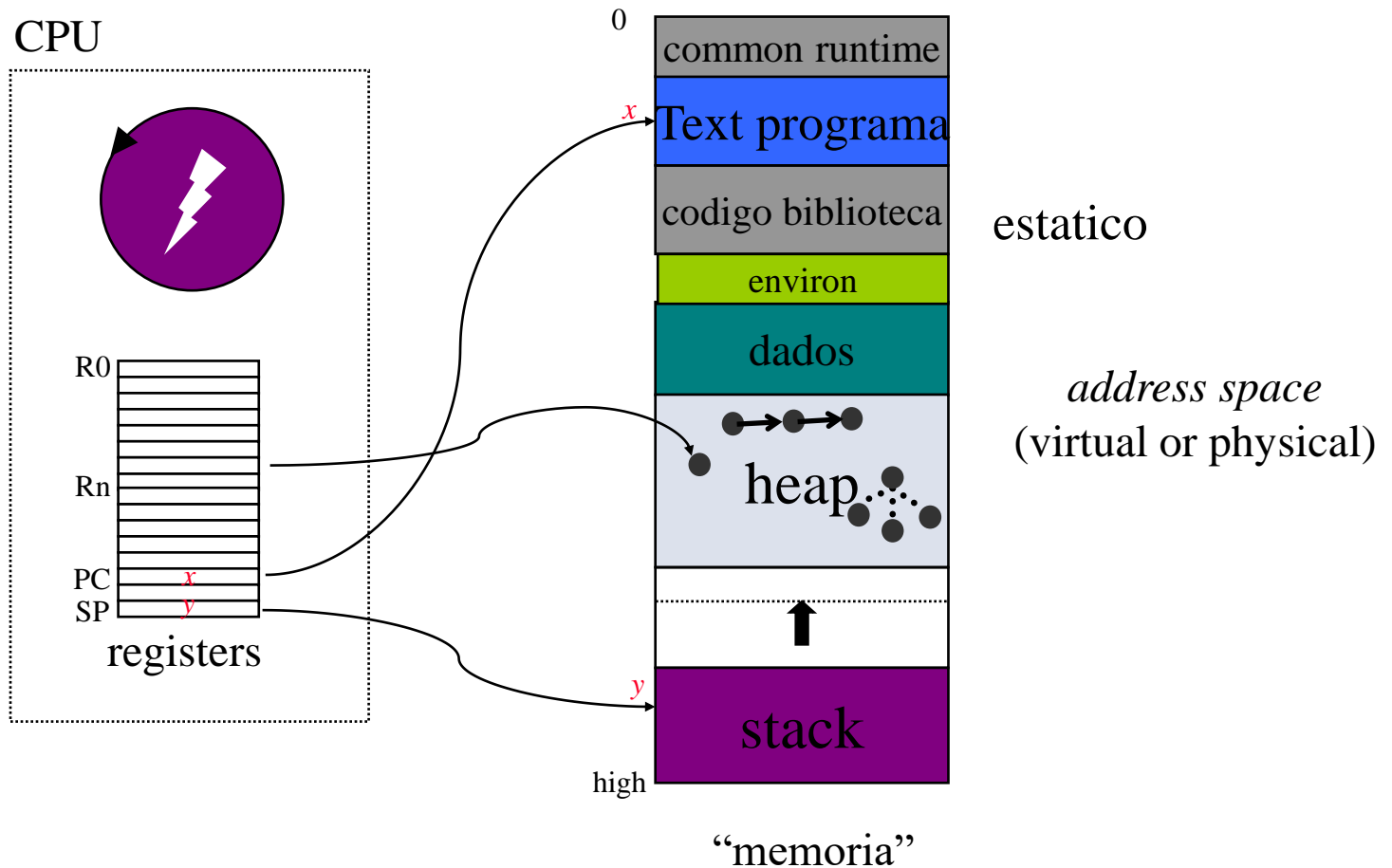Biblioteca Dinâmica dll/so/dylib

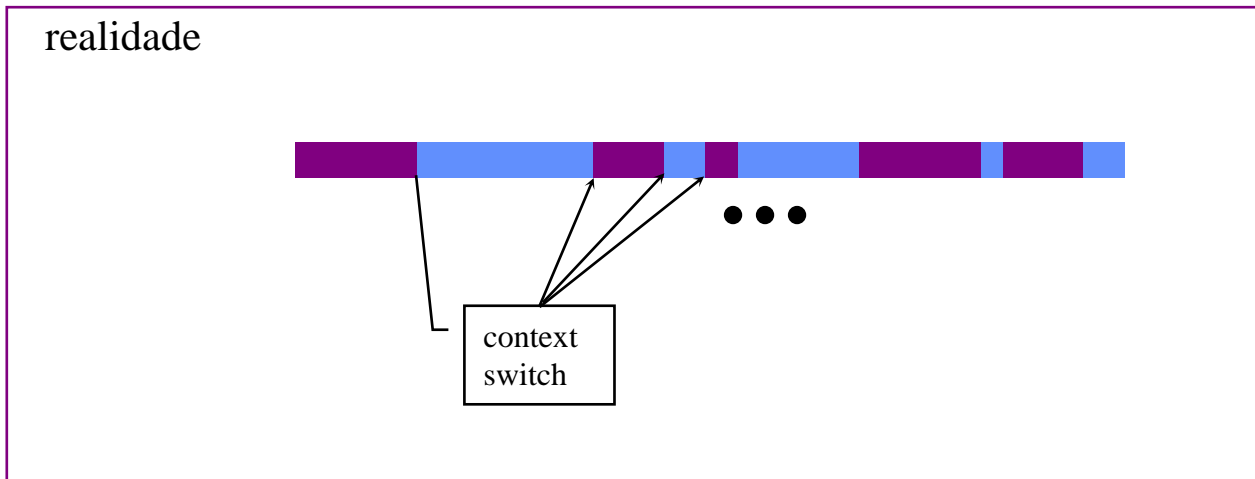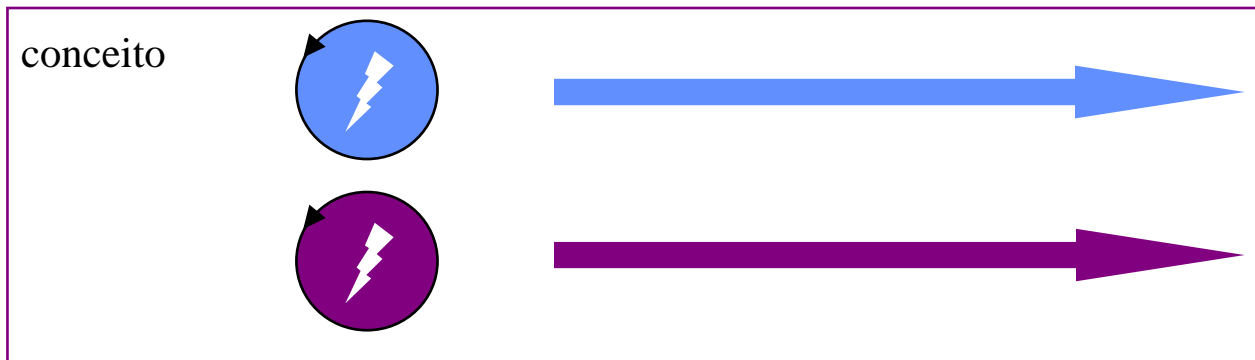2^n Memoria principal

Um Programa em Disco
ELF Object File Format

Elf header
- Magic number, type (.o, exec, .so), machine, byte ordering, etc.

Program header table
- Page size, virtual addresses memory segments (sections), segment sizes.

`.text` section
- Code

`.data` section
- Initialized (static) data

`.bss` section
- Uninitialized (static) data
- "Block Started by Symbol" or "Better Save Space"

`.symtab` section
- Symbol table
- Procedure and static variable names
- Section names and locations

`.rel.text` section
- Relocation info for `.text` section
- Addresses of instructions that will need to be modified in the executable and Instructions for modifying.

`.rel.data` section
- Relocation info for `.data` section
- Addresses of pointer data that will need to be modified in the merged executable

`.debug` section
- Info for symbolic debugging (`gcc -g`)

0

ELF header
Program header table (required for executables)
`.text` **section**
`.data` **section**
`.bss` **section**
`.symtab`
`.rel.txt`
`.rel.data`
`.debug`
Section header table (required for relocatables)

Um Programa em Execução

CPU

R0

Rn

PC   *x*

SP   *y*

registers

0

common runtime

x Text programa

codigo biblioteca   estatico

environ

dados

heap   *address space*
(virtual or physical)

stack

y

high

"memoria"

Dois Threads/Processos partilhando **um** CPU

conceito

realidade

context switch

Processo Context Switch

address space

0

common runtime

nano

y

codigo biblioteca

Environ $USER=A1

dados

stack

y

"memoria"

switch out

switch in

CPU

1. *save registers*

R0

Rn

PC   *x*
SP

registers

2. *load registers*

address space

0

common runtime

nano

x

codigo biblioteca

Environ$USER=B2

dados

high

stack

"memoria"

high
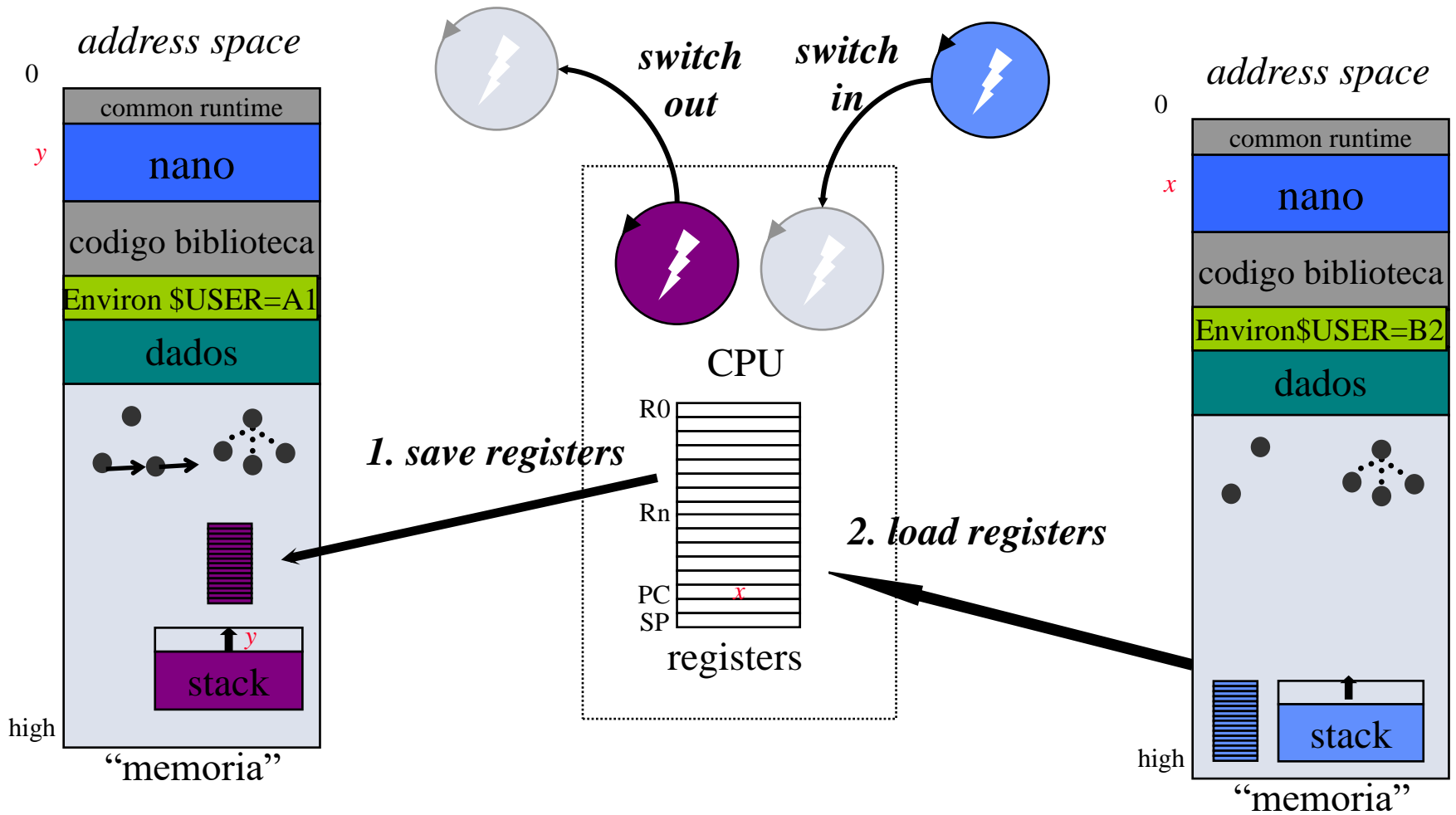
Um Programa com dois Threads

address space

pronto para executar

0

common runtime

x Text programa

codigo biblioteca

Environ $USER=A1

dados

Thread Em execução

CPU

R0

Rn

PC *x*

SP *y*

registers

stack *y*

stack

high

"memoria"

Thread Context Switch

switch out *switch in*

address space

0

common runtime

x Text programa

codigo biblioteca

dados

CPU

R0

Rn

PC   *x*
SP   *y*

registers

1. save registers

2. load registers

stack

stack

high

"memoria"