Embedded Systems Research Group

Licenciatura em Telecomunicações e Informática

Arquitetura e Tecnologia de Computadores
organização interna do computador

Paulo Cardoso

Grupo de Sistemas Embebidos
Departamento de Eletrónica Industrial Escola de Engenharia
Universidade do Minho

Sumário

Embedded Systems Research Group

- Componentes de um computador
- O modelo de von Neumann
- O ciclo do processador
- Linguagens de alto nível
- Sistemas computacionais
- Sistemas computacionais baseados em microcontroladores

Organização interna do computador
Componentes de um computador

Embedded Systems Research Group

- Componentes de um computador
  - Na sua essência, um computador é um sistema composto por três partes
    - Um microprocessador
      - Capaz de interpretar e executar instruções/programas
    - Uma memória
      - Para armazenar dados e programas
    - Entrada/saída
      - Um mecanismo para a transferência de dados de e para o mundo exterior
  - Modelo de von Neumann
    - Mais detalhes adiante ...

Organização interna do computador
Componentes de um computador

Embedded Systems Research Group

- O que diz o dicionário (generalista)
  - Microprocessador
    - Circuito integrado complexo que efetua as operações básicas de um microcomputador
      Fonte: dicionário da Porto Editora
  - Microcomputador
    - Computador de pequenas dimensões cujo órgão central é um microprocessador
      Fonte: dicionário da Porto Editora
    - Computador
      - Aparelho eletrónico que é capaz de receber, armazenar e processar grande quantidade de informação em função de um conjunto de instruções com que é programado
        Fonte: dicionário da Porto Editora

Organização interna do computador
Componentes de um computador

Embedded Systems Research Group

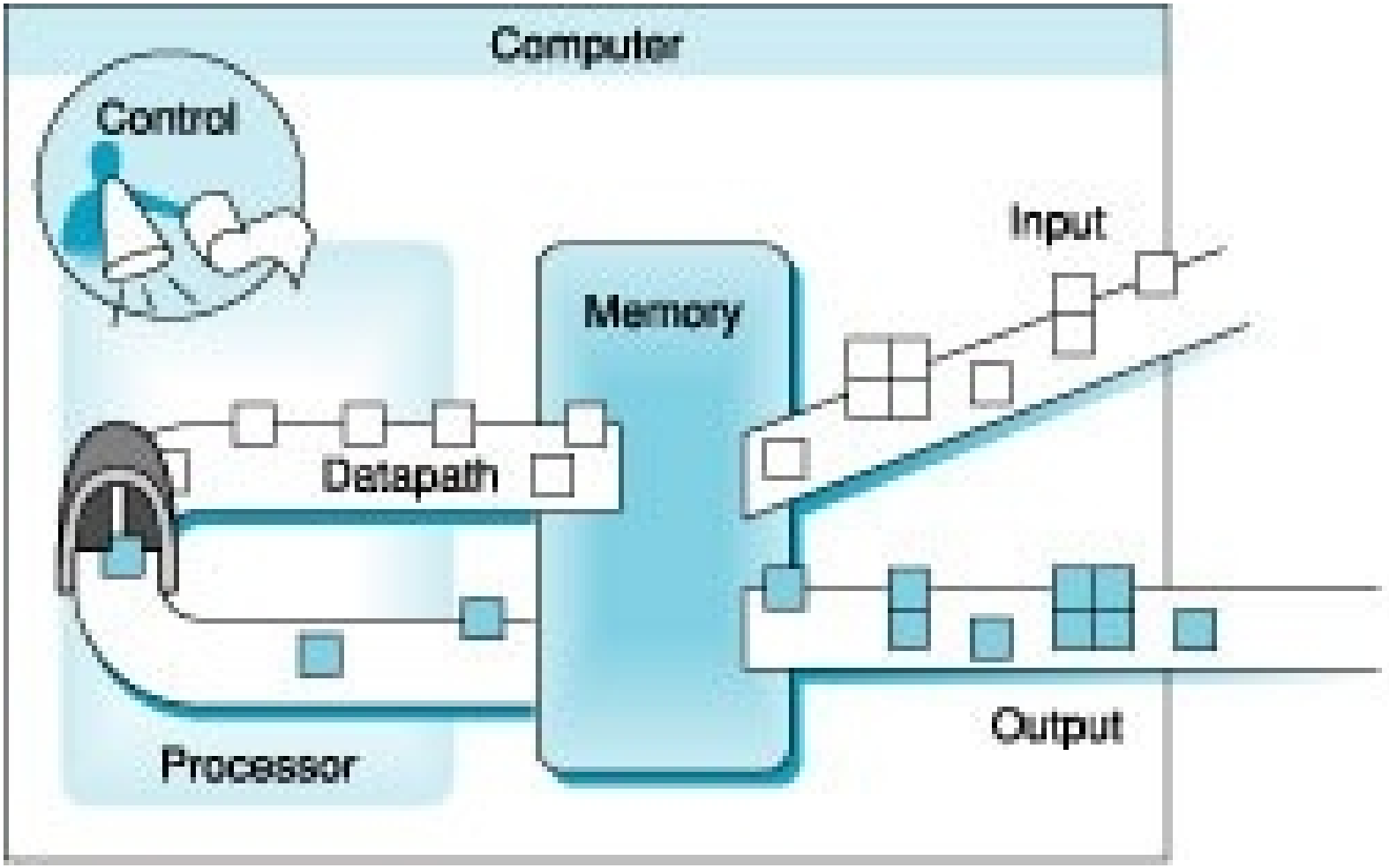
- Palavras chave
  - Microprocessador
    - Circuito integrado complexo
  - Microcomputador
    - receber
    - armazenar
    - processar
      - conjunto de instruções
        - programado

- Componentes de um computador
  - Microprocessador
    - Unidade lógica/aritmética e unidade de controlo
  - Entrada
    - rato, teclado, ...
  - Saída
    - monitor, impressora, ...
  - Memória
    - Cache, RAM, disco, CD
    - Onde são armazenadas as instruções e os dados

Organização interna do computador
Componentes de um computador

Embedded Systems Research Group

- Componentes de um computador



Fonte: David A. Patterson, John L. Hennessy, “Computer Organization and Design”

O processador divide-se em unidade de controlo, unidade lógica e aritmética e datapath (espécie de tapete rolante que transporta da memoria para o processador e vice-versa)

Organização interna do computador

Componentes de um computador

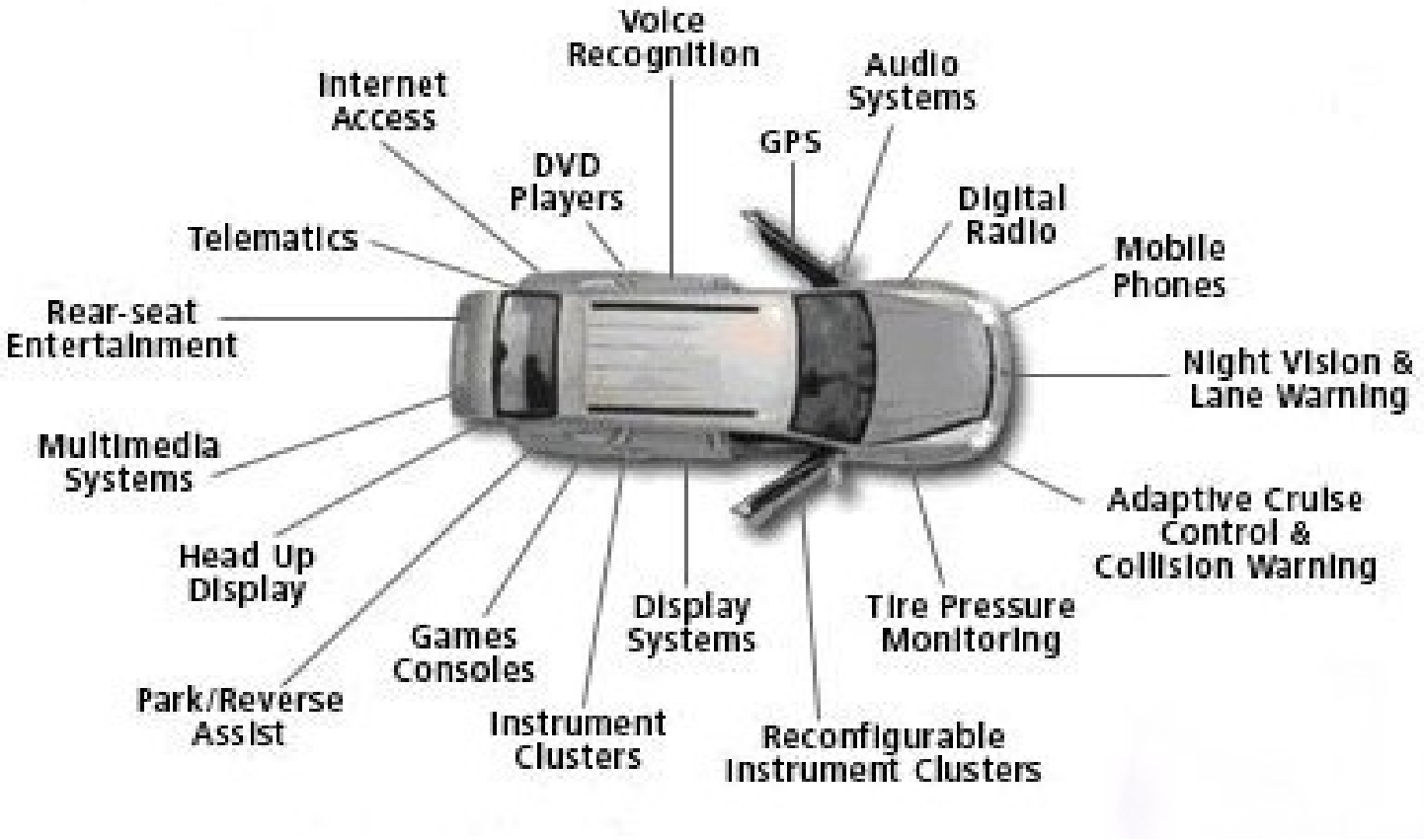
Embedded Systems Research Group

- Mas o que é um computador?
  - Diferentes tipos
    - embebidos, laptop, desktop, servidor
  - Diferentes utilizações
    - Processador de texto, ...
    - Casas, carros, televisões, telefones, ...
  - Diferentes fabricantes
    - Intel, AMD, IBM, Atmel, MicroChip...
  - Diferentes tecnologias de suporte e custos

Organização interna do computador
Componentes de um computador

Embedded Systems Research Group

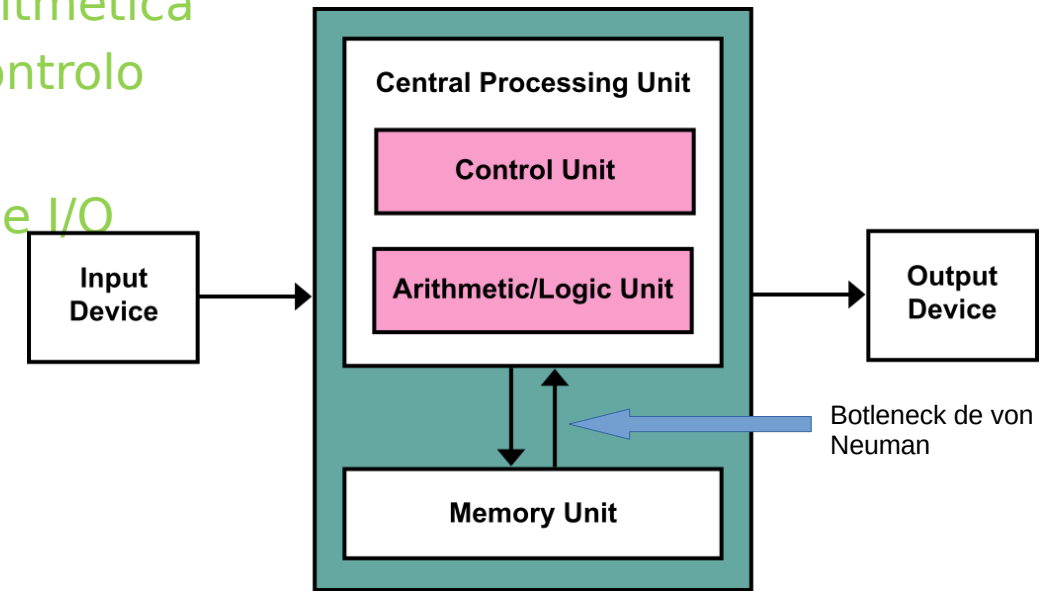
- Nem sempre vemos os



Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- O modelo de von Neumann
  - Em 1945 publicou um relatório
    - “First Draft of a Report on the EDVAC”
      - O documento descreve a arquitetura de um computador com os seguintes componentes
        - Unidade de aritmética
        - Unidade de controlo
        - Memória
        - Mecanismos de I/O



Fonte: https://en.wikipedia.org/wiki/Von_Neumann_architecture

Botleneck de von Neumann → é um gargalho porque existe apenas um caminho para a passagem dos dados e/ou instruções da memória para o processador.

Para mitigar este efeito utilizamos a memoria cache que estando “colado” ao processador o seu acesso é muito mais rápido.

O seu carregamento na primeira instrução dá-se de forma igual, utilizando o botleneck, mas são carregadas também todas as instruções da vizinhança (que provavelmente também serão necessárias devido à arquitetura da unidade de memoria).

Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- Define o conceito de **stored program**
  - Instruções (programas) e dados são armazenados na memória sob a forma de números
    - Sendo assim fácil a sua alteração
    - Levando ao conceito de **stored program computer**
    - No ENIAC (computador anterior onde Von Newmann colaborou)
      - Toda a programação era ao nível da lógica digital
        - A programação do computador envolvia mover fichas e fios
        - Era necessária uma configuração de hardware diferente para resolver cada tipo de problema

Conceito mais importante do modelo de Von Neumann a par do seu modelo de computador

Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- É o paradigma sob o qual assenta a noção de computador atual
  - Atualmente os computadores stored programs têm as seguintes características
    - Três sistemas de hardware
      - Uma unidade de processamento central
        - Contendo o processamento e controlo
      - Um sistema de memória principal
      - Um sistema de I / O
    - A capacidade de realizar o processamento sequencial de instruções
    - Um caminho único entre o CPU e a memória
      - Bottleneck de von Neumann

Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- Os convencionais computadores stored program
  - Tiveram avanços incrementais ao longo do tempo
    - Incluindo a adição de barramentos especializados, unidades de virgula flutuante, memórias cache, . . .
    - Alguns sistemas atuais possuem barramentos separados para dados e instruções
      - Designando-se arquiteturas Harvard

O nosso micro não consegue realizar cálculos de virgula flutuante por hardware apenas os consegue fazer por software (demorando muito mais para tal).

O 8051 só tem 8bits (1byte)

Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- Mais poder computacional exige outras abordagens
  - Instruction-level parallelism
    - Paralelismo dentro do processador (superscalar processor)
      - Pipelining → processador super escalado (é como se fosse uma linha de produção em que em cada instante esta mais de uma instrução dentro do processador para acelerar o processo)
      - Duplicação de unidades funcionais do processador
  - Hardware-multithreading
    - Múltiplas threads partilham as unidades funcionais do processador

Organização interna do computador

O modelo de von Neumann

Embedded Systems Research Group

- Mais poder computacional exige outras abordagens
  - Instruction-level parallelism
    - Paralelismo dentro do processador (superscalar processor)
      - Pipelining
      - Duplicação de unidades funcionais do processador
  - Hardware-multithreading
    - Múltiplas threads partilham as unidades funcionais do processador

Organização interna do computador

O modelo de von Neumann

Embedded Systems Research Group

- Mais poder computacional exige outras abordagens
  - Computação paralela/distribuída/concorrente
  - Aceleradores
    - GPUs (Graphics Processing Units)
    - Hardware especializado
  - ...

Organização interna do computador

O modelo de von Neumann

Embedded Systems Research Group

- O modelo prevê a existência de um conjunto de registos específicos contendo nomeadamente

Endereços de memoria discreta

Program counter →

- PC: Endereço de memória da próxima instrução
- MAR: Endereço da instrução atual/dado a ser carregada
- MDR: Dado recebido/a ser enviado para memória
- CIR: Instrução a ser executada
- ACC: Dado em processamento/resultado

Organização interna do computador
O modelo de von Neumann

- Atualmente os sistemas não contêm alguns destes registos especiais
  - Têm no entanto alguns registos genéricos que são usados como memória para diminuir os acessos à memória

Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- Representação

A cada ciclo de relógio o processador vai buscar à memoria a próxima instrução.

Ir buscar → FETCH
Ler a instrução → DECODE
Executar → Execute

Os dados e os programas estão na memoria principal no modelo de von Neuman.

O 8051 usa uma memoria para os dados (memoria de barras) e uma memoria separada para os programas: a memoria de dados é igual à memoria RAM

Memoria RAM → memoria volátil que necessita de energia para estar “viva”

Memoria flahs (ex:. Pen) → não necessita de energia para guardar os dados

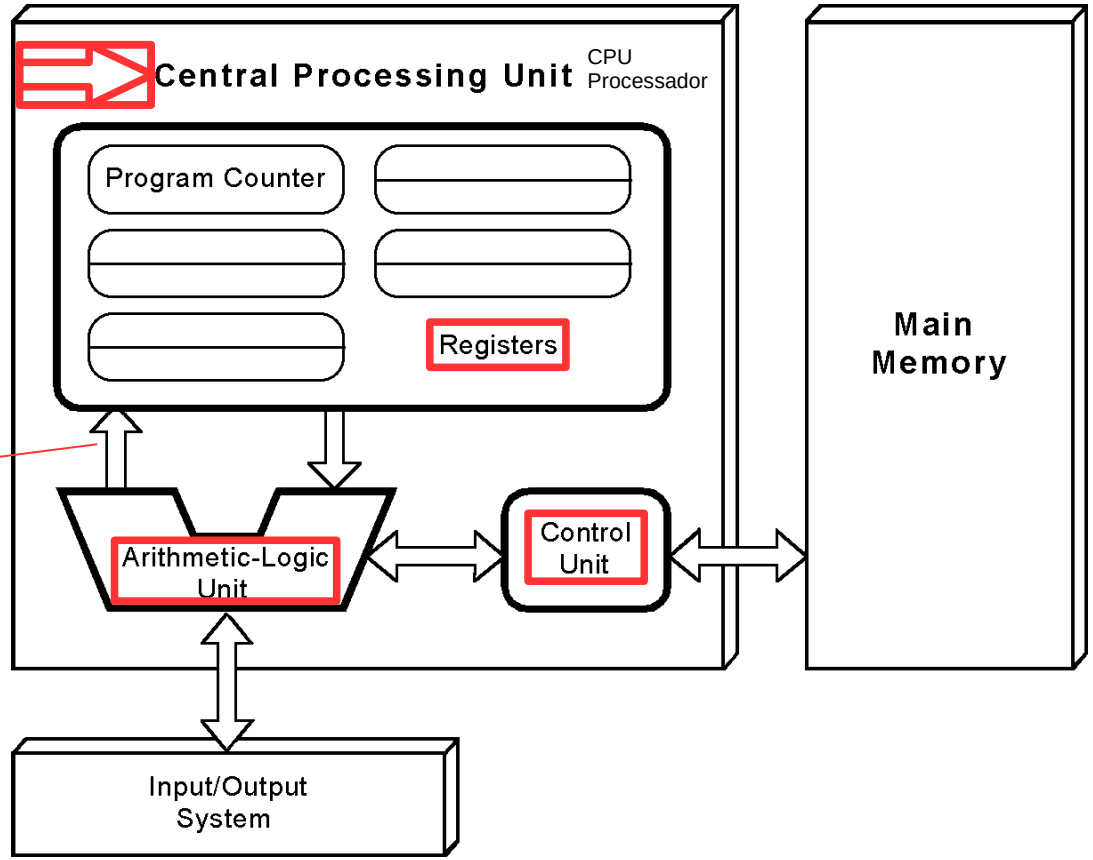
SSD → memoria não volátil evolução dos discos mecânicos

Memoria de programas do 8051 (não volátil) é como se fosse um SSD de 64k.

Vantagens:
Aceder aos dados e ao programa ao mesmo tempo (impossível no modelo de von Neumann)
Visto que os micro são usados em sistemas industriais onde podem haver cortes de energia é importante que não seja necessário reprogramar o micro sempre que a luz volta-se

Datapath

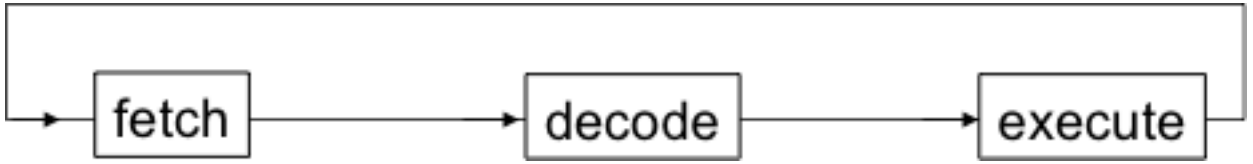
A esta separação de memoria chama-se modelo de Harvad



Organização interna do computador
O modelo de von Neumann

Embedded Systems Research Group

- Execução das instruções feita em três passos
  - Fetch
  - Decode
  - Execute

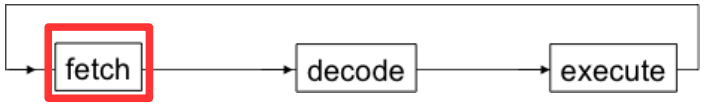


- O processador executa indefinidamente estes três passos
  - O ciclo do processador

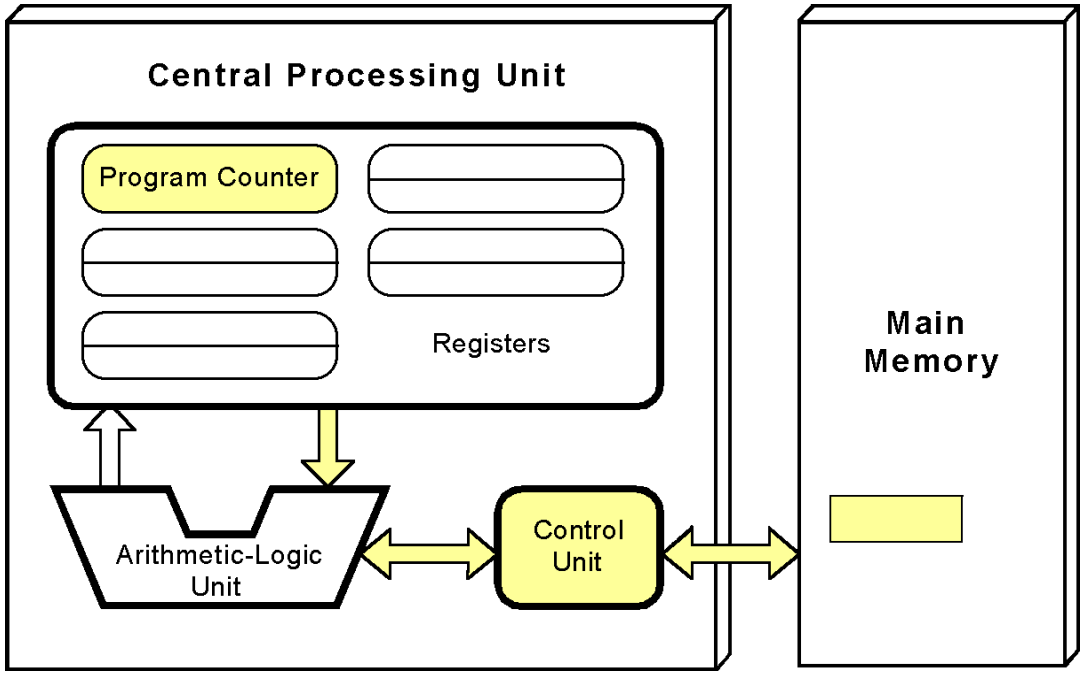
Organização interna do computador
O ciclo do processador

Embedded Systems Research Group

- Fetch



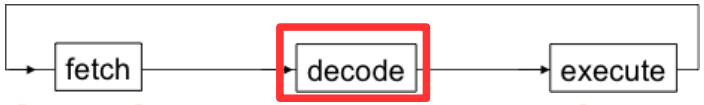
  - A unidade de controlo obtém a próxima instrução
    - Usando o program counter para determinar a sua localização na memória



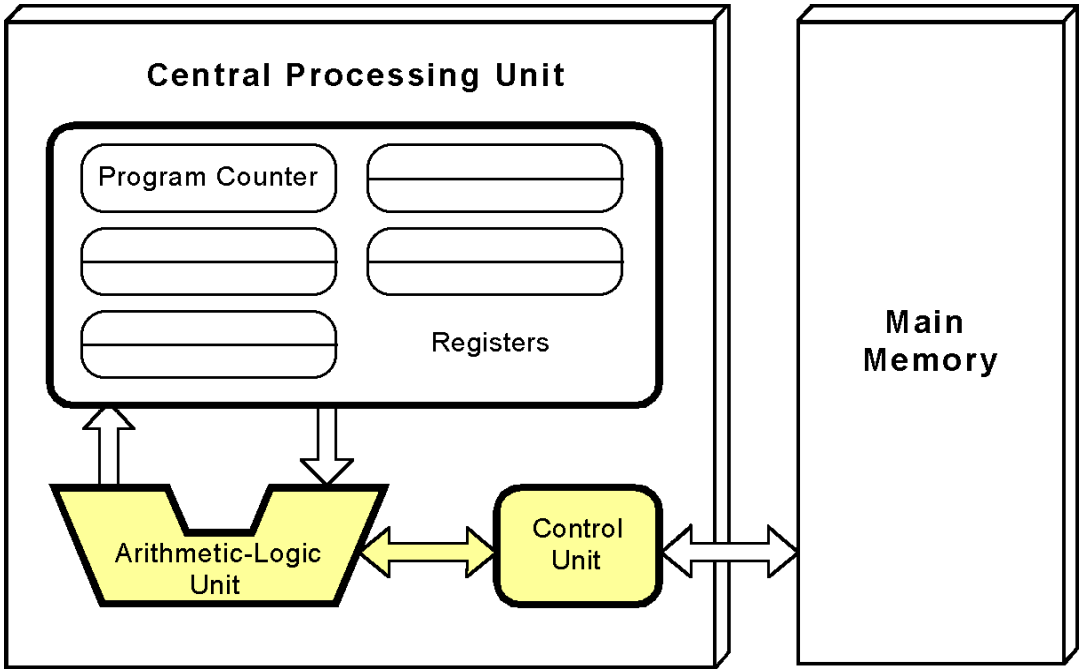
Organização interna do computador
O ciclo do processador

Embedded Systems Research Group

- Decode



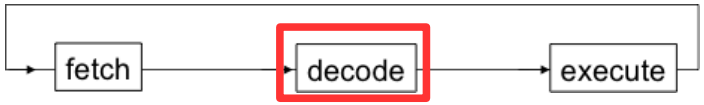
  - A unidade de controlo descodifica a instrução e
    - Gera os sinais que controlam o funcionamento da ALU



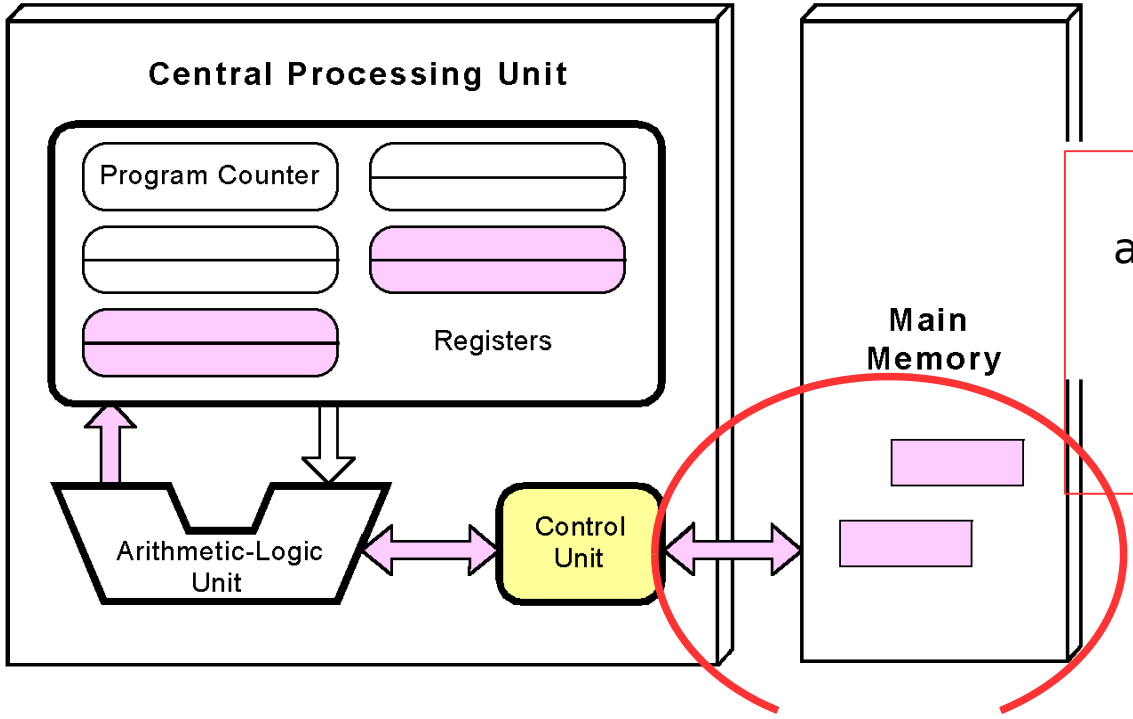
Organização interna do computador
O ciclo do processador

Embedded Systems Research Group

- Decode



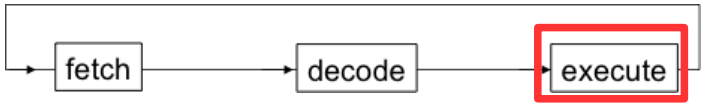
  - A unidade de controlo descodifica a instrução e
    - Seleciona os registos/memória a entrar na ALU



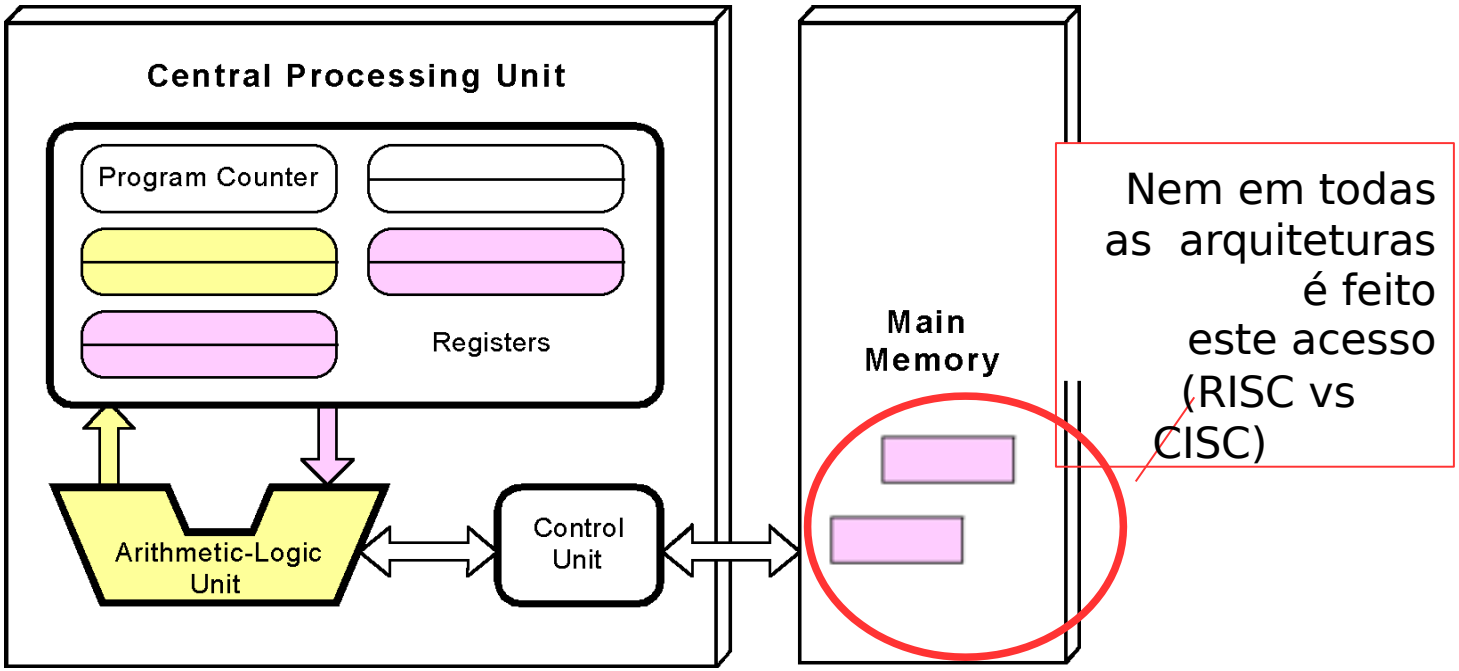
Organização interna do computador
O ciclo do processador

Embedded Systems Research Group

- Execute



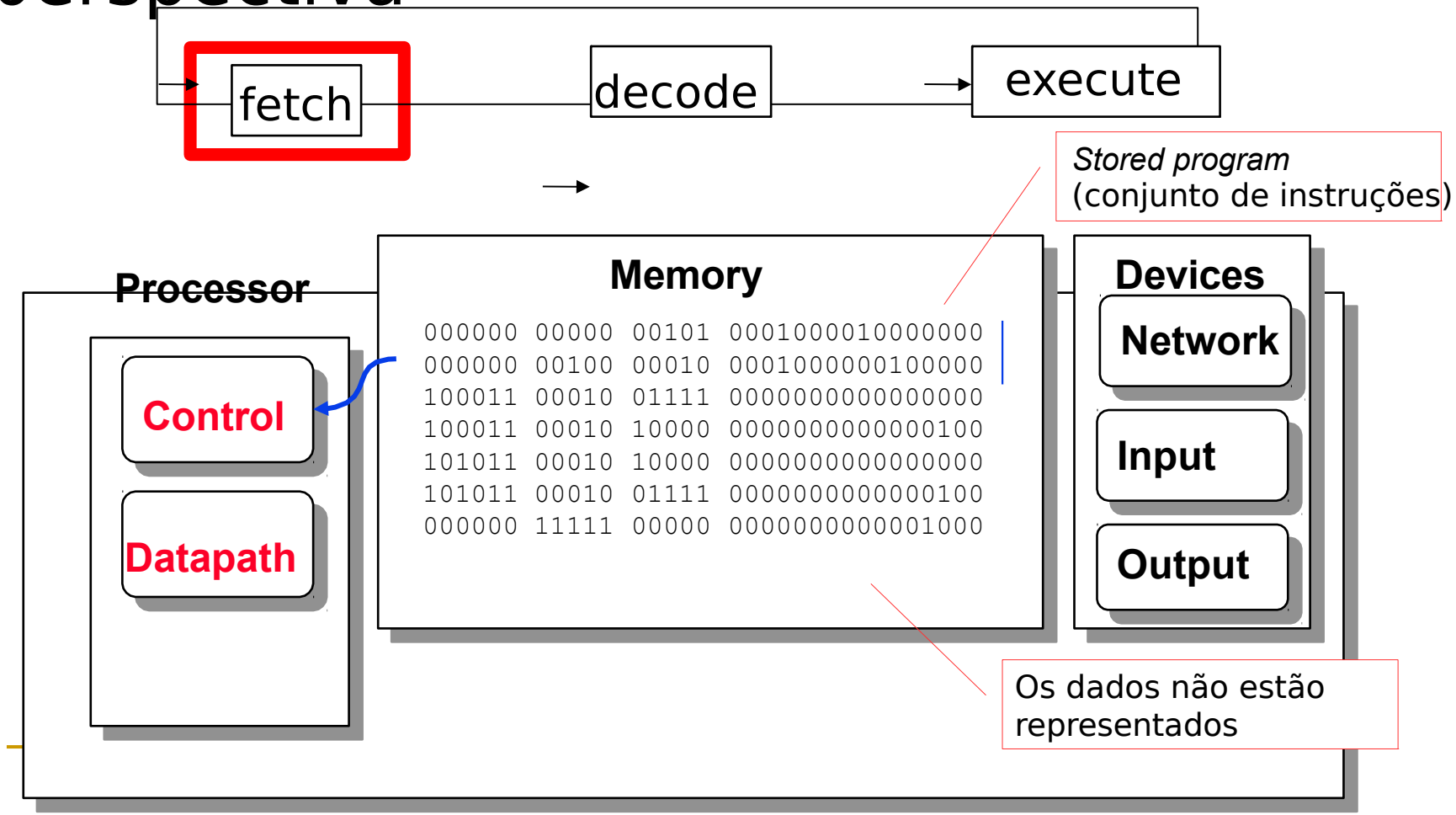
  - A ALU executa a instrução e
    - Coloca os resultados nos registos ou memória



Organização interna do computador
O ciclo do processador

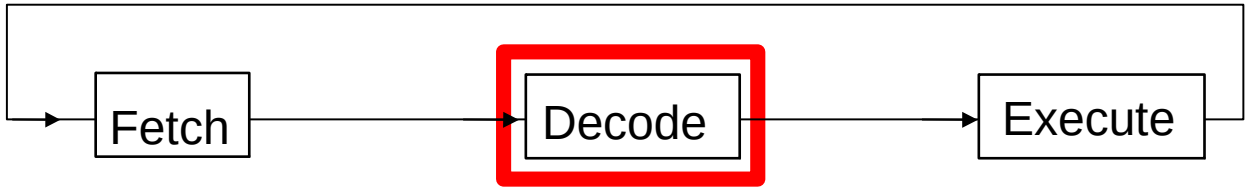
Embedded Systems Research Group

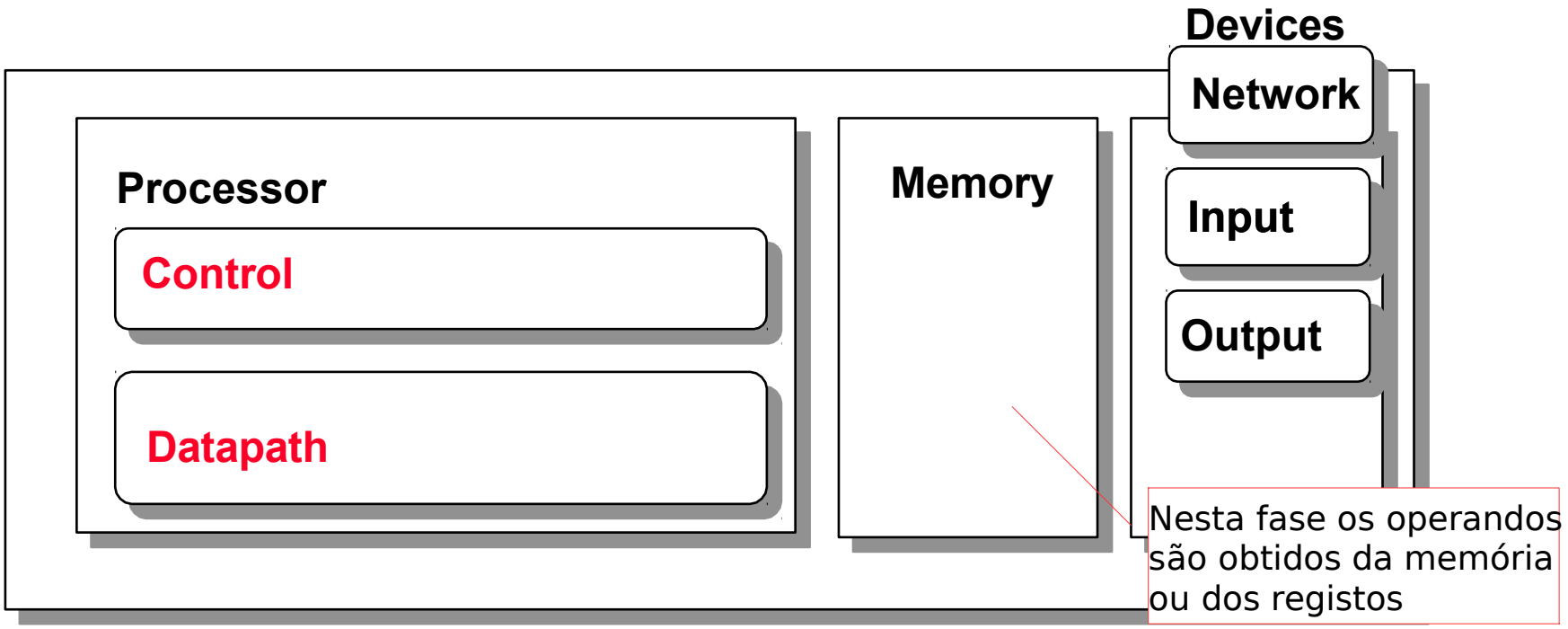
- O ciclo do processador – outra perspectiva



Organização interna do computador
O ciclo do processador

- O ciclo do processador – outra perspectiva

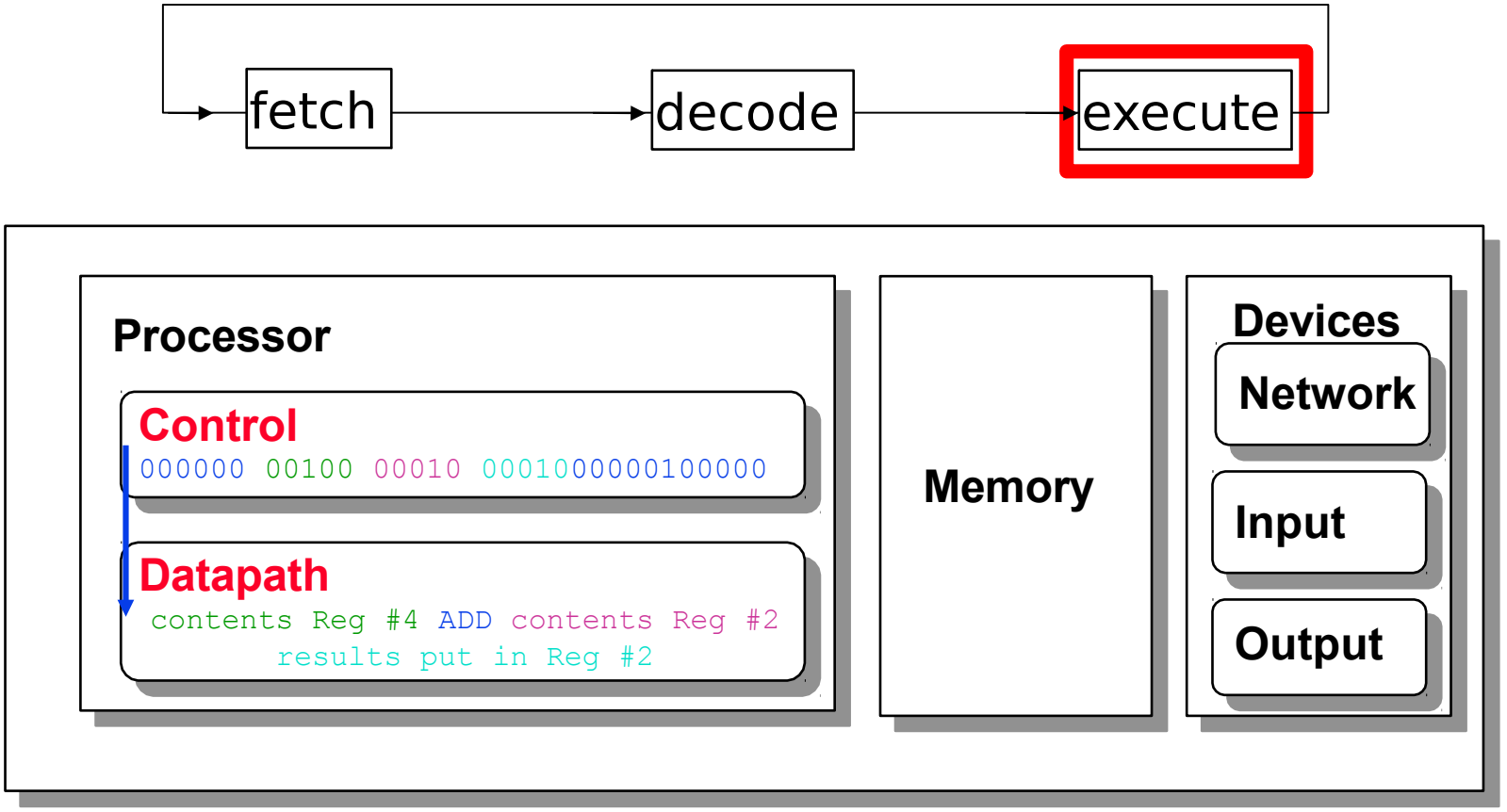




Organização interna do computador
O ciclo do processador

Embedded Systems Research Group

- O ciclo do processador – outra perspectiva



Organização interna do computador

Linguagens de alto nível

Embedded Systems Research Group

- O que é o “stored program“?
  - É o programa que está a executar
    - São conjuntos de bits com instruções para o processador
      - O processador só entende bits
      - Linguagem binária

Organização interna do computador

Linguagens de alto nível

Embedded Systems Research Group

- Escrita dos programas
  - Em linguagem binária?
    - A linguagem dos humanos é de muito mais alto nível
  - Em linguagem assembly?
    - Ainda é muito baixo nível...
  - Porque não fazer os programas na “linguagem humana”?
    - Linguagens de programação de alto nível
      - C, Pascal, Ada, ...

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Vantagens da linguagens de alto nível
  - Programador pensa numa linguagem “natural”
  - Aumenta a produtividade do programador
    - Código mais legível e mais fácil de testar e depurar
  - Melhor capacidade de manutenção do programa
  - Portabilidade dos programas entre arquiteturas
    - No caso dos microcontroladores exige algum cuidado
      - Devido às extensões da linguagem
      - Diferenças de implementação entre periféricos
  - Otimizações dos compiladores
    - Produzem código assembly muito eficiente para a arquitetura alvo

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Programa em linguagem de alto nível (C)

```
swap (int v[], int k)
    {  int temp;
temp = v[k];  v[k]
    = v[k+1];
    v[k+1] = temp;
}
```

um para muitos

compilador C

- Programa em linguagem assembly

```
swap:   sll     $2, $5, 2
        add             $2, $4, $2
        lw              $15, 0($2)
        lw              $16, 4($2)
        sw              $16, 0($2)
        sw              $15, 4($2)
        jr              $31
```

um para um

assembler

- Código máquina (objecto)

```
000000 00000 00101 0001000010000000
000000 00100 00010 0001000000100000
.  .  .
```

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Programa em linguagem de alto nível (C)

```
swap (int v[], int k)
    {  int temp;
temp = v[k];  v[k]
    = v[k+1];
    v[k+1] = temp;
}
```

um para muitos

compilador C

- Programa em linguagem assembly

```
swap:  sll   $2, $5, 2
       add   $2, $4, $2
       lw    $15, 0($2)
       lw    $16, 4($2)
       sw    $16, 0($2)
       sw    $15, 4($2)
       jr    $31
```

um para um

assembler

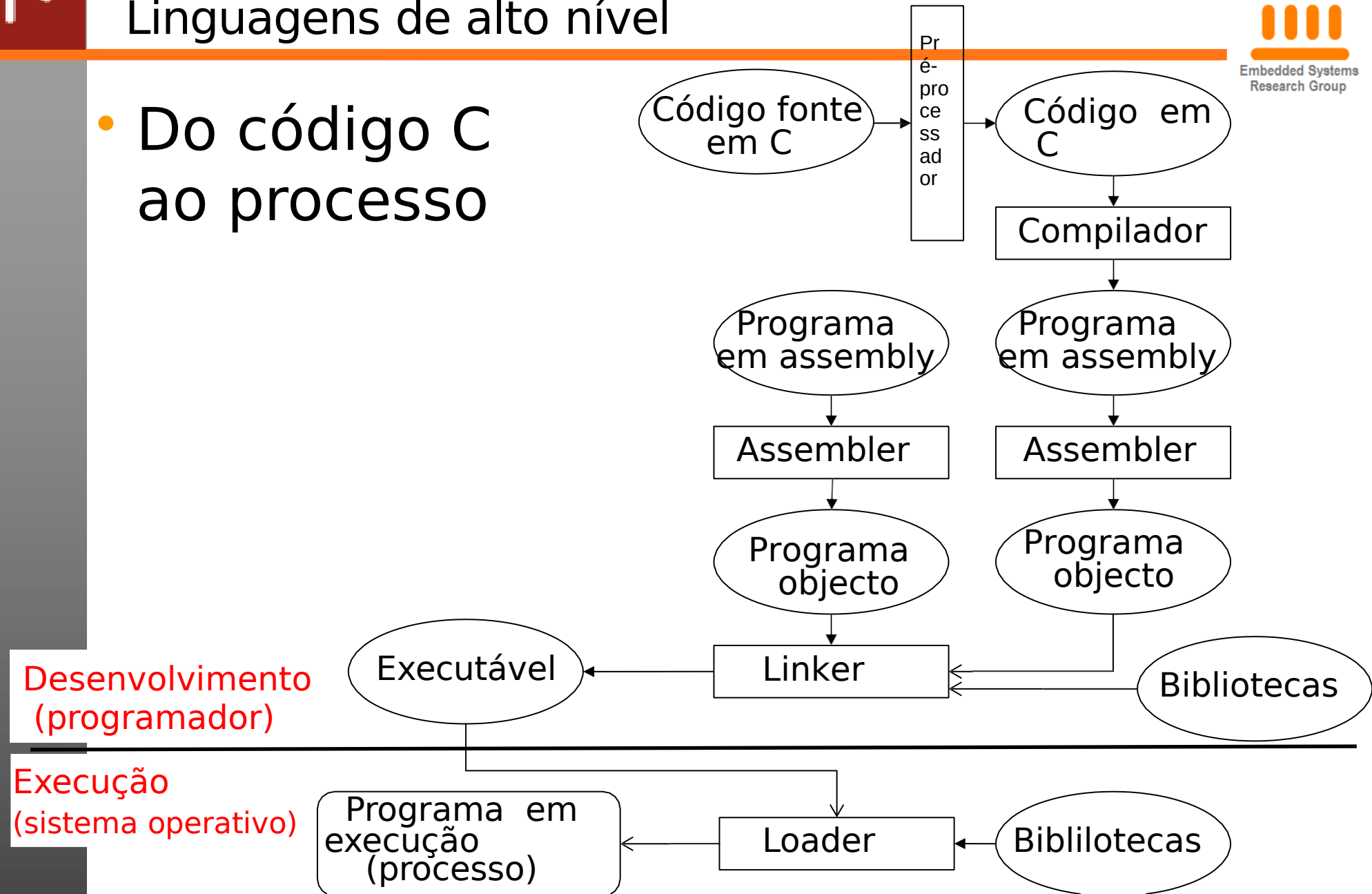
- Código máquina (objeto)

```
000000 00000 00101 000100 0010000000
000000 00100 00010 0001000000100000
. . .
```

Organização interna do computador

Linguagens de alto nível

- Do código C ao processo



Organização interna do computador

Linguagens de alto nível

Embedded Systems Research Group

- Pré-processador
  - Processador de macros
    - Passa ao compilador um ficheiro de código C
      - Sem qualquer diretiva ou macro
        - Apenas código C

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Compilador
  - Transforma o código alto nível em assembly
  - Os optimizadores dos compiladores atuais
    - Produzem código quase tão bom como o de um expert em assembly
      - Muitas vezes melhor, no caso de um grande programa

Organização interna do computador

Linguagens de alto nível

Embedded Systems Research Group

- Assembler
  - Gera um ficheiro objeto (código máquina)
    - Criado a partir do código fonte (assembly)
      - Mas não é executável (chamado código relocatable)
        - Constrói uma tabela de símbolos associando as etiquetas (labels) aos respetivos endereços
        - Resolve as referências a dados e funções definidos dentro do ficheiro
        - Não resolve as referências a dados e funções definidos fora do ficheiro
          - Definidos noutro ficheiro de código fonte ou em bibliotecas

Organização interna do computador

Linguagens de alto nível

Embedded Systems Research Group

- Assembler
  - Geração do ficheiro objeto
    - Cria um ficheiro objeto a partir do código assembly
      - Código relocatable
        - Cujos endereços só são atribuídos em link time
          - Relocation é o processo de ligar referências simbólicas a definições simbólicas (papel do linker)
        - Contém uma lista de
          - Nomes e endereços de símbolos visíveis externamente
          - Nomes e referências de símbolos que necessitam de relocation

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Linker
  - Liga (Link) os vários ficheiros objeto
    - Junta o código objeto dos vários ficheiros (módulos)
      - Utiliza as tabelas de símbolos para resolver as invocações noutros módulos
      - Para os símbolos que não ficarem resolvidos
        - Procura nas bibliotecas fornecidas pelo compilador

Organização interna do computador
Linguagens de alto nível

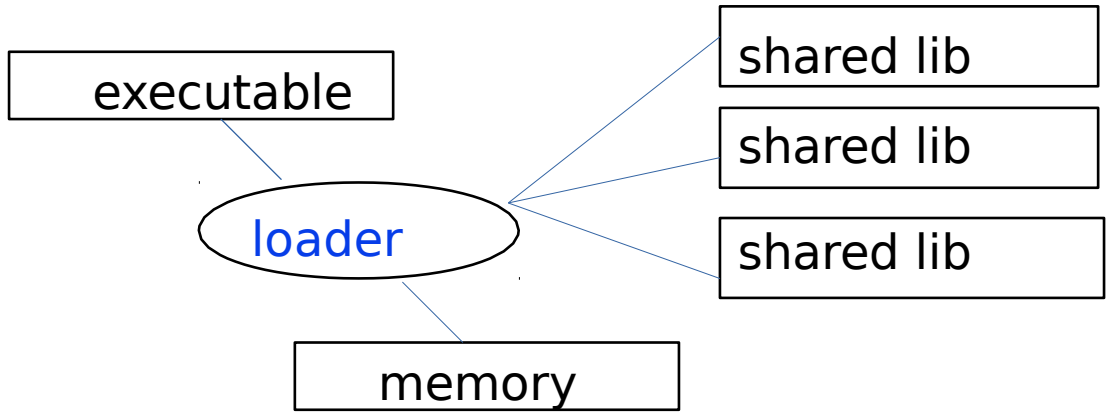
Embedded Systems Research Group

- Linker
  - Liga (Link) os vários ficheiros objeto
    - Junta o código objeto dos vários ficheiros (módulos)
      - Utiliza as tabelas de símbolos para resolver as invocações noutros módulos
        - Para os símbolos que não ficarem resolvidos
          - Procura nas bibliotecas fornecidas pelo compilador

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Loader
  - Corre dentro do S.O.
    - Carrega o programa para ser executador
      - Bem como bibliotecas dinâmicas que possam ser necessárias



Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Para além da compilação
  - Existem outras técnicas de conversão de código de alto nível
    - Interpretação
      - Um “interpretador” traduz e executa uma linha de cada vez
        - O programa é sempre reinterpretado sempre que executa novamente
      - Para executar os programas
        - O interpretador é sempre carregado para memória
    - A interpretação “pura” atualmente não existe
      - Máquina virtual
        - Neste contexto é um programa que virtualiza uma aplicação
        - Não um programa que virtualiza um sistema
          - VMWare, VirtualBox, Qemu (virtualizam o hardware)

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

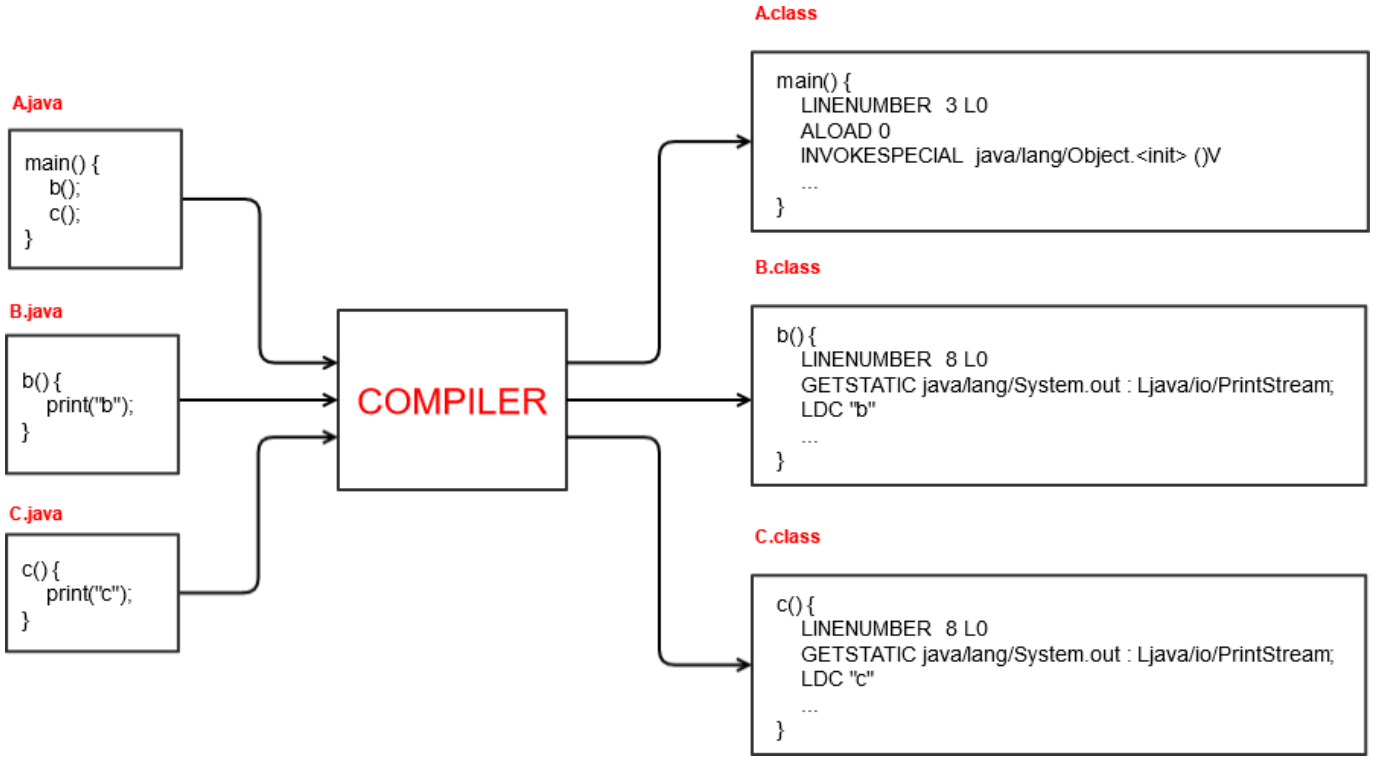
- Linguagem Java
  - É traduzida para uma linguagem intermédia (compilador)
    - Byte code
  - É executada numa máquina virtual (interpretador)
    - Java Virtual Machine (JVM)
      - Traduz e executa cada instrução

Fonte: https://www.baeldung.com/java-compiled-interpreted

Organização interna do computador

Linguagens de alto nível

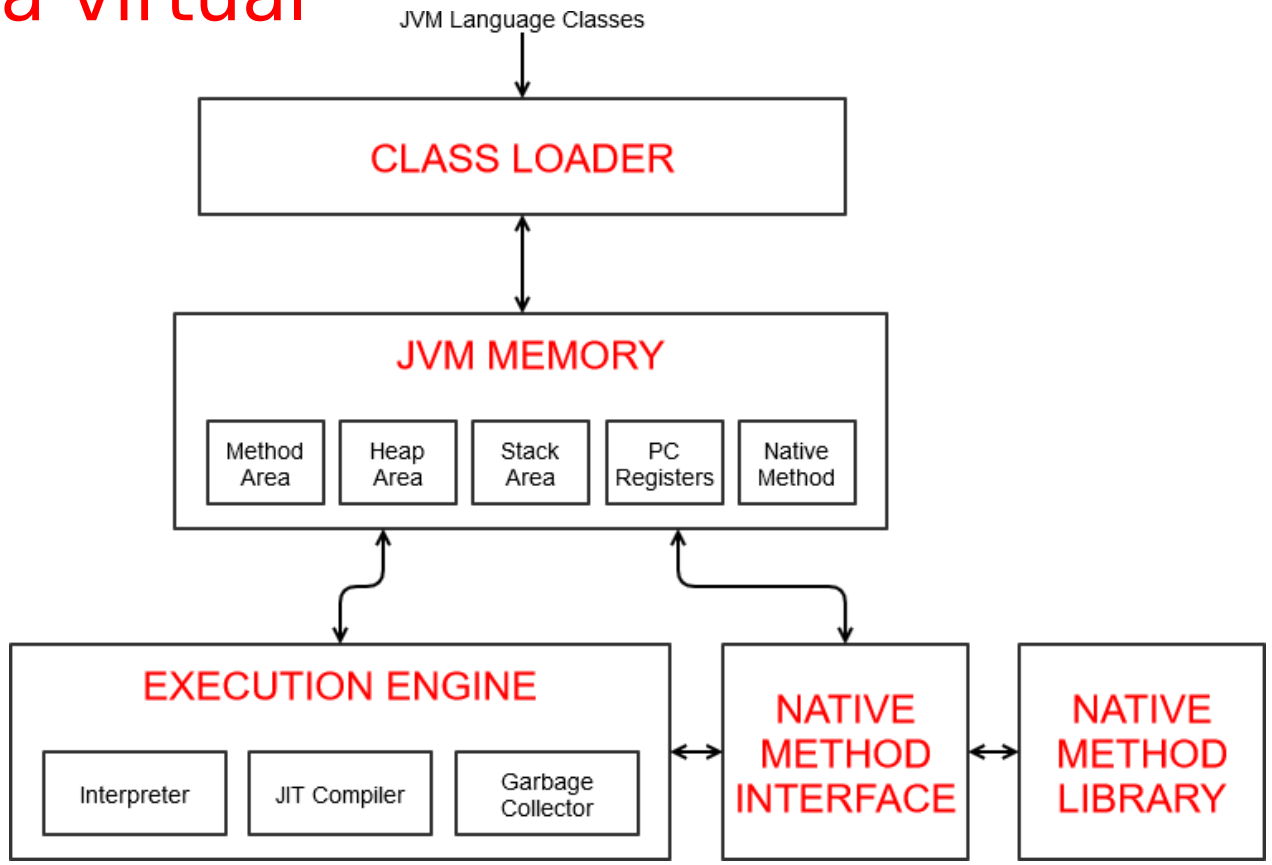
Embedded Systems Research Group



Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Linguagem Java
  - Máquina virtual

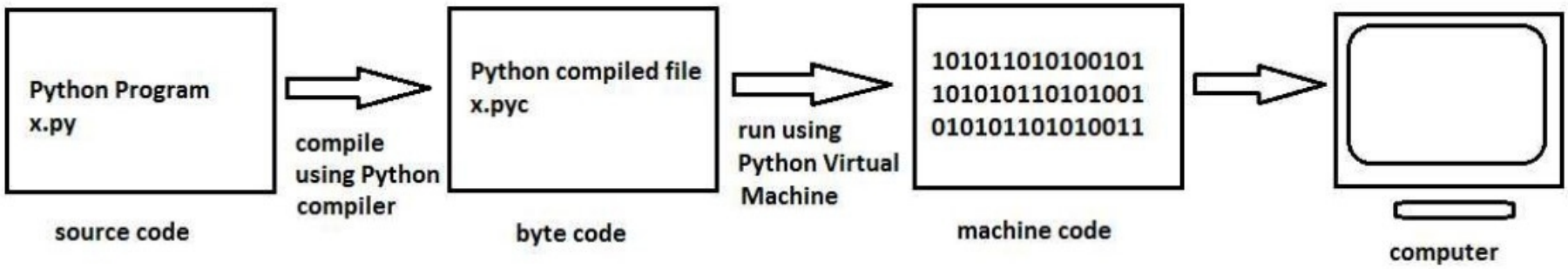


Fonte: https://www.baeldung.com/java-compiled-interpreted

Organização interna do computador
Linguagens de alto nível

Embedded Systems Research Group

- Linguagem Python
  - Compilação
  - Máquina virtual



Fonte: https://
www.gkindex.com/python-tutorial/python-virtual-machine.jsp

Organização interna do computador
Virtualização

Embedded Systems Research Group

- Virtualização
  - Está na base da computação na cloud
    - Versão de um recurso baseada em software
      - E.g.: Computação; rede; armazenamento;
- Hipervisor
  - Estão na base da virtualização
    - Gestor de recursos de uma máquina virtual
      - Tipo 1 (base metal)
        - Server virtualization
          - E.g.: VMWare ESXi; Microsoft Hyper-V; KVM (oper source)
      - Tipo 2 (OS hosted)
        - End user virtualization
          - E.g. Oracle Virtualbox; VMWare Workstation

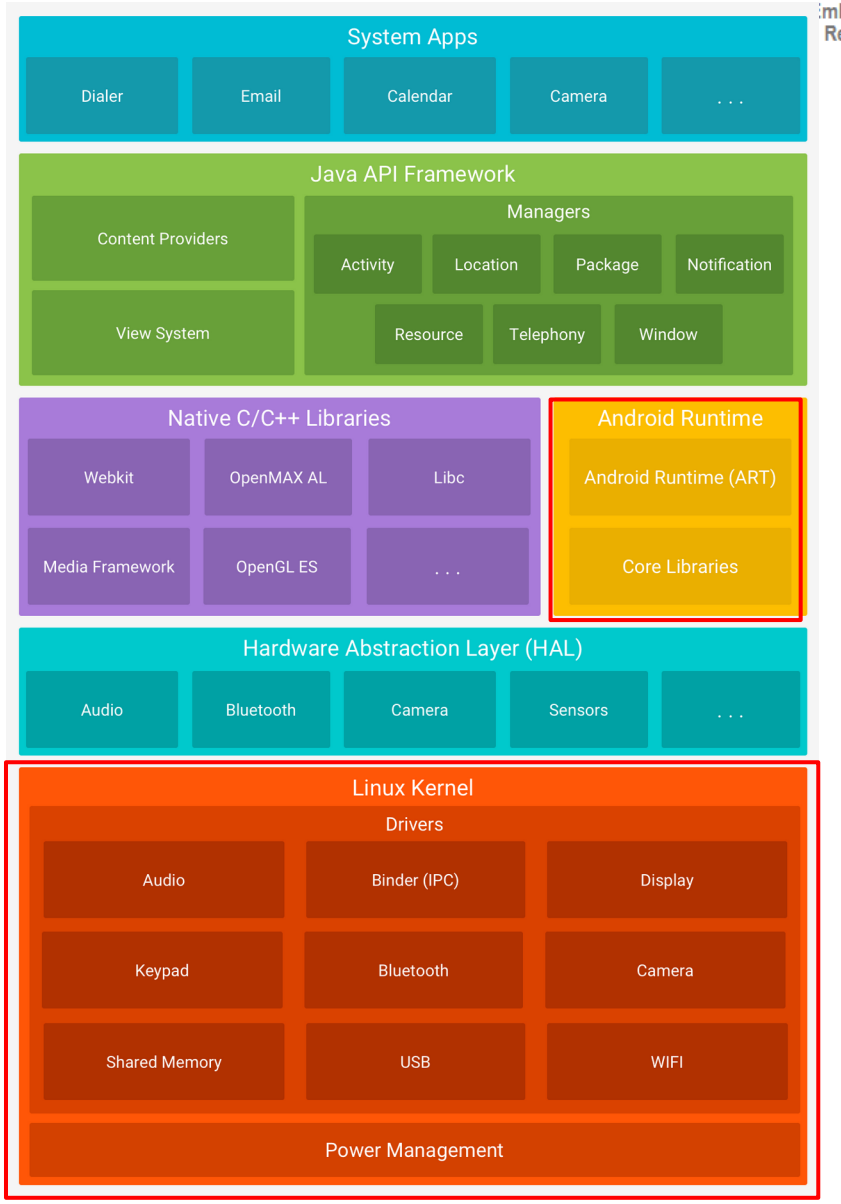
Organização interna do computador
Virtualização

Embedded Systems Research Group

- Máquina virtual
  - Corre por cima do hipervisor
    - Software based computer
- Process virtual machine
  - Ou plataforma de execução
    - Visa fornecer um ambiente de programação independente de plataforma
      - Abstraindo os detalhes do S.O. e hardware
    - E.g. JVM, PVM, ART (Android Runtime)
      - O ART ao contrário do JVM permite a compilação completa da aplicação após instalada

Organização interna do computador
Virtualização

- A pilha de software do Android



Organização interna do computador
Virtualização

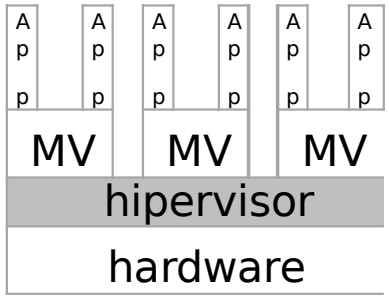
Embedded Systems Research Group

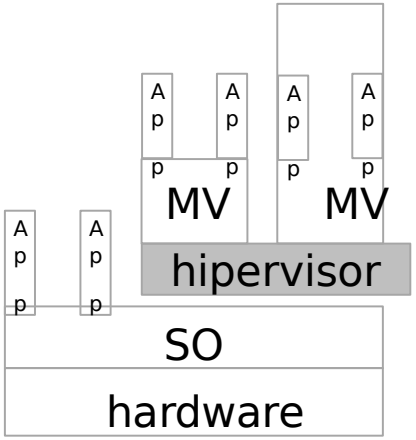
- Contentor
  - Baseados na tecnologia introduzida pelo Linux
    - Namespaces; cgroups
  - Agrupa uma aplicação e todas as suas dependências
    - E.g. Docker, Cloud Foundry (container runtimes)

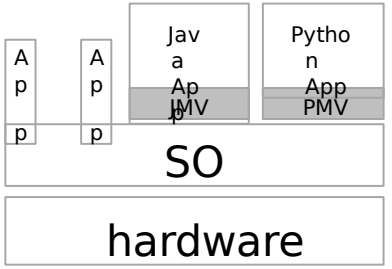
Organização interna do computador
Virtualização

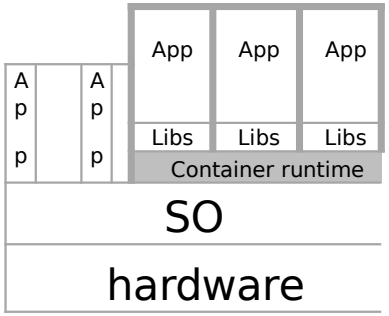
Embedded Systems Research Group

- Virtualização





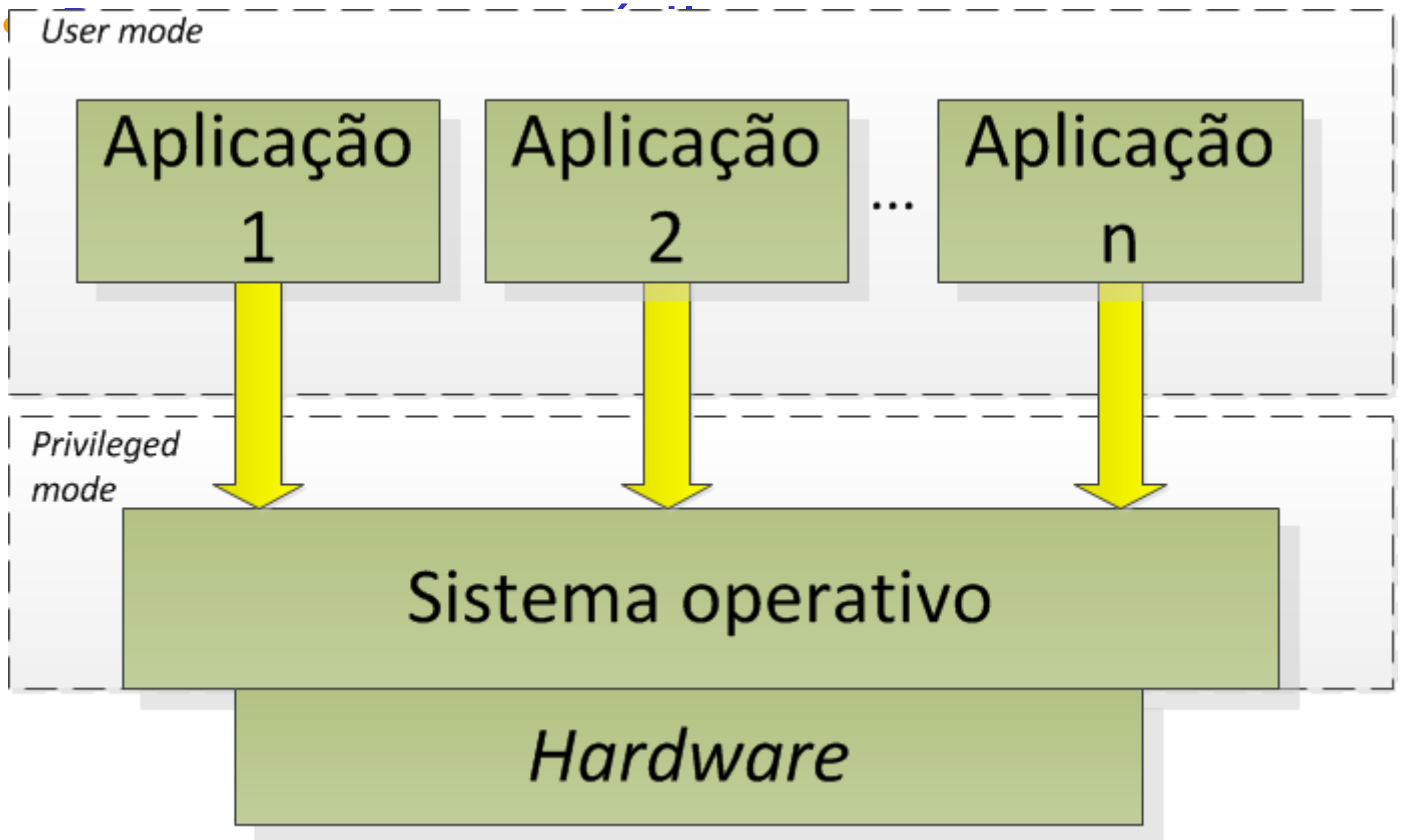




Organização interna do computador
Sistema computacional

Embedded Systems Research Group

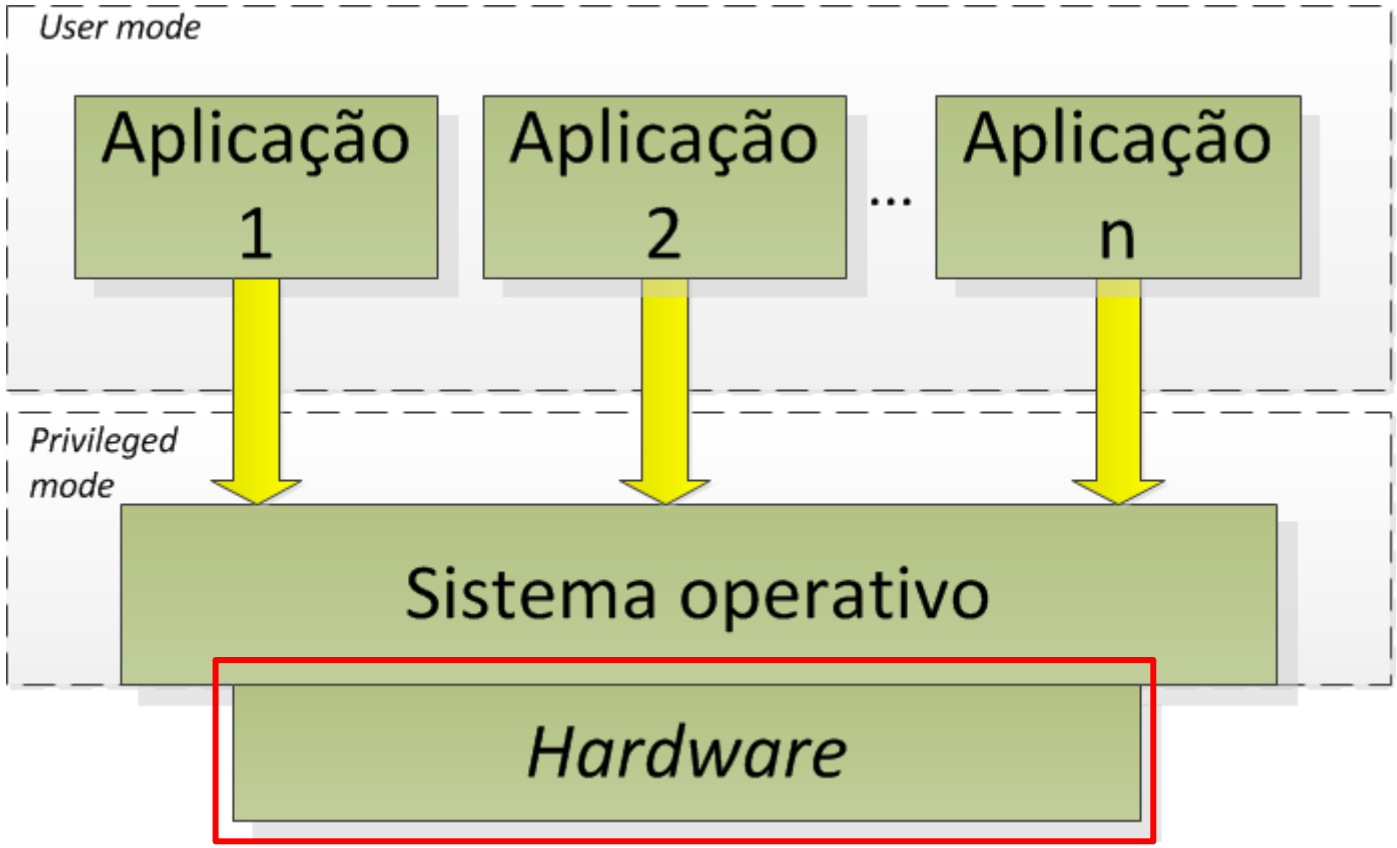
- Deixando a virtualização
  - Sistema computacional de uso genérico



Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Sistema computacional



Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Processador
  - Tem pelo menos 2 modos de operação
    - User mode
      - Modo não privilegiado de execução
        - Limita o acesso a determinadas regiões de memória e recursos
        - Impede o acesso a I/O
        - Algumas instruções não podem ser executadas
    - Privileged mode
      - Modo privilegiado de execução
        - Acede sem restrições a todos os recursos

NOTA: Isto não tem nada a ver com privilégios de administrador na gestão do Windows/Linux

Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Processador
  - Aplicações
    - Executam em user mode
      - Para evitar que as aplicações possam comprometer o sistema
  - Sistema operativo (kernel)
    - Executa em privileged mode
      - Para poder gerir e multiplexar os recursos entre as aplicações
    - Definição de sistema operativo
      - Controla os recursos de hardware e a sua multiplexagem pelas aplicações
      - Fornece um interface às aplicações para utilização desses recursos

Organização interna do computador

Sistema computacional

Embedded Systems Research Group

- Processador
  - Tem pelo menos 2 modos de operação
    - Quer isto dizer que se escrevêssemos um programa em assembly
      - Não podíamos usar todas as instruções aceites pelos processador
        - O nosso programa não corre em modo privilegiado

Organização interna do computador
Sistema computacional

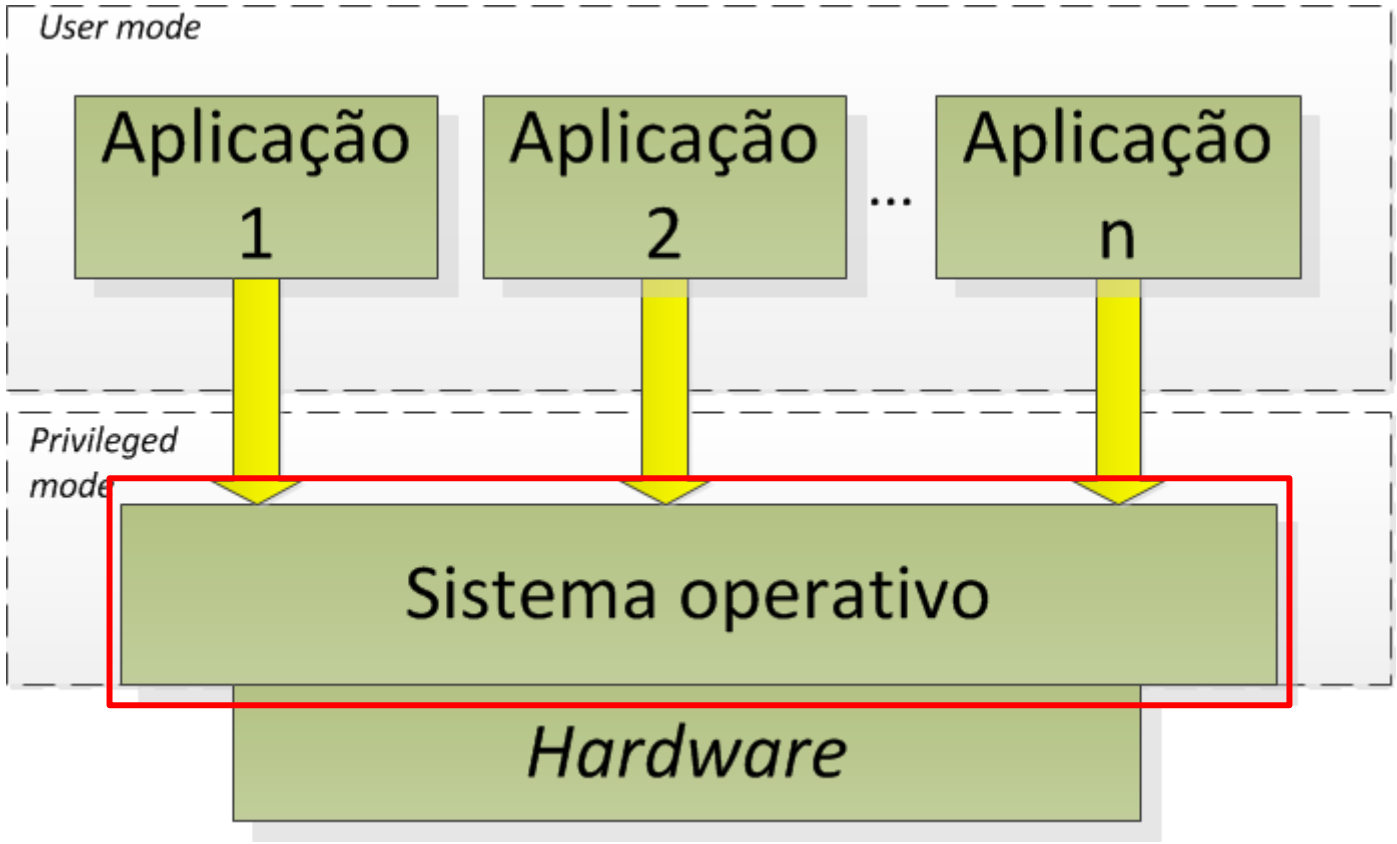
Embedded Systems Research Group

- Processador
  - MMU – Memory management unit
    - Conceito relacionado mas independente
      - Implementa o conceito de memória virtual
        - Os endereços de memória não correspondem a endereços físicos
          - É necessário um “conversor” de endereços, a MMU
        - Um programa vê o espaço de memória como se fosse todo seu
    - O conceito de memória virtual impede que um programa acede à memória do outro
      - Cada programa ”vive” no seu “contentor” de memória

Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Sistema computacional



Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Sistemas operativo
  - Os dois mecanismos apresentados são a base de qualquer sistema operativo moderno
    - Garantindo robustez ao sistema para que uma aplicação mal comportada não comprometa todo o sistema
      - Um programa “normal” deixa de poder “mexer” directamente com os recursos de hardware
  - Definição
    - Controla os recursos de hardware e a sua multiplexagem pelas aplicações
    - Fornece um interface às aplicações para utilização desses recursos

Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Sistema operativo
  - Então como fazem as aplicações para aceder aos recursos (disco, memória, teclado, vídeo, etc.)?
    - Invocam chamadas ao sistema operativo
      - System calls
        - Mas os meus programas usam tudo isso e nunca invoquei uma system call…
          - As funções da biblioteca do C fazem-no por nós
          - Ex: `printf()`, `fwrite()`, `malloc()`, etc.
          - Nem todas as funções da biblioteca do C invocam system calls
          - Ex: `sqrt()`, `pow()`, `strcmp()`, `strtok()`, etc.
        - Que “acedem” aos device drivers se necessário

Organização interna do computador
Sistema computacional

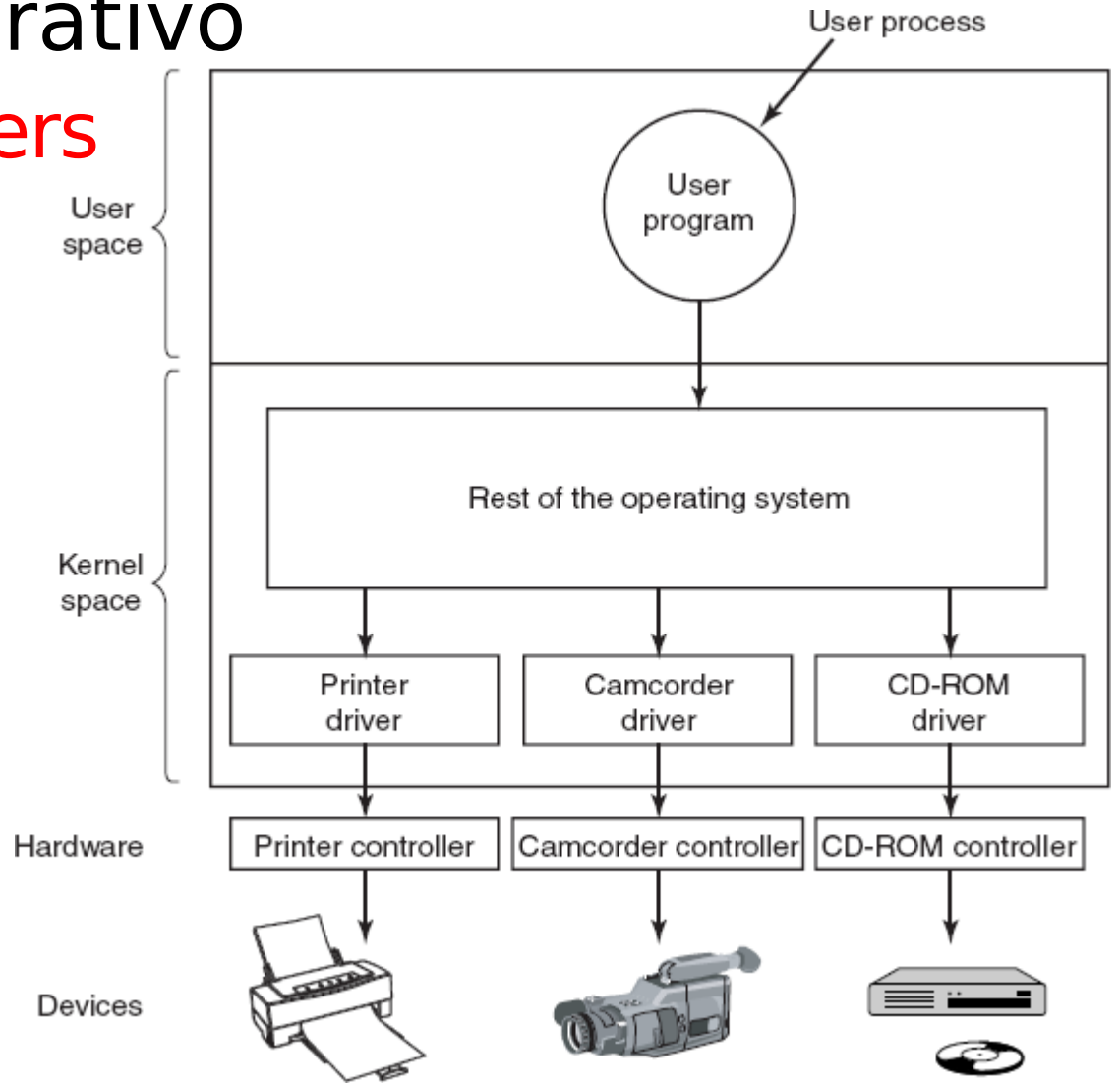
Embedded Systems Research Group

- Sistema operativo
  - Por outro lado, todos os dias novos dispositivos surgem no mercado
    - Para se ligarem ao computador
      - E.g. impressoras
    - O S.O. não está preparado para comunicar estes dispositivos
      - Isto é feito através de device drivers

Organização interna do computador
Sistema computacional

Embedded Systems Research Group

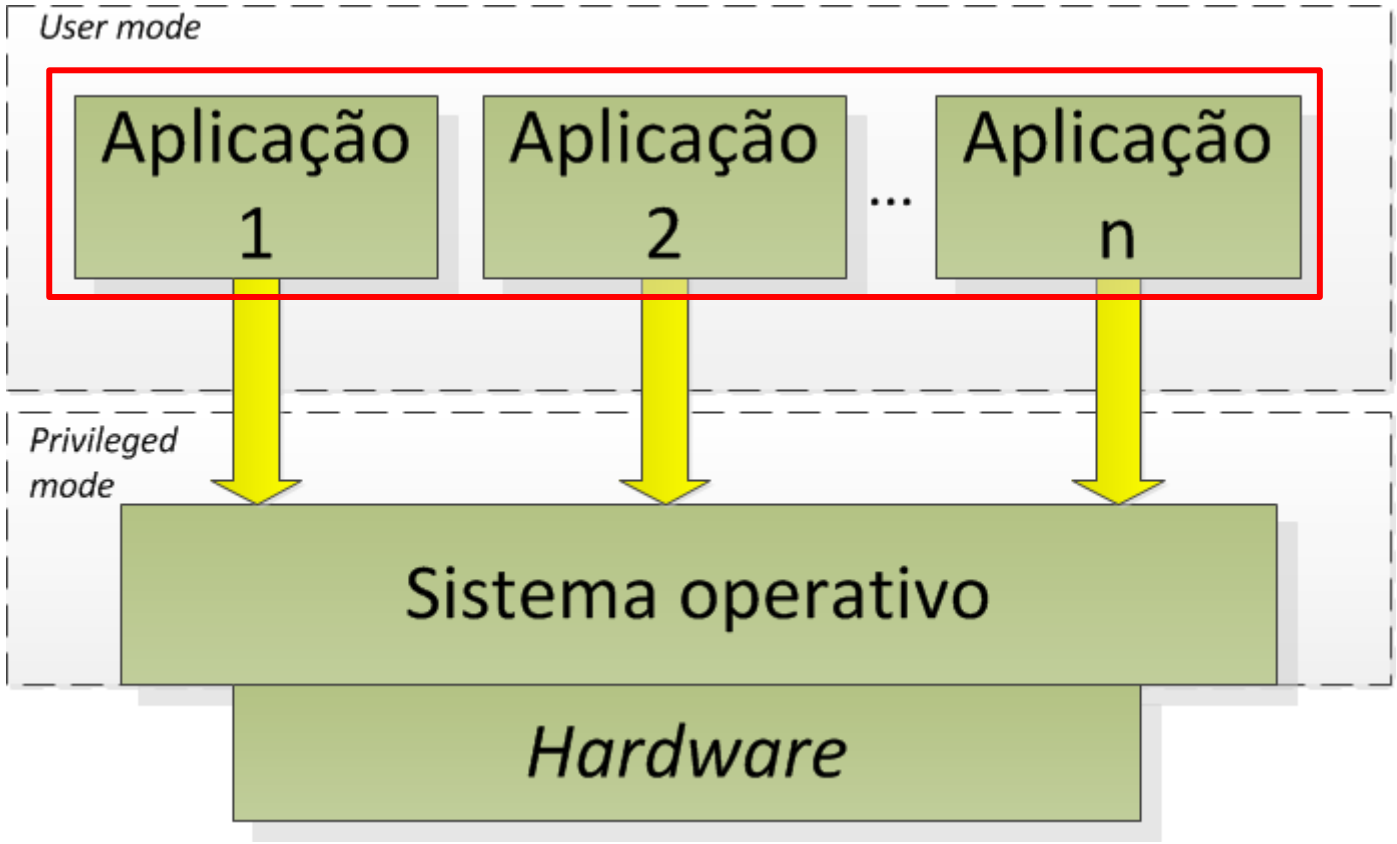
- Sistema operativo
  - Device drivers



Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Sistema computacional



Organização interna do computador
Sistema computacional microcontrolador

Embedded Systems Research Group

- Processos
  - Aplicações em execução
    - Usam diretamente o processador
      - Para executar as instruções do programa
        - Processarem os dados
        - Interagindo com a memória do processo
    - Interagem com o S.O. através de system calls
      - Sempre que precisam de mais memória
      - Aceder a recursos externos
        - Disco, teclado vídeo, rede, porta série, etc...
      - Comunicação entre processos
      - ...

Organização interna do computador
Sistema computacional

Embedded Systems Research Group

- Processos
  - Já vimos que os S.O. modernos colocam os processos em "contentores"
    - Estão confinados ao seu próprio espaço de endereçamento
      - De forma a não comprometerem o sistema, nem outros processos
  - Mas... e quando os processos querem comunicar entre si?
    - Terá de ser o S.O. a ajudar

Organização interna do computador
Sistema computacional

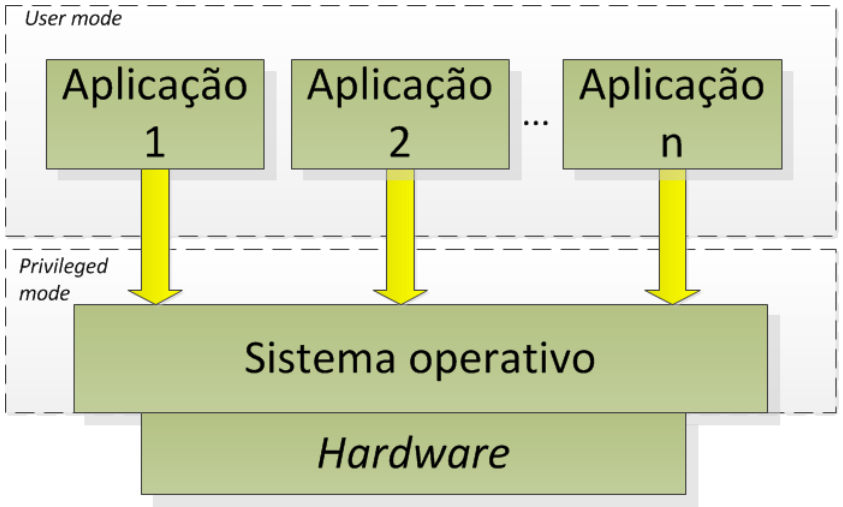
Embedded Systems Research Group

- Processos
  - Comunicação entre processos
    (Inter-process communication – IPC)
    - Razões para processo querer comunicar com outro
      - Partilha de dados
      - Modularidade
      - Desempenho
      - Privilégios
      - Conveniência

Organização interna do computador
Sistema computacional microcontrolador

Embedded Systems Research Group

- Sistema computacional
  - Desktop vs microcontrolador



Organização interna do computador
Sistema computacional microcontrolador

Embedded Systems Research Group

- Sistema computacional microcontrolador
  - Não tem um sistema operativo a gerir o hardware como o Windows/Linux
    - O micro não é complexo
    - Tipicamente corre só uma aplicação/tarefa
      - Quando há mais do que uma tarefa há necessidade de um “library OS”
        - Um SO que não é mais do que um conjunto de rotinas compiladas com o nosso programa
  - A aplicação corre diretamente no hardware
    - Quando termina não pode voltar para o S.O.
    - Tem acesso a todo o hardware e suas particularidades

Organização interna do computador
Sistema computacional microcontrolador

Embedded Systems Research Group

- A aplicação corre diretamente no hardware
  - Quando termina não pode voltar para o S.O.
    - Assim o programa que fizermos, nunca pode terminar
      - Visto o micro estar sempre no ciclo fetch>decode>execute
        - A processar instruções
        - Que instruções vai processar se o nosso programa terminar?
          - Comportamento indeterminado...
      - O programa deve “terminar” com um ciclo infinito
        - `while (1)`
        - `for(;;)`

Organização interna do computador
Sistema computacional microcontrolador

Embedded Systems Research Group

- Anatomia de um programa em C

```
//declaracoes include

//Variaveis globais

//Funcoes auxiliares

 void main (void) {
    //Declaracao de
    variaveis

    //inicializacoes

    while (1) {
        //codigo do
        programa

    }
}
```

Organização interna do computador

Sistema computacional microcontrolador

```
#include "STM32L1xx.h"      /* I/O port/register names/addresses for the STM32L1xx microcontrollers */

/* Global variables – accessible by all functions */
int count, bob;             //global (static) variables – placed in RAM

/* Function definitions*/
int function1(char x) {     //parameter x passed to the function, function returns an integer value
  int i,j;                  //local (automatic) variables – allocated to stack or registers
  -- instructions to implement the function
}

/* Main program */
void main(void) {
  unsigned char sw1;        //local (automatic) variable (stack or registers)
  int k;                    //local (automatic) variable (stack or registers)
/* Initialization section */
  -- instructions to initialize  variables, I/O ports, devices, function registers
/* Endless loop */
  while (1) {               //Can also use:  for(;;) {
   -- instructions to be repeated
  } /* repeat forever */
}
```

Declare local variables

Initialize variables/devices

Body of the program

https://www.eng.auburn.edu/~nelsovp/courses/elec3040_3050/C%20programming%20for%20embedded%20system%20applications.pdf

Organização interna do computador
Sistema computacional microcontrolador

Embedded Systems Research Group

- A aplicação corre diretamente no hardware
  - Tem acesso a todo o hardware e suas particularidades
    - Para suporte aos elementos da arquitetura do micro
      - Existem extensões à linguagem C
        - Áreas de memória
        - Tipos de dados
        - Apontadores
        - Atributos de funções
  - Mas isto fica para depois...

Organização interna do computador

Obrigado