

Programação Reativa para Educação

Francisco Sant'Anna

`francisco@ime.uerj.br`

`http://github.com/fsantanna-no/101`

Computador???



Dispositivos de Entrada e Saída (I/O)

PERIFÉRICOS DO COMPUTADOR

1 - DE ENTRADA (INPUT):



TECLADO



Mouse



Joystick



Webcam



microfone



Scanner



Mesa Digitalizadora

2 – DE SAÍDA (OUTPUT):

MONITOR

CRT



Cristal Líquido



LCD



Projeter Multimídia



Impressora



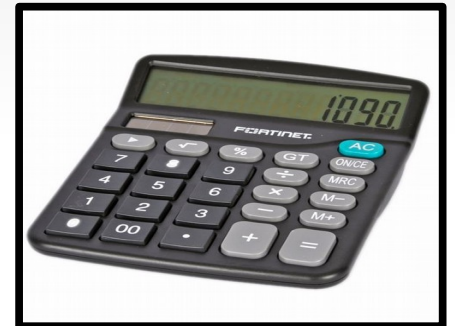
Caixas de som



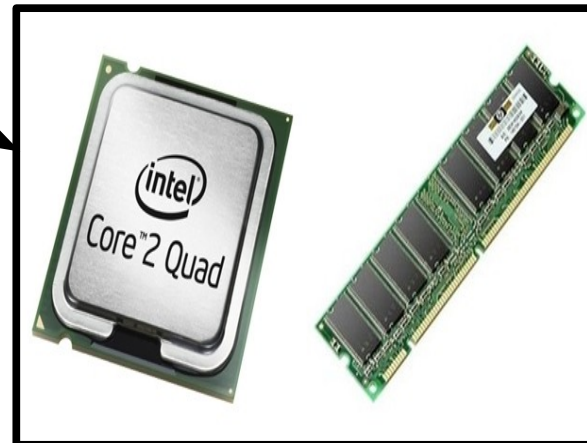
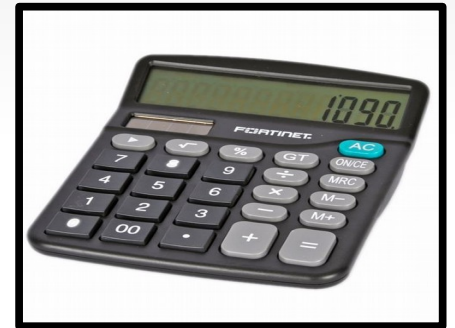
Dispositivos de Entrada e Saída (I/O)



Computador???



Computador???



Computador é Programável

Hardware + Software

Hardware vs Software

The main difference between hardware and software are as follows:

Hardware

1. Physical parts of the computer are called hardware.
2. You can touch, see and feel hardware.
3. Hardware is constructed using physical materials or components.
4. Computer is hardware, which operates under the control of a software.
5. If hardware is damaged, it is replaced with new one.
6. Hardware is not affected by computer viruses.
7. Hardware cannot be transferred from one place to another electronically through network.
8. User cannot make new duplicate copies of the hardware.



Software

1. A set of instructions given to the computer is called software.
2. You cannot touch and feel software.
3. Software is developed by writing instructions in programming language.
4. The operations of computer are controlled through software.
5. If software is damaged or corrupted, its backup copy can be reinstalled.
6. Software is affected by computer viruses.
7. Software can be transferred from one place to another electronically through network.
8. User can make many new duplicate copies of the software.



Hardware vs Software

The main difference between hardware and software are as follows:

Hardware

1. Physical parts of the computer are called hardware.
2. You can touch, see and feel hardware.
3. Hardware is constructed using physical materials or components.



Software é o que você xinga, hardware o que você chuta

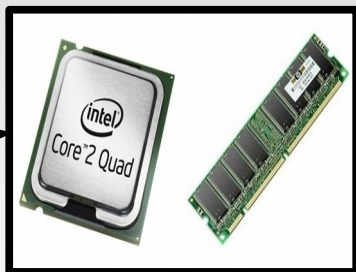
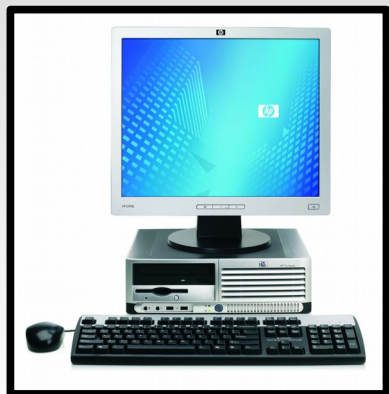
- 4.
- 5.
- 6.
7. Hardware cannot be transferred from one place to another electronically through network.
8. User cannot make new duplicate copies of the hardware.

Software

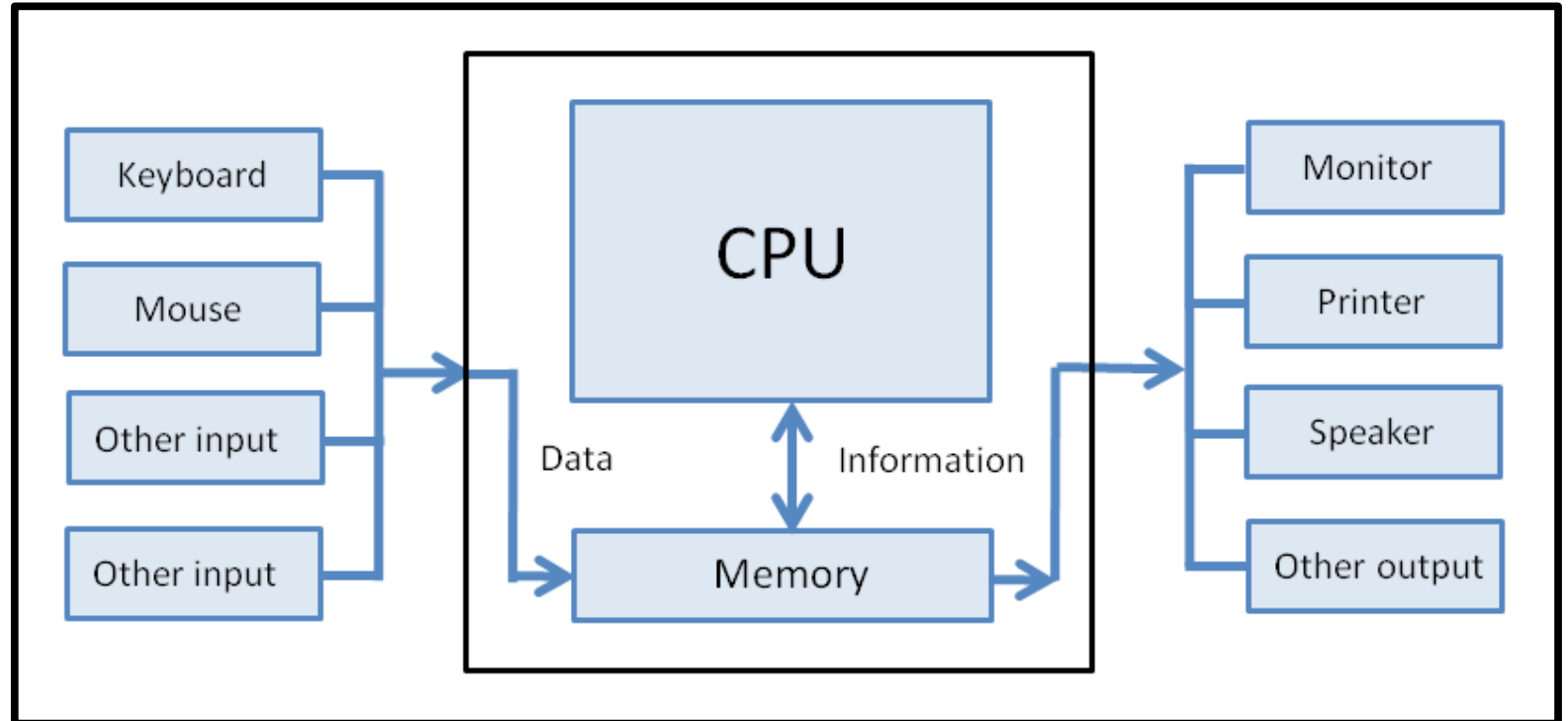
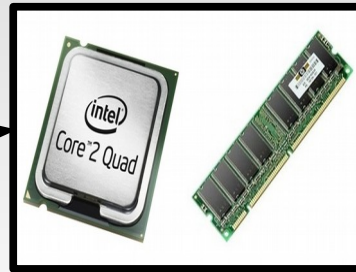
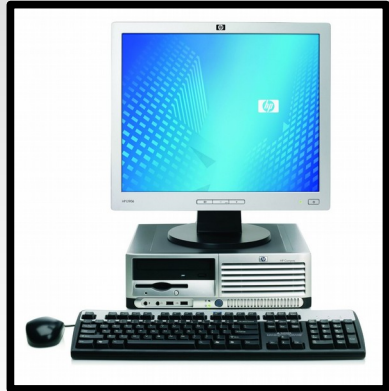
1. A set of instructions given to the computer is called software.
2. You cannot touch and feel software.
3. Software is developed by writing instructions in programming language.
4. The operations of computer are controlled through software.
5. If software is damaged or corrupted, its backup copy can be reinstalled.
6. Software is affected by computer viruses.
7. Software can be transferred from one place to another electronically through network.
8. User can make many new duplicate copies of the software.



Computador!



Computador!



Entrada e Saída

```
var integer x ← 25;
var integer y ← 25;
emit GRAPHICS_DRAW_PIXEL(x,y);

loop do
    var integer key ← await KEY_PRESS;

    if key = KEY_LEFT then
        x ← x - 1;
    else/if key = KEY_RIGHT then
        x ← x + 1;
    else/if key = KEY_UP then
        y ← y - 1;
    else/if key = KEY_DOWN then
        y ← y + 1;
    end

    emit GRAPHICS_DRAW_PIXEL(x,y);
end
```


Exercício 2.1

- Não deixar “rastro” ao movimentar o pixel

Exercício 2.1

- Não deixar “rastro” ao movimentar o pixel
- <https://fsantanna.github.io/pico-ceu/out/manual/v0.30/graphics/>

Entrada e Saída

Entrada e Saída



Entrada e Saída



Classificação de “Sistemas”

(Berry 1989)

Classificação de “Sistemas”

(Berry 1989)

- Sistemas transformacionais
 - “*compute results from a given set of inputs*”

Classificação de “Sistemas”

(Berry 1989)

- Sistemas transformacionais
 - *“compute results from a given set of inputs”*
- Sistemas interativos
 - *“interact at their own speed with users or with other programs”*

Classificação de “Sistemas”

(Berry 1989)

- Sistemas transformacionais
 - *“compute results from a given set of inputs”*
- Sistemas interativos
 - *“interact at their own speed with users or with other programs”*
- Sistemas reativos
 - *“interact with their environment, but at a speed which is determined by the environment, not by the program itself”*

Classificação de “Sistemas”

(Berry 1989)

- Sistemas transformacionais
 - *“compute results from a given set of inputs”*
- Sistemas interativos
 - *“interact at their own speed with users or with other programs”*
- Sistemas reativos
 - *“interact with their environment, but at a speed which is determined by the environment, not by the program itself”*

Classificação de “Sistemas”

(Berry 1989)

- Sistemas transformacionais
 - *“compute results from a given set of inputs”*
- Sistemas interativos
 - *“interact at their own speed with users or with other programs”*
- Sistemas reativos
 - *“interact with their environment, but at a speed which is determined by the environment, not by the program itself”*
- gcc, gdb, quake, md5, shell, zip, http, gui

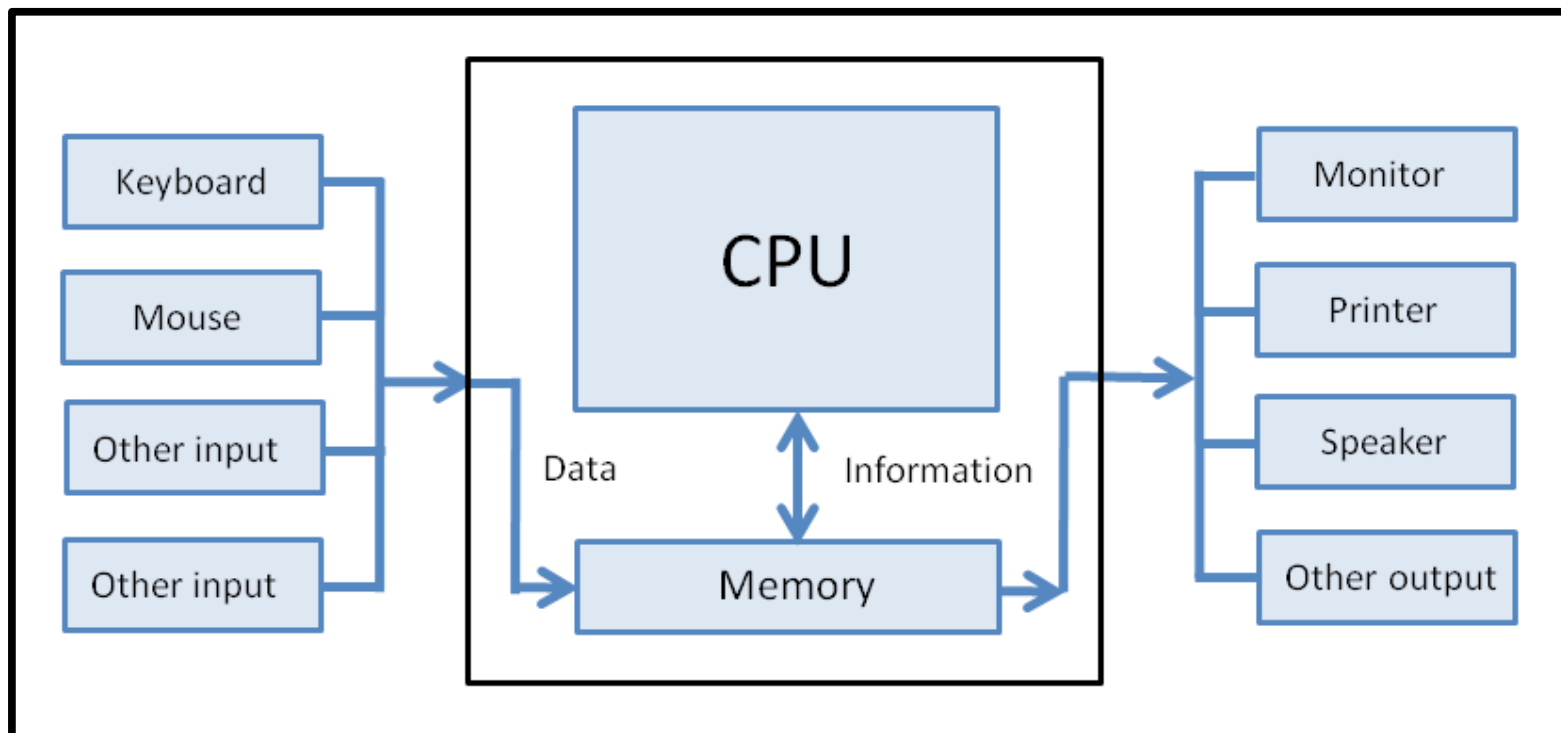
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



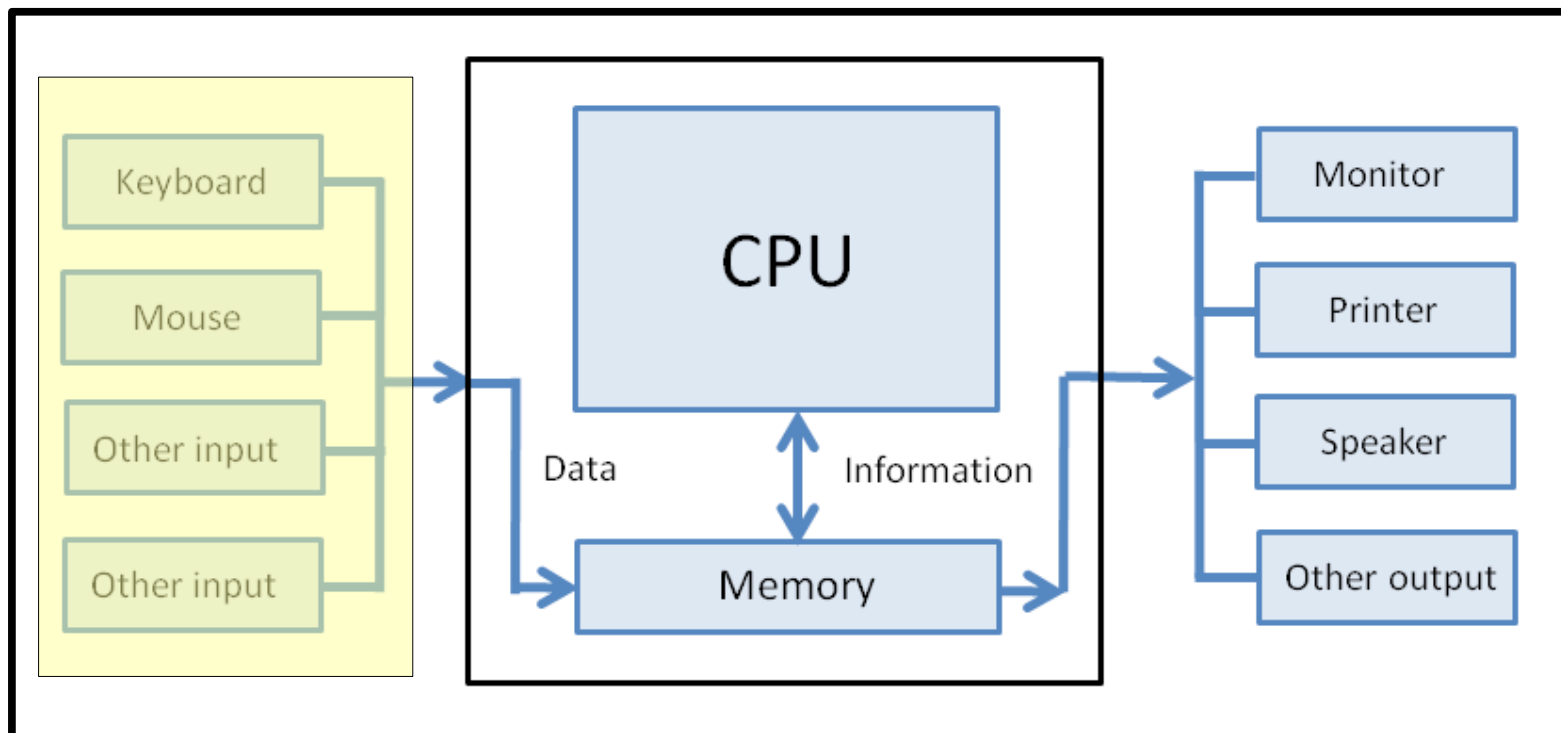
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



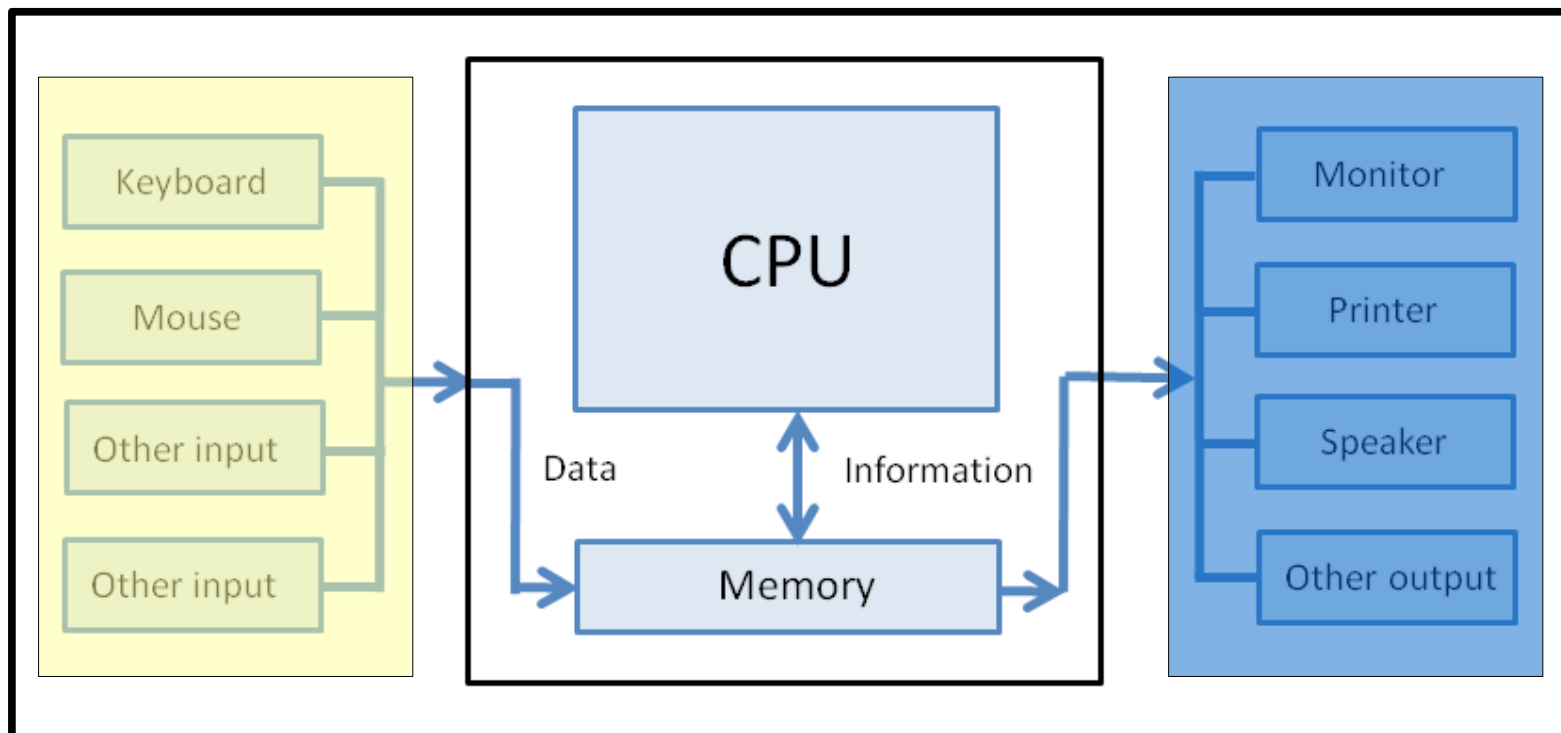
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



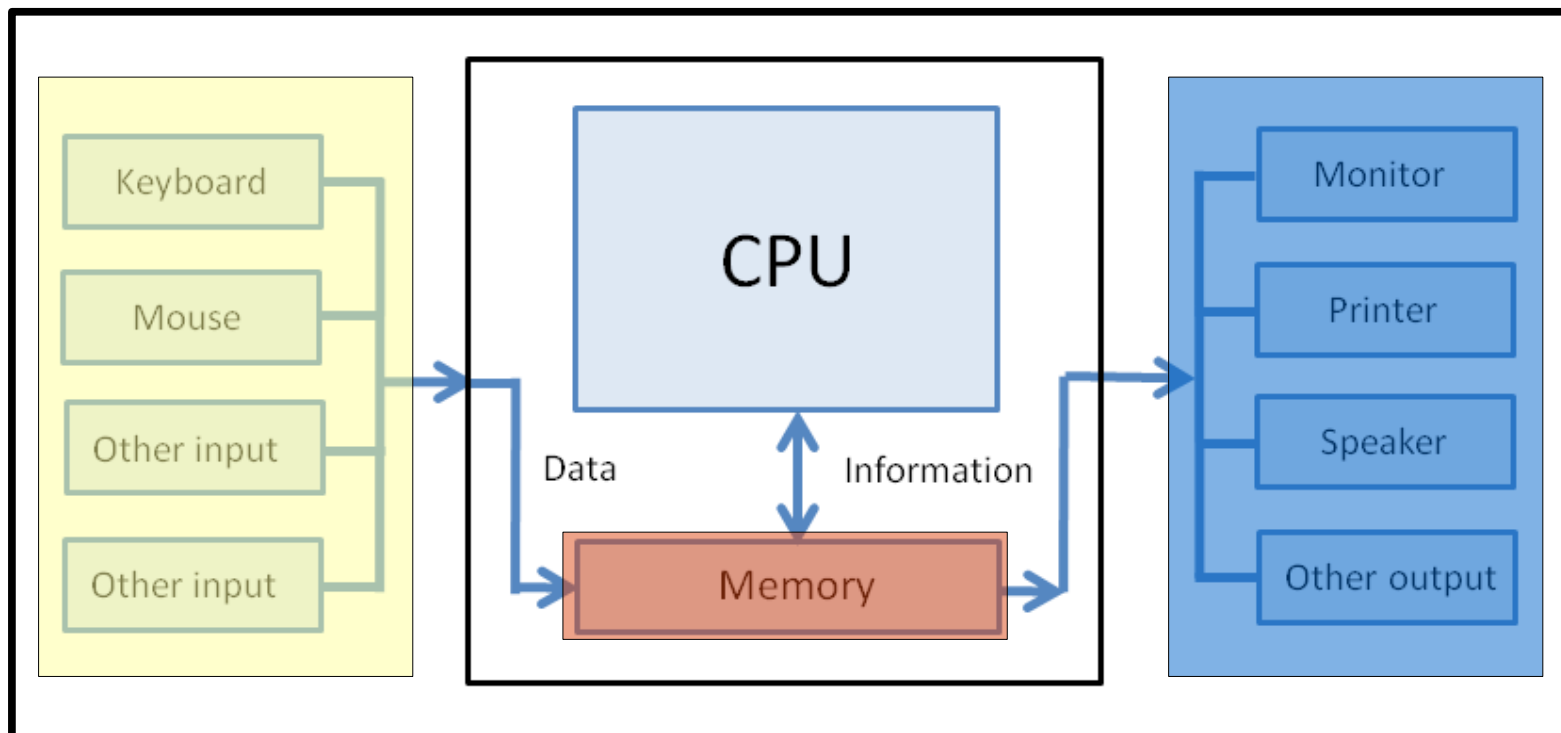
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



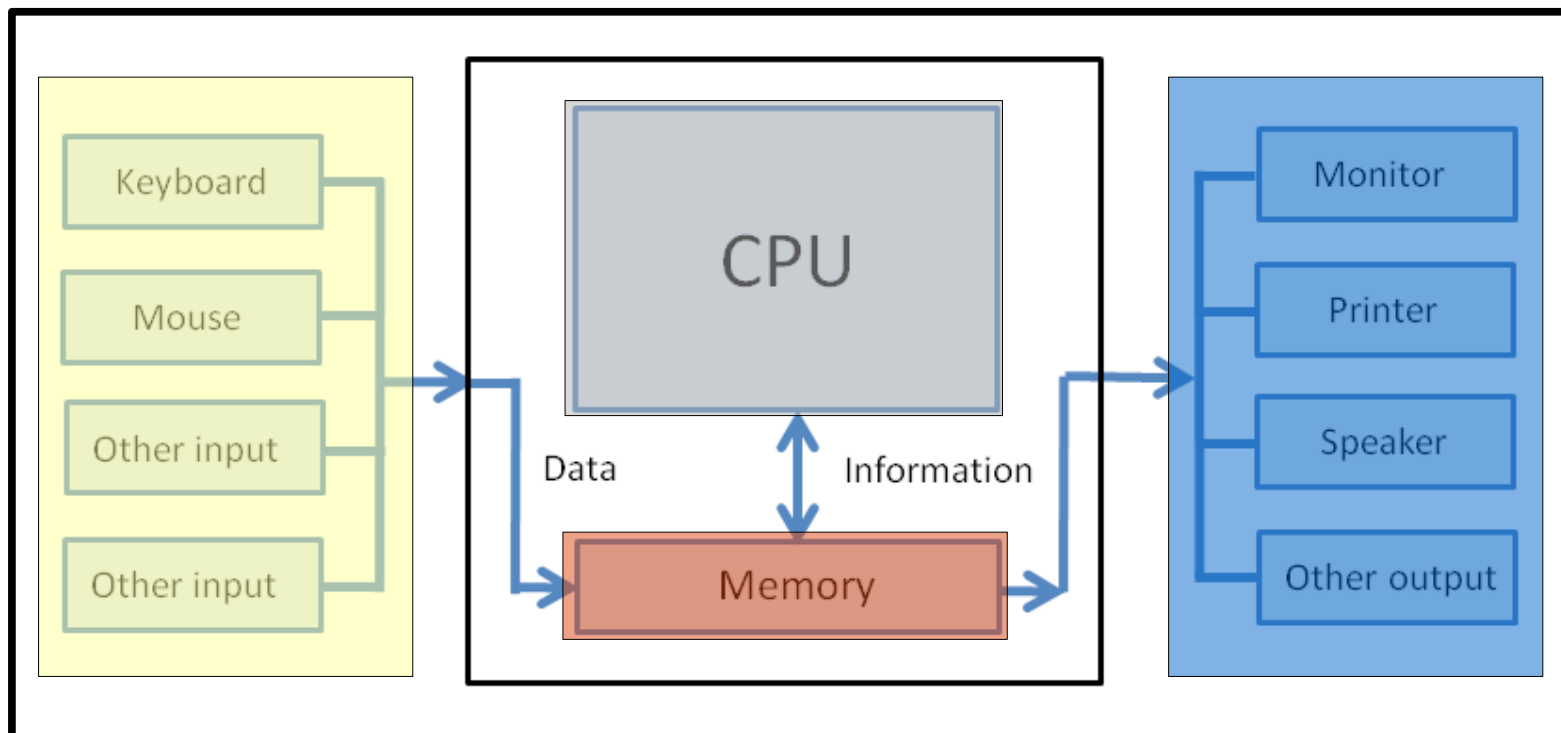
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



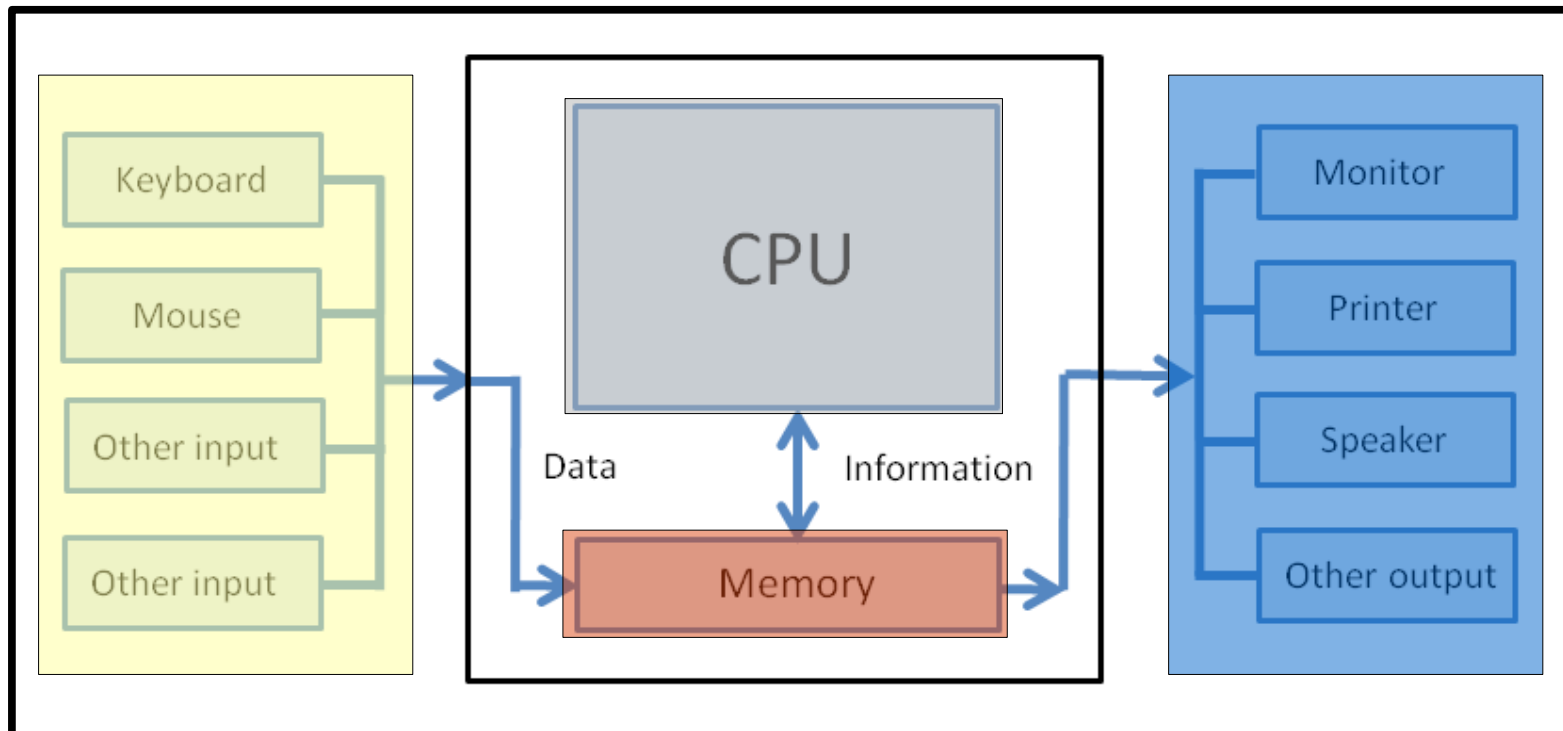
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



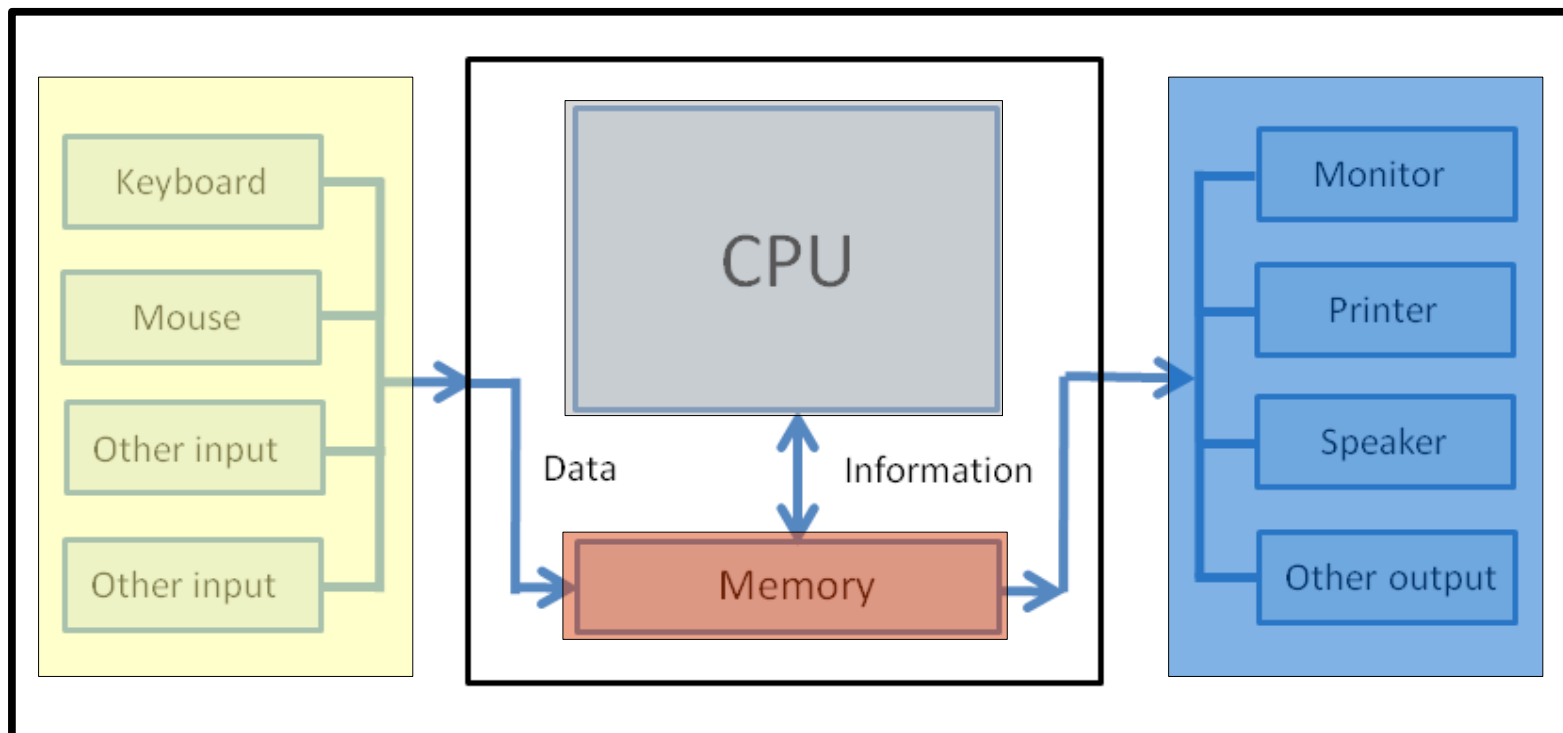
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



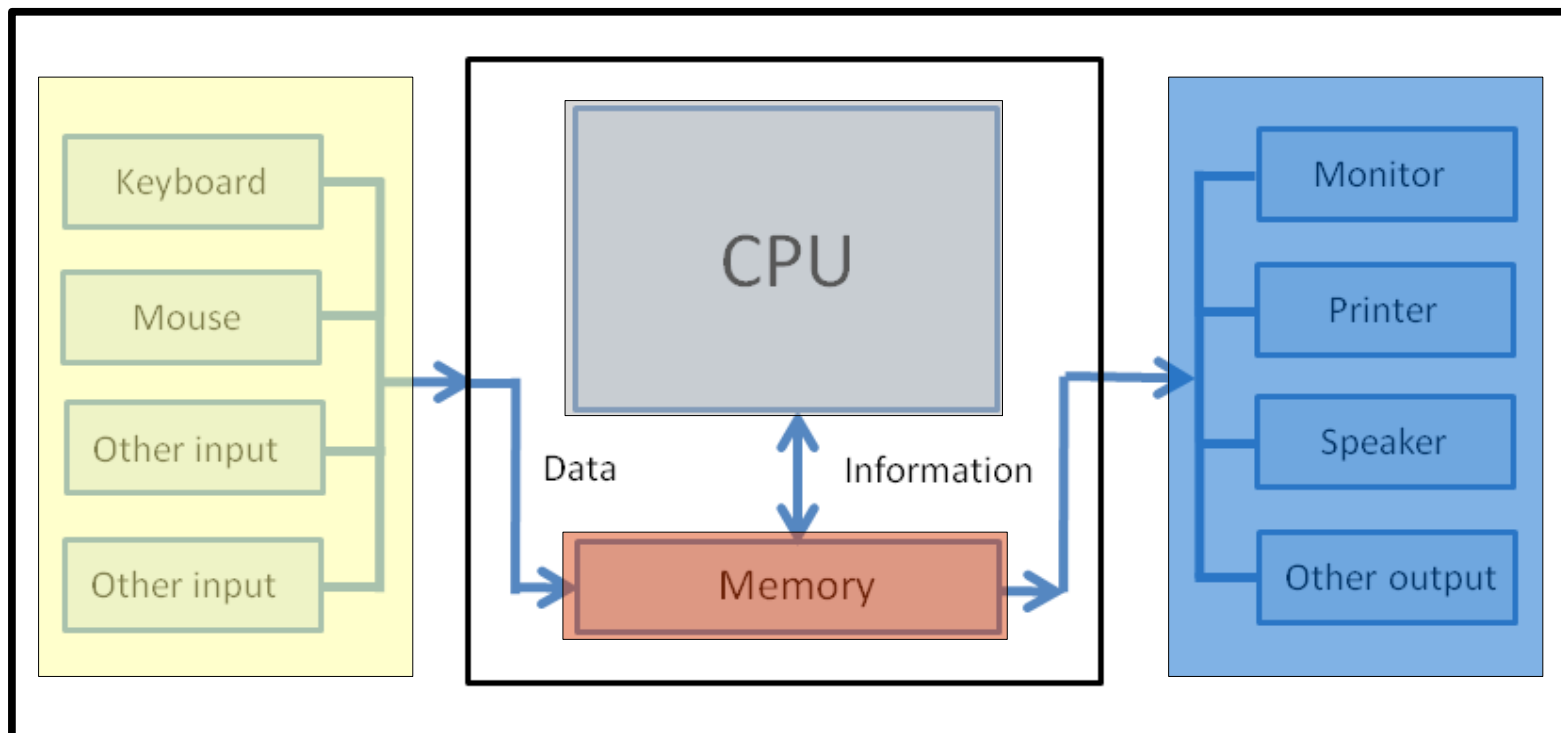
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



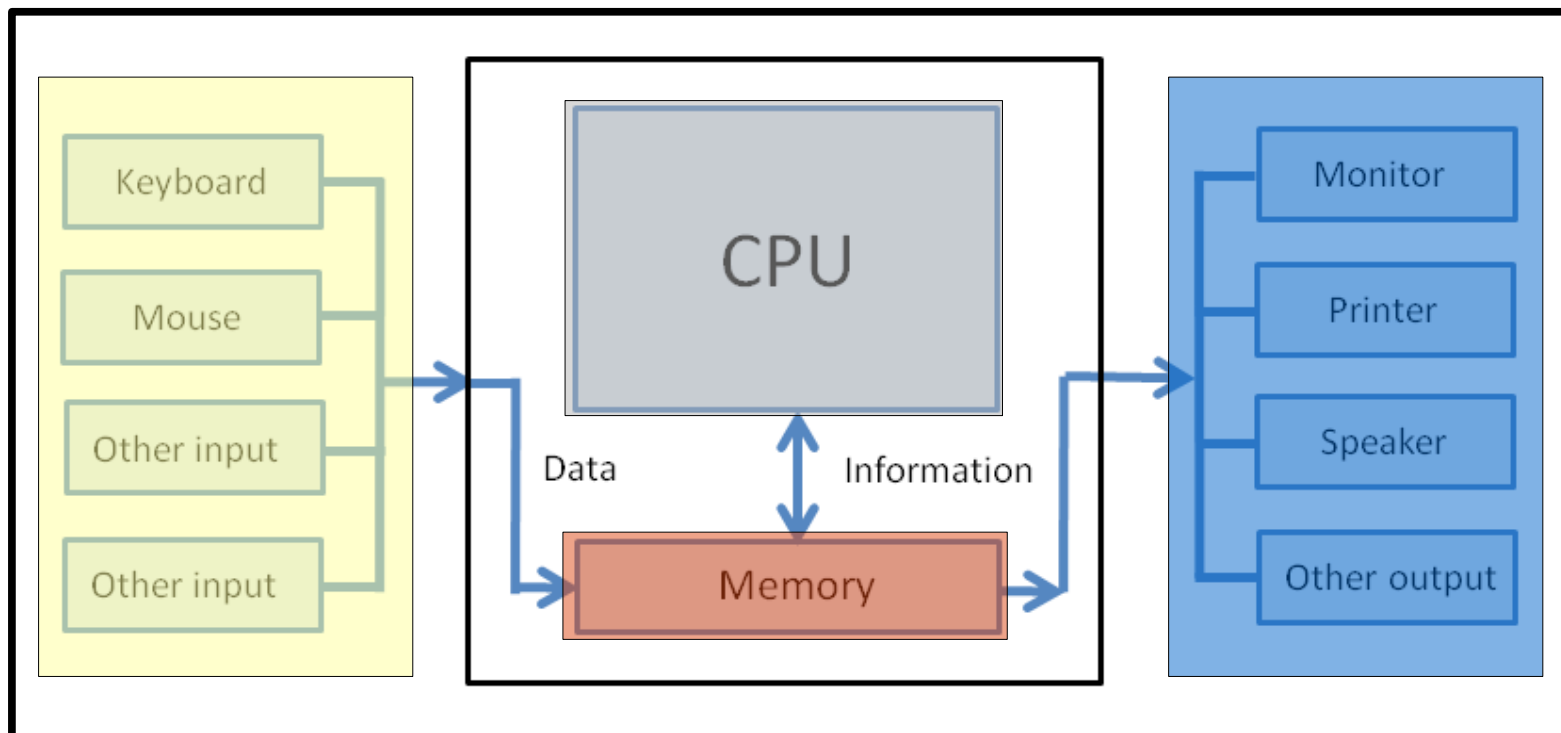
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow x - 1$ ;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



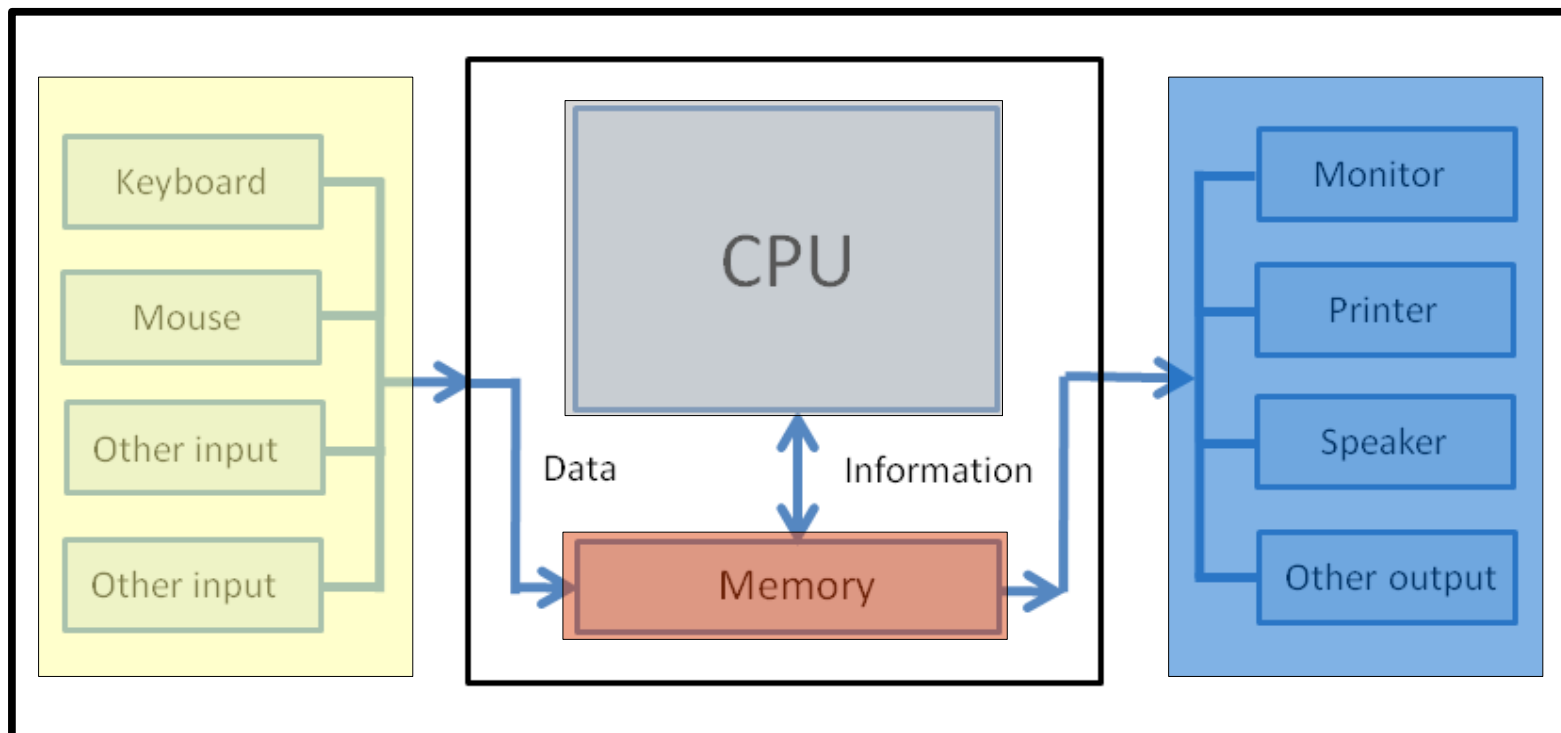
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow x - 1$ ;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



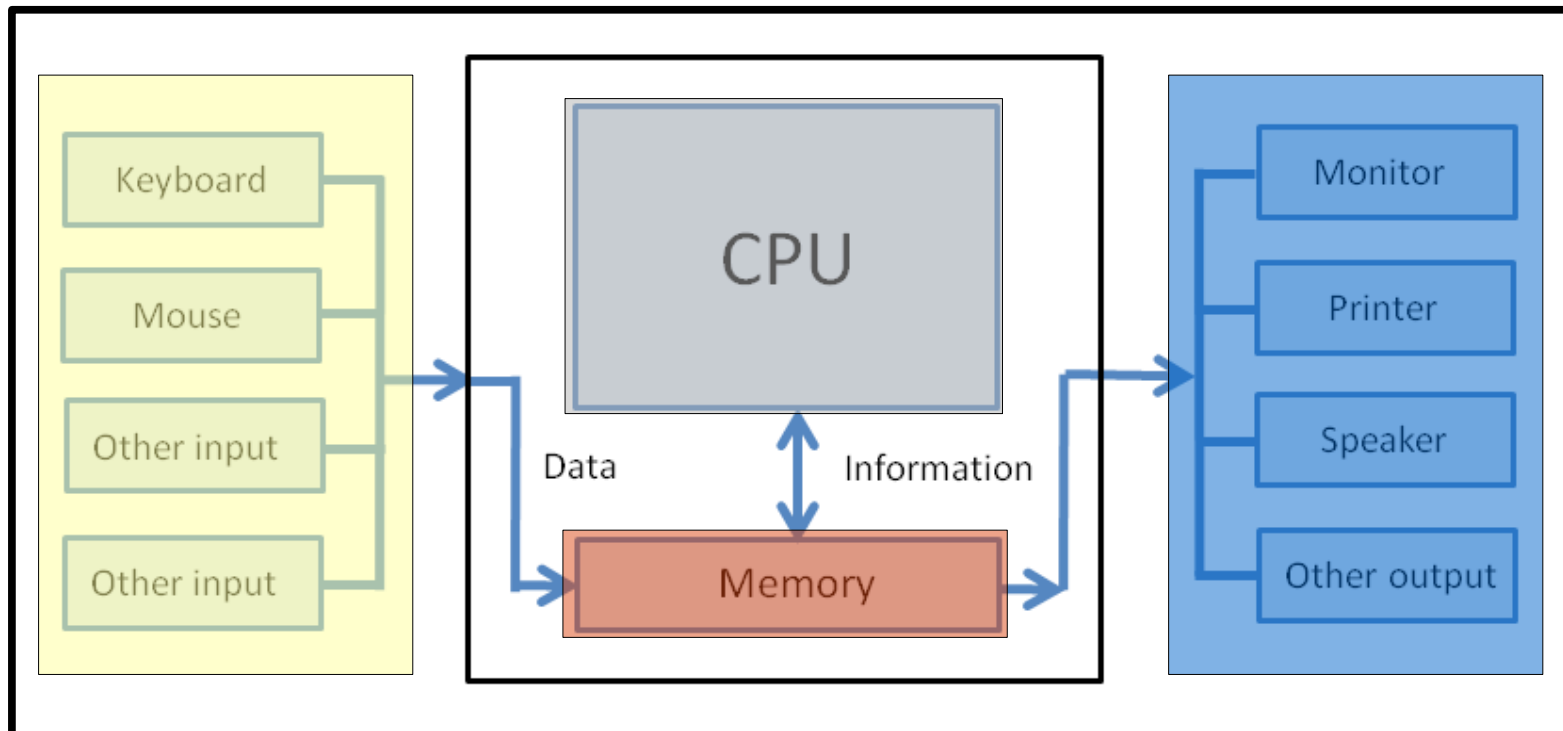
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow x - 1$ ;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



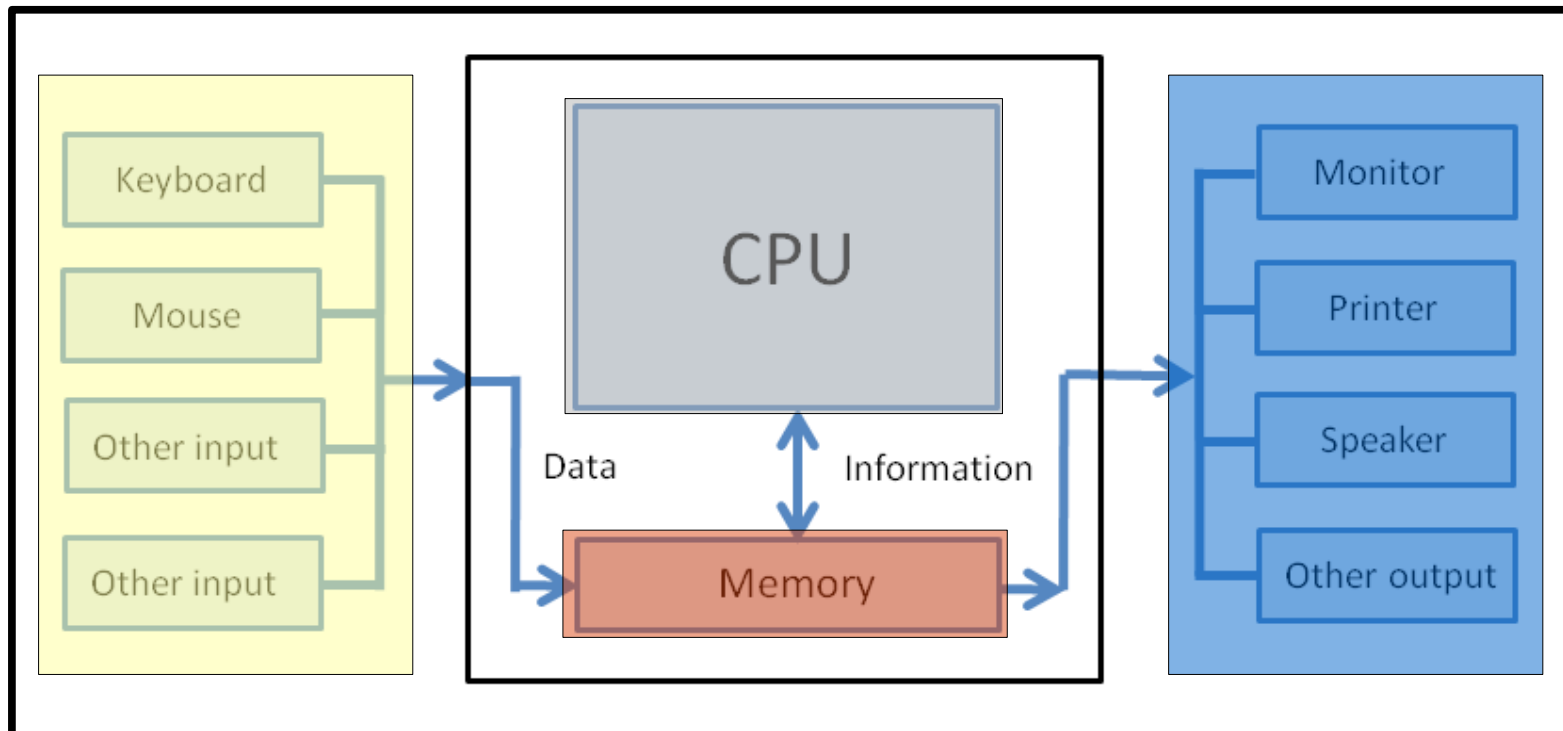
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow x - 1$ ;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



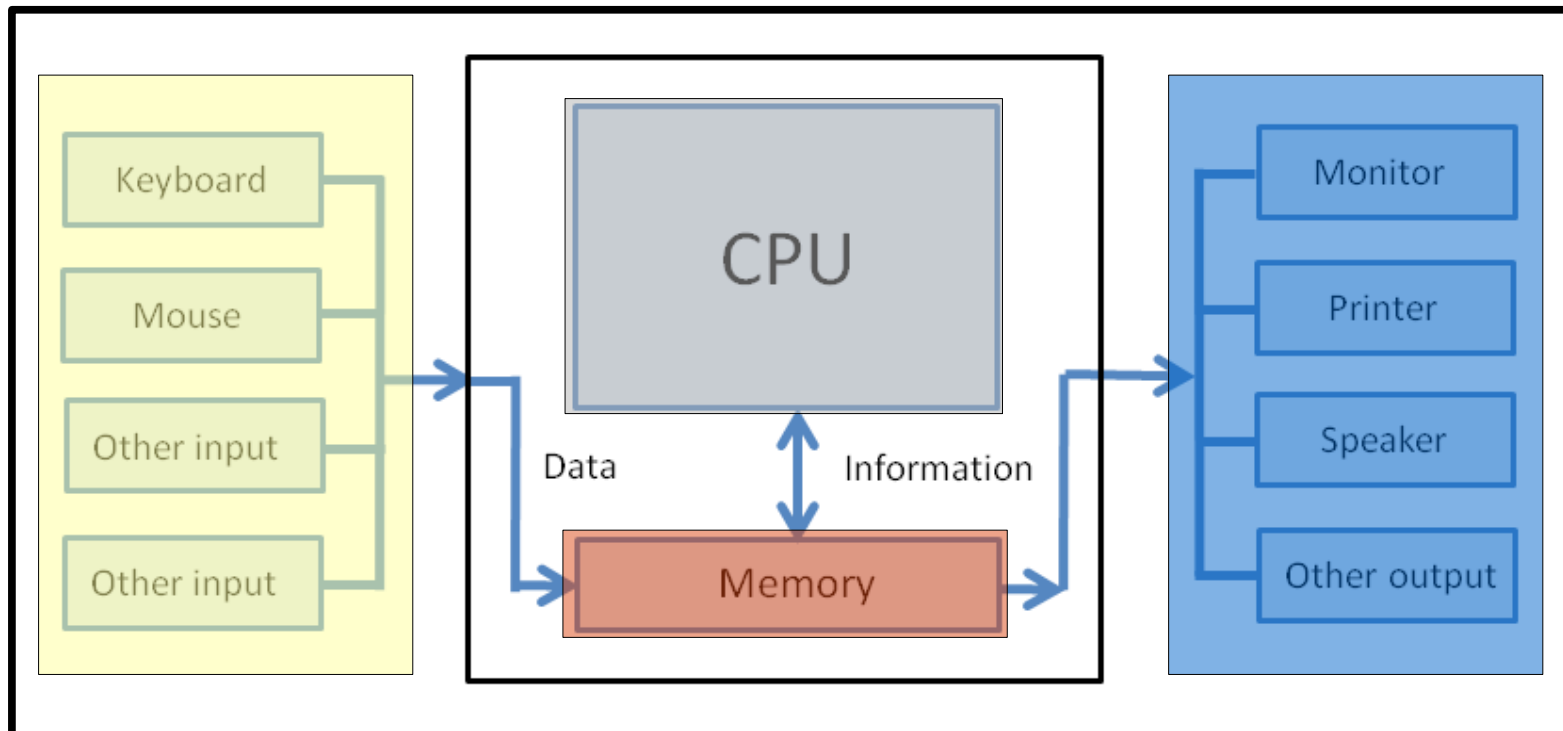
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow x - 1$ ;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



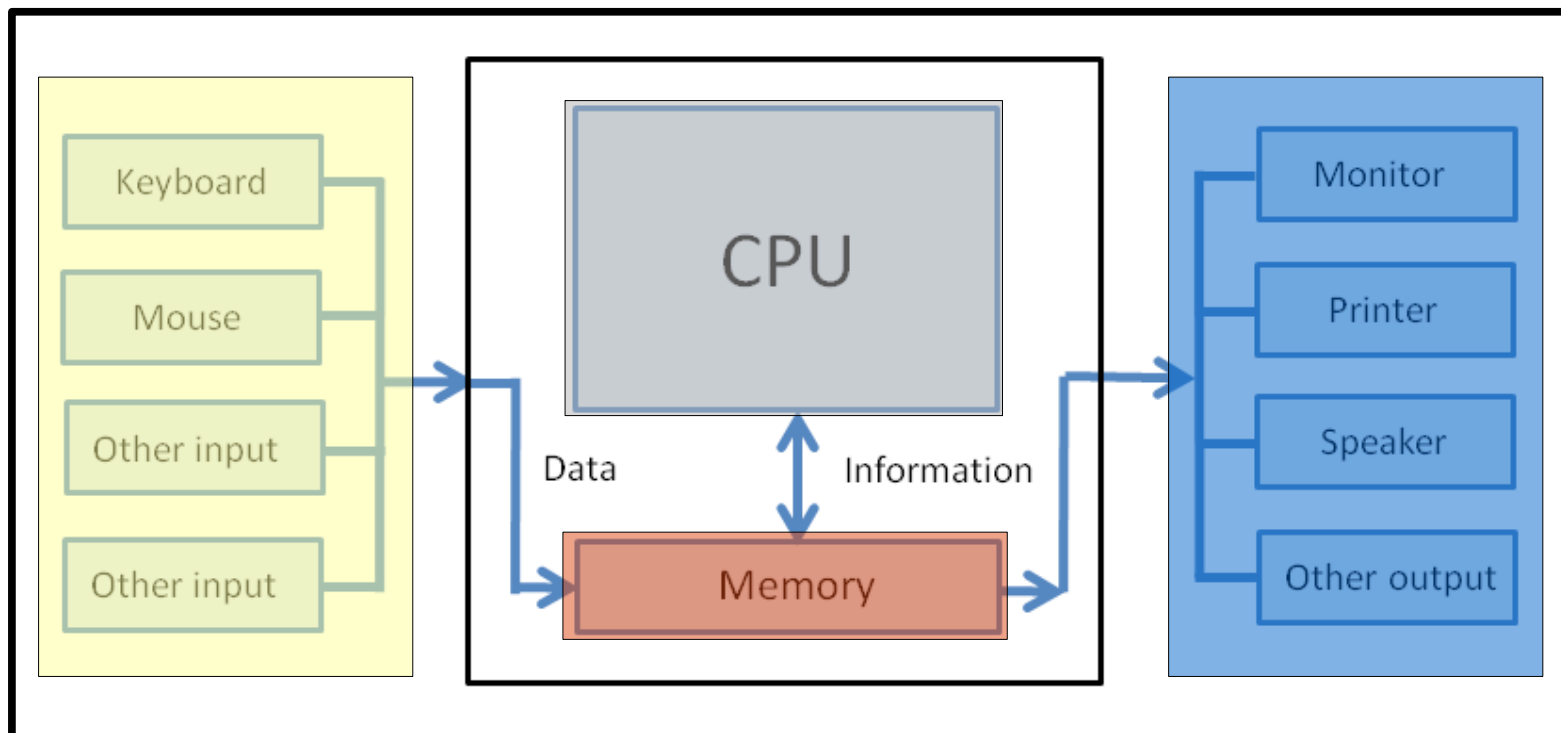
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow x - 1$ ;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



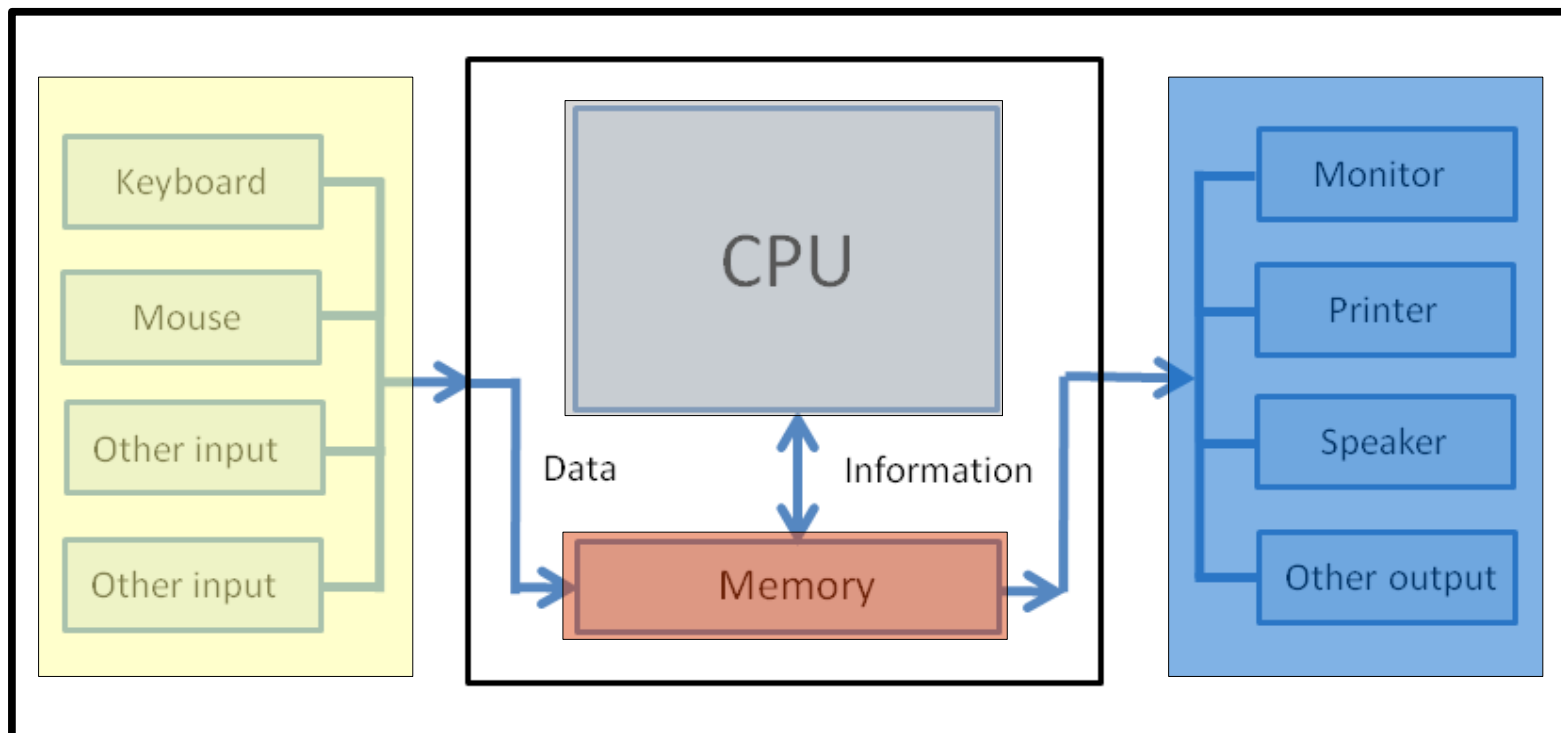
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
     $x \leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



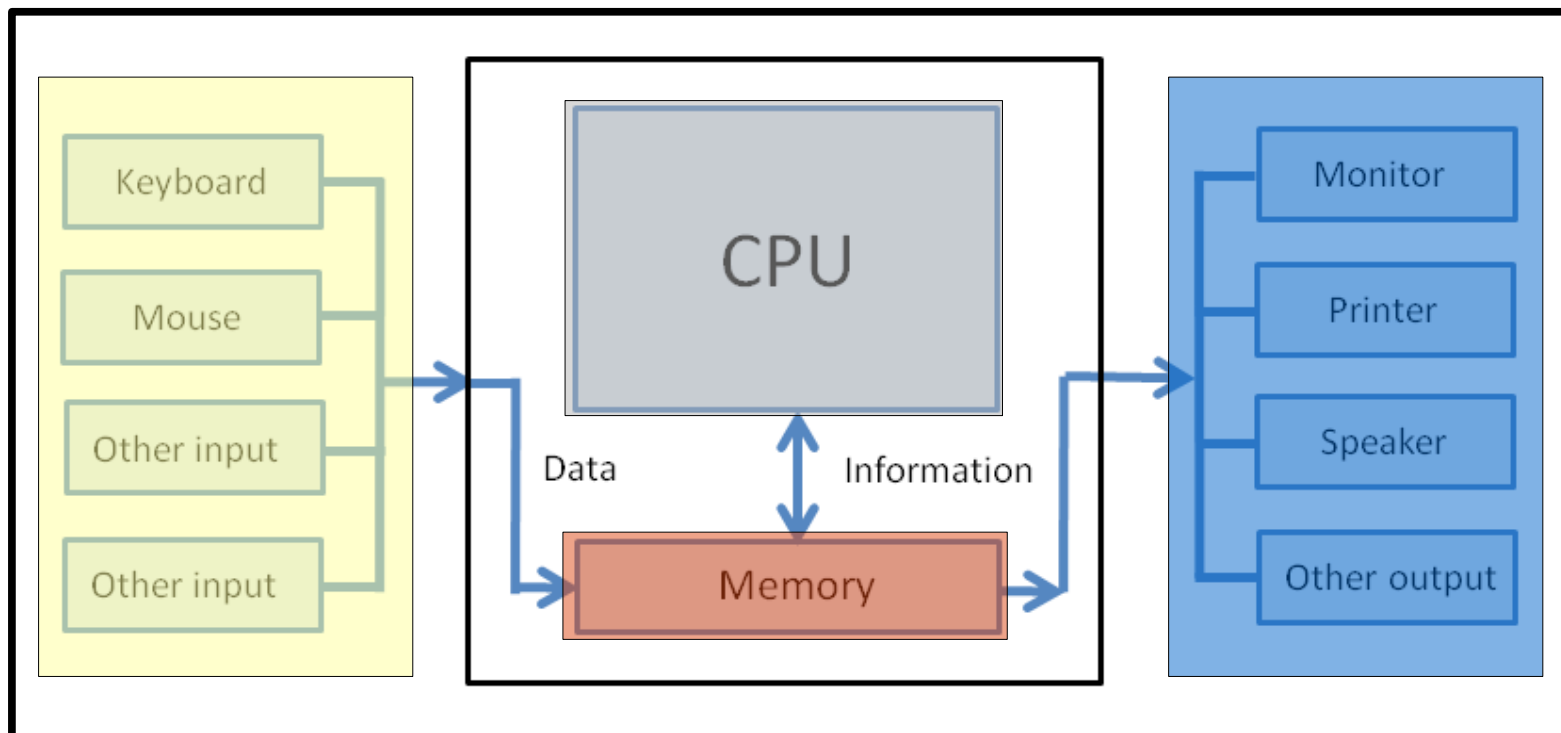
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



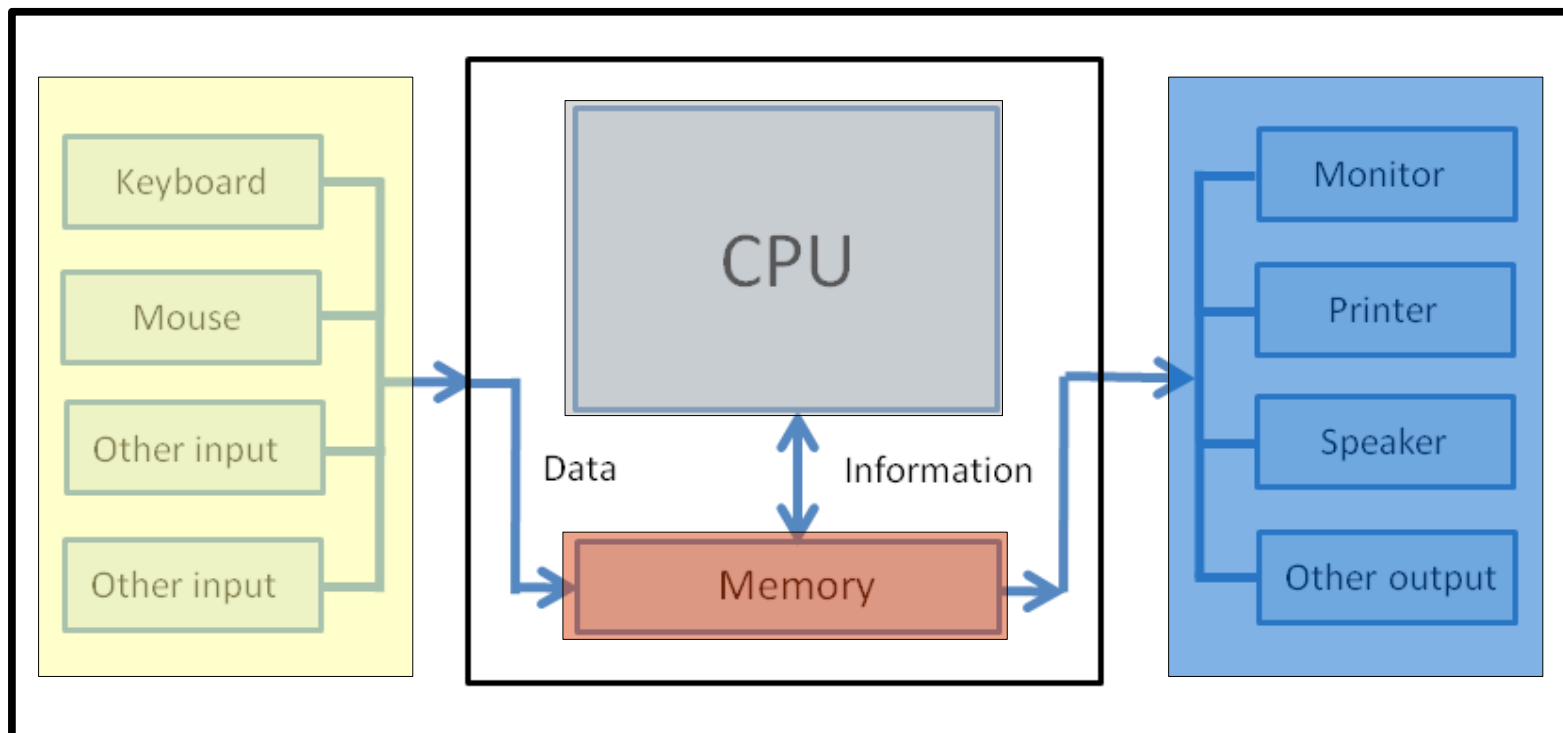
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



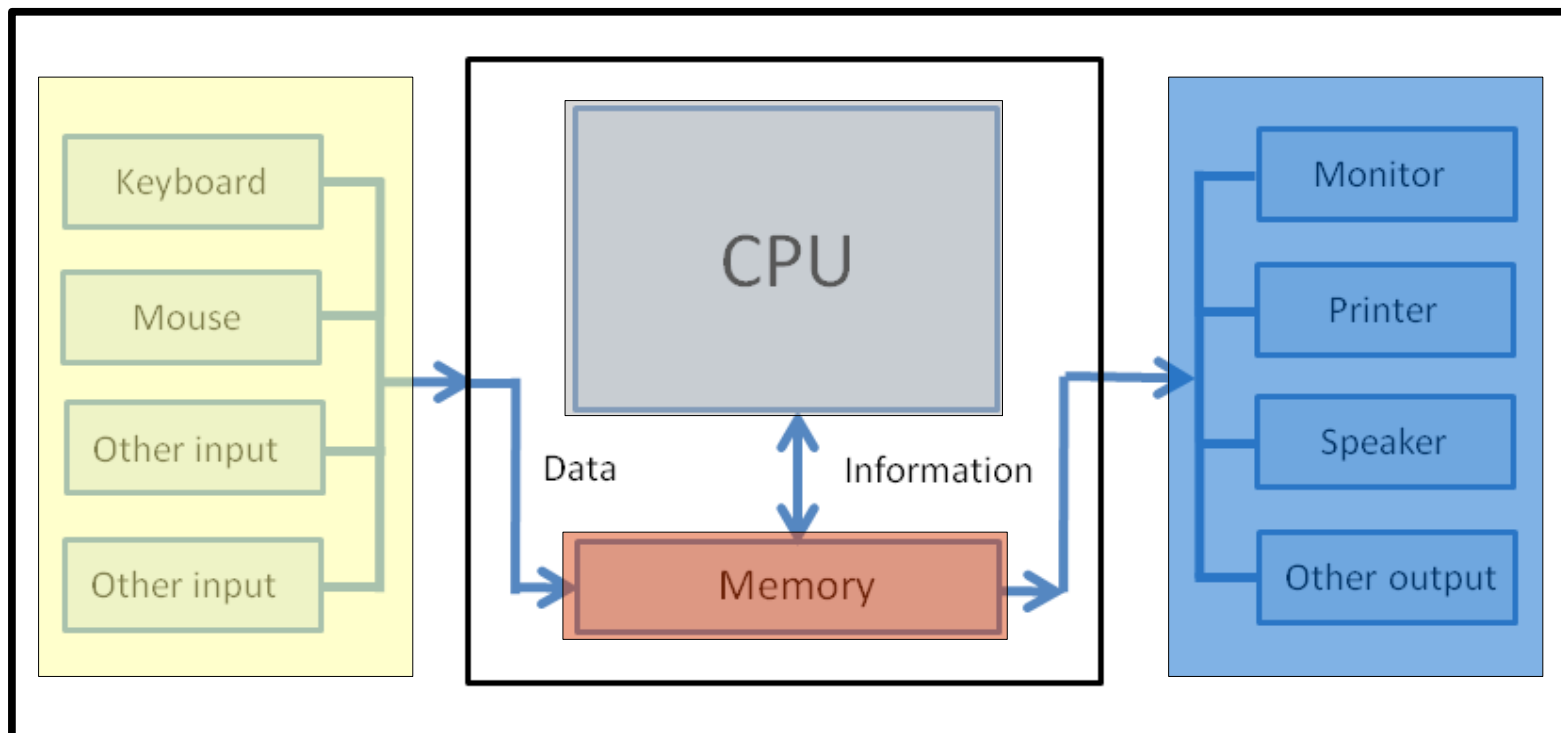
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



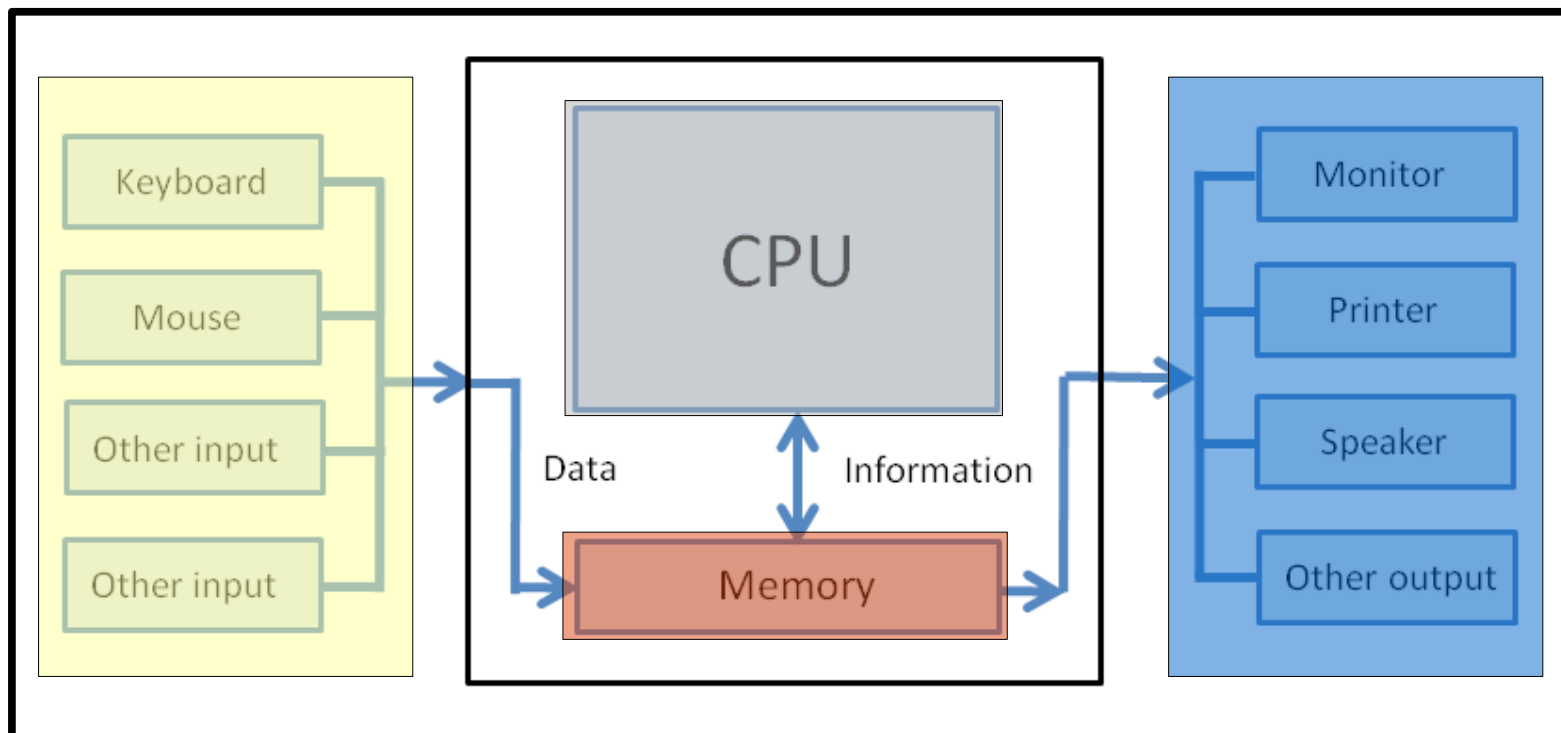
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



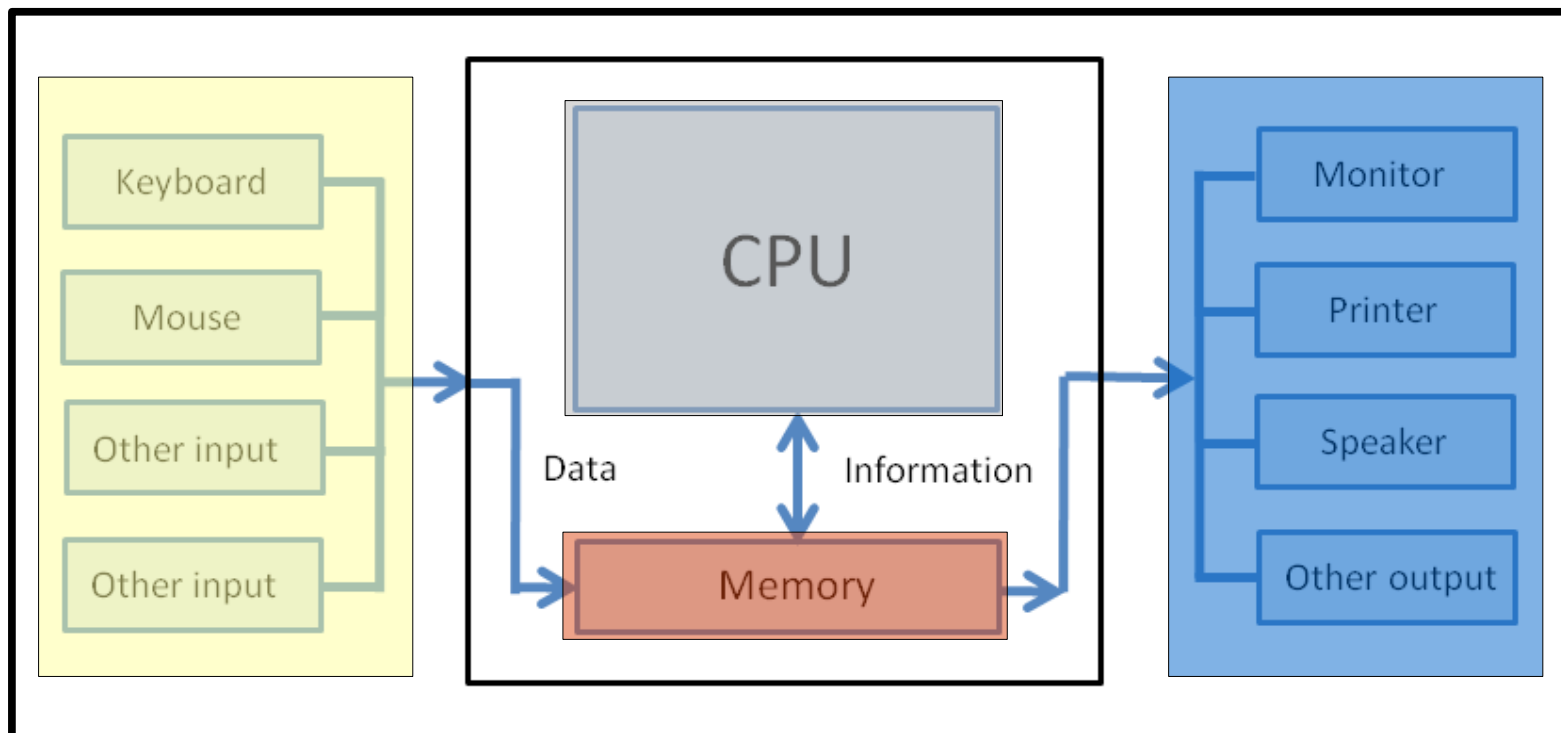
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



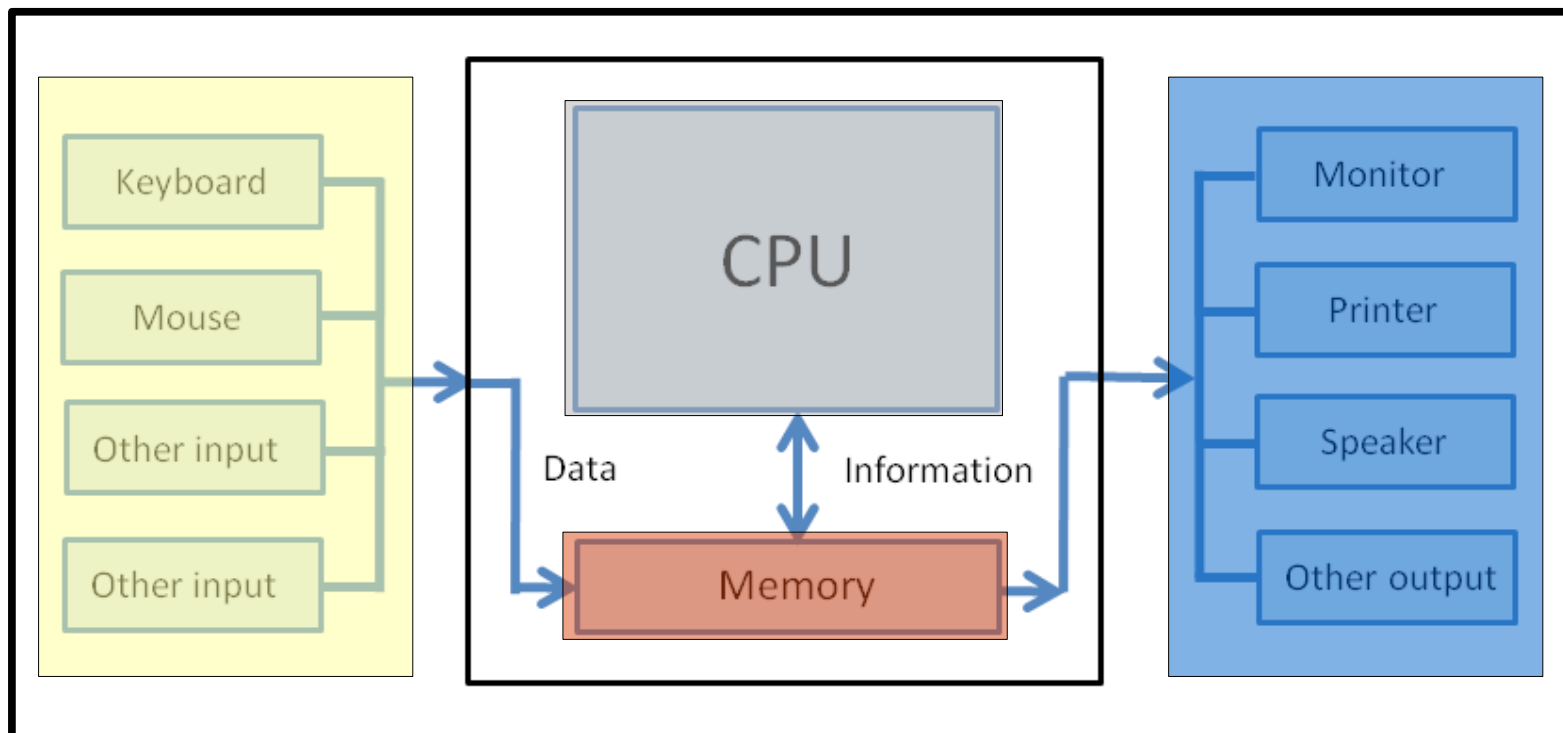
Escrita

$x \leftarrow$

```
var integer x  $\leftarrow$  25;  
var integer y  $\leftarrow$  25;  
emit GRAPHICS_DRAW_PIXEL(x,y);  
loop do  
  var integer key  $\leftarrow$  await KEY_PRESS;  
  if key = KEY_LEFT then  
    x  $\leftarrow$  x - 1;  
  else/if key = <...> then  
    <...>  
  end  
  emit GRAPHICS_DRAW_PIXEL(x,y);  
end
```

Leitura

x



Exercício 2.2

Exercício 2.2

- Movimentar dois pixels ao mesmo tempo

Exercício 2.2

- Movimentar dois pixels ao mesmo tempo
- Usar conjunto de teclas diferentes
 - como se fossem dois jogadores

Exercício 2.2

- Movimentar dois pixels ao mesmo tempo
- Usar conjunto de teclas diferentes
 - como se fossem dois jogadores
- <https://fsantanna.github.io/pico-ceu/out/manual/v0.30/input/>