

Informática II

Funcionamiento de un μ P y μ C

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2024 –

Programación en lenguaje C

- ¿Cuál es el primer programa realizado en “*Informática I*”?

Programación en lenguaje C

- ¿Cuál es el primer programa realizado en “*Informática I*”?

hola_mundo.c

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Hola mundo.\n");
6     return 0;
7 }
```

Programación en lenguaje C

- ▶ ¿Cuál es el primer programa realizado en “*Informática I*”?
- ▶ ¿Qué otro programa simple recuerda?

Programación en lenguaje C

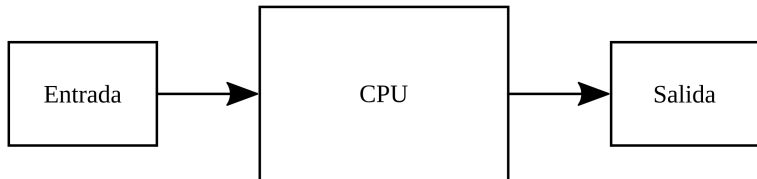
- ▶ ¿Cuál es el primer programa realizado en “*Informática I*”?
- ▶ ¿Qué otro programa simple recuerda?

suma1.c

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int entero1, entero2, suma;
6
7      printf("Ingrese el primer entero: ");
8      scanf("%d", &entero1);
9      printf("Ingrese el segundo entero: ");
10     scanf("%d", &entero2);
11
12     suma = entero1 + entero2;
13     printf("La suma es: %d\n", suma);
14     return 0;
15 }
```

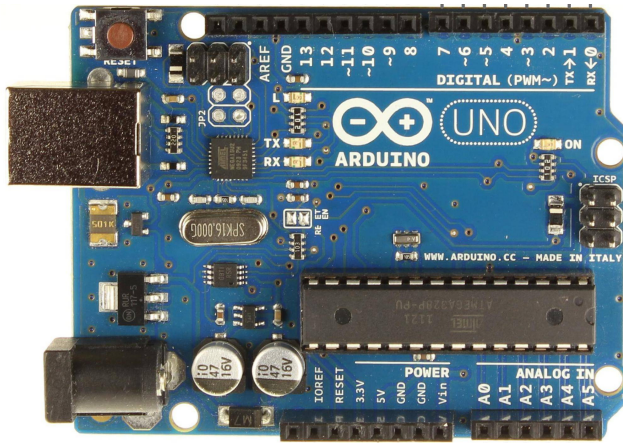
Programación en lenguaje C

- ▶ ¿Cuál es el primer programa realizado en “*Informática I*”?
- ▶ ¿Qué otro programa simple recuerda?



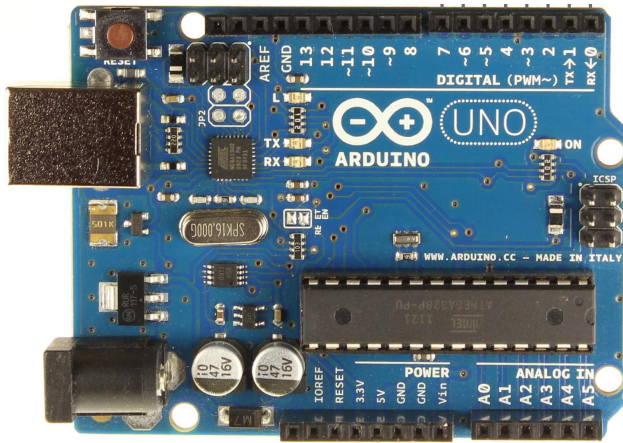
Programación en lenguaje C

Placa Arduino UNO



Programación en lenguaje C

Placa Arduino UNO



Cuenta con el μ C ATmega328 (8-bits de arquitectura AVR RISC).

Programación en lenguaje C

"Hola mundo" para el ATmega328

```
1 #include <avr/io.h>
2
3
4
5 int main (void)
6 {
7     DDRB |= (1 << DDB5); /* Pin 5 de PORTB como salida */
8
9     while(1)
10    {
11        PORTB |= (1 << PORTB5); /* Pin 5 en 1, enciende el LED */
12
13
14        PORTB &= ~(1 << PORTB5); /* Pin 5 en 0, apaga el LED */
15
16    }
17    return 0;
18 }
```

Programación en lenguaje C

"Hola mundo" para el ATmega328

```
1 #include <avr/io.h>
2 #include <util/delay.h>
3 #define BLINK_DELAY_MS 1000
4
5 int main (void)
6 {
7     DDRB |= (1 << DDB5); /* Pin 5 de PORTB como salida */
8
9     while(1)
10    {
11        PORTB |= (1 << PORTB5); /* Pin 5 en 1, enciende el LED */
12        _delay_ms(BLINK_DELAY_MS);
13
14        PORTB &= ~(1 << PORTB5); /* Pin 5 en 0, apaga el LED */
15        _delay_ms(BLINK_DELAY_MS);
16    }
17    return 0;
18 }
```

Programación en lenguaje C

(volviendo a la PC)

suma2.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     int c = a + b;
5 }
```

Programación en lenguaje C

(volviendo a la PC)

suma2.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     int c = a + b;
5 }
```

suma3.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     return (a + b);
5 }
```

Programación en lenguaje C

(volviendo a la PC)

suma2.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     int c = a + b;
5 }
```

suma3.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     return (a + b);
5 }
```

¿Qué diferencia encuentra entre `hola_mundo.c`, `suma1.c`, `suma2.c` y `suma3.c`?

Programación en lenguaje C

(volviendo a la PC)

suma2.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     int c = a + b;
5 }
```

suma3.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     return (a + b);
5 }
```

¿Qué diferencia encuentra entre `hola_mundo.c`, `suma1.c`, `suma2.c` y `suma3.c`?

1. `hola_mundo.c` utiliza funciones de la biblioteca estándar de entrada/salida.

Programación en lenguaje C

(volviendo a la PC)

suma2.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     int c = a + b;
5 }
```

suma3.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     return (a + b);
5 }
```

¿Qué diferencia encuentra entre `hola_mundo.c`, `suma1.c`, `suma2.c` y `suma3.c`?

1. `hola_mundo.c` utiliza funciones de la biblioteca estándar de entrada/salida.
2. `suma1.c` tiene interacción con el usuario (`stdin`, `stdout`).

Programación en lenguaje C

(volviendo a la PC)

suma2.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     int c = a + b;
5 }
```

suma3.c

```
1 int main(void)
2 {
3     int a = 2, b = 3;
4     return (a + b);
5 }
```

¿Qué diferencia encuentra entre `hola_mundo.c`, `suma1.c`, `suma2.c` y `suma3.c`?

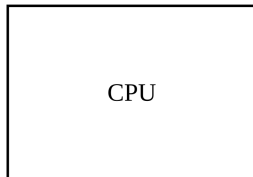
1. `hola_mundo.c` utiliza funciones de la biblioteca estándar de entrada/salida.
2. `suma1.c` tiene interacción con el usuario (`stdin`, `stdout`).
3. `suma2.c` y `suma3.c` no tienen interacción con el usuario.

¿Cómo funciona un μ P (PC)?

¿Qué función cumple una PC?

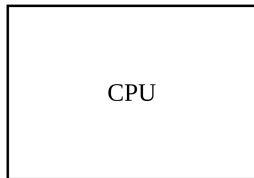
¿Cómo funciona un μ P (PC)

Procesa datos



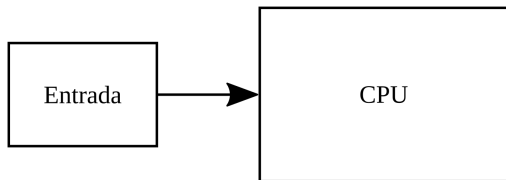
¿Cómo funciona un μ P (PC)

¿De donde se obtienen los datos?



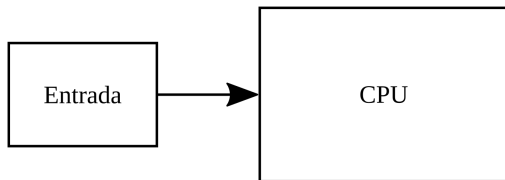
¿Cómo funciona un μ P (PC)

¿De donde se obtienen los datos?



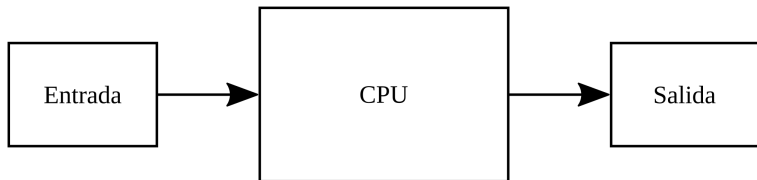
¿Cómo funciona un μ P (PC)

¿Cómo pone a disposición el resultado?



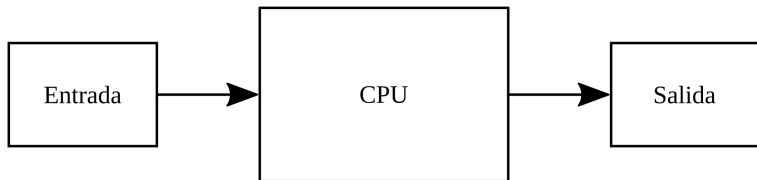
¿Cómo funciona un μ P (PC)

¿Cómo pone a disposición el resultado?



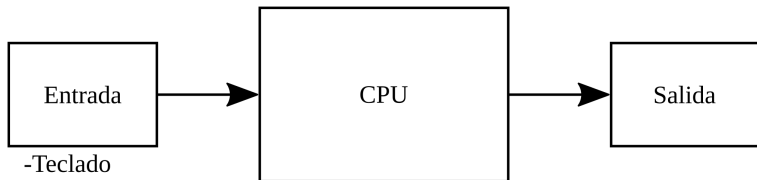
¿Cómo funciona un μ P (PC)

¿Qué dispositivos de **entrada** conoce?



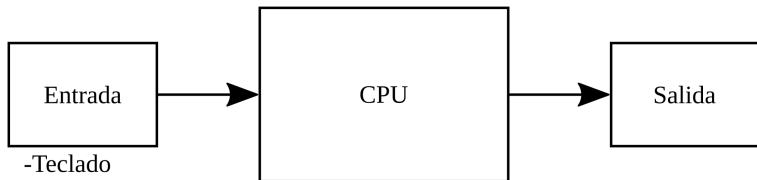
¿Cómo funciona un μ P (PC)

¿Qué dispositivos de **entrada** conoce?



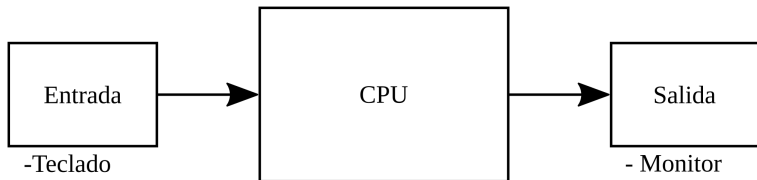
¿Cómo funciona un μ P (PC)

¿Qué dispositivos de **salida** conoce?



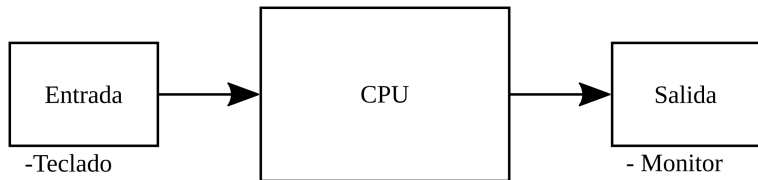
¿Cómo funciona un μ P (PC)

¿Qué dispositivos de **salida** conoce?



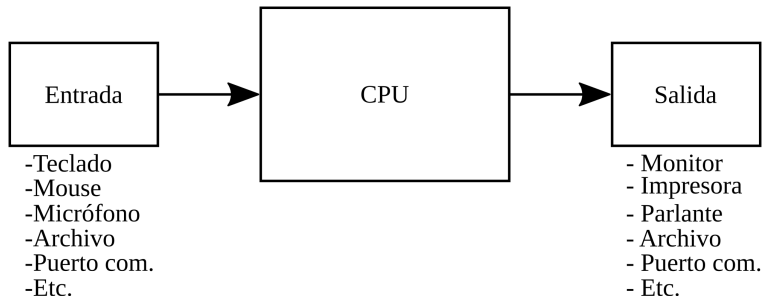
¿Cómo funciona un μ P (PC)

Otros dispositivos de entrada/salida



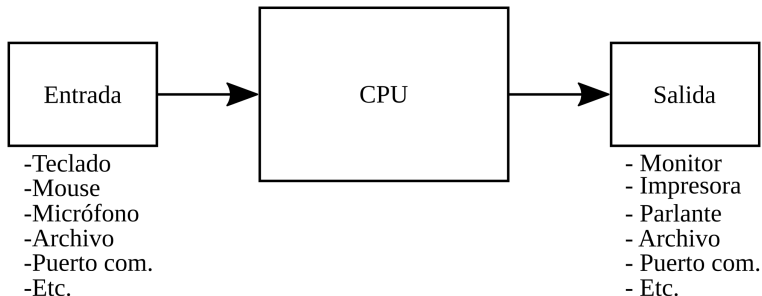
¿Cómo funciona un μ P (PC)

Otros dispositivos de entrada/salida



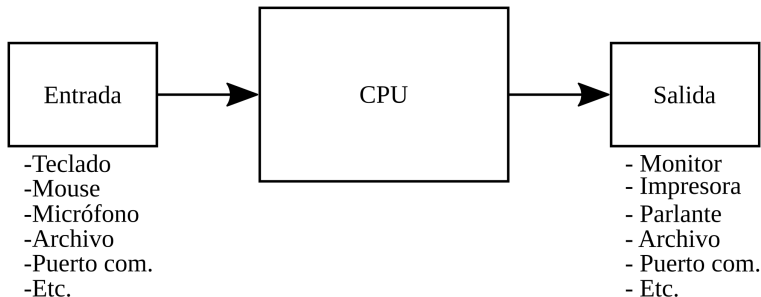
¿Cómo funciona un μ P (PC)

Analice el ejemplo de suma dos números (programa en C)



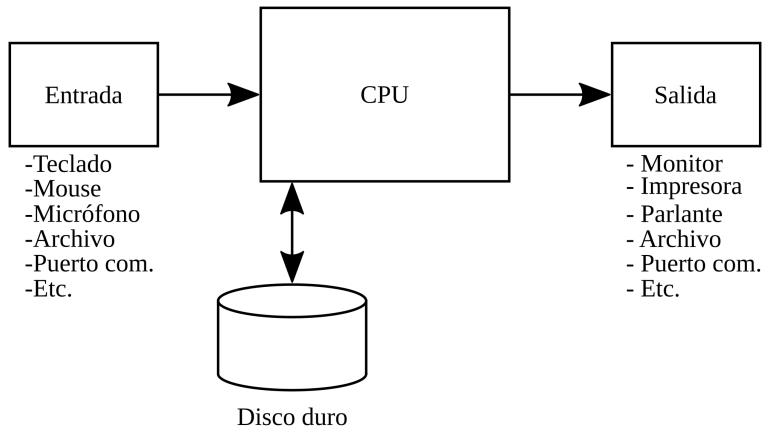
¿Cómo funciona un μ P (PC)

¿Dónde se guarda el programa **antes** de su ejecución?



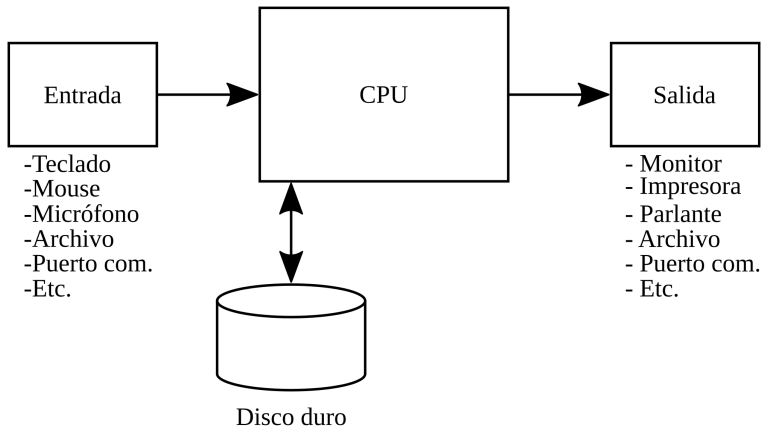
¿Cómo funciona un μ P (PC)

¿Dónde se guarda el programa **antes** de su ejecución?



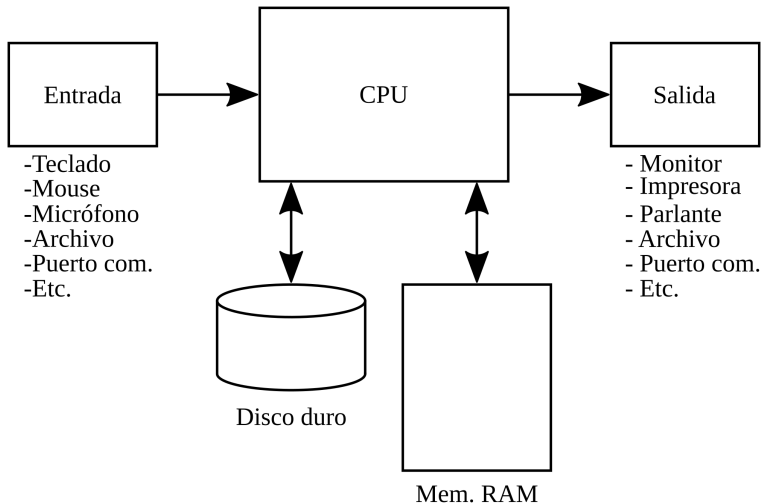
¿Cómo funciona un μ P (PC)

¿Dónde se guarda el programa **durante** su ejecución?



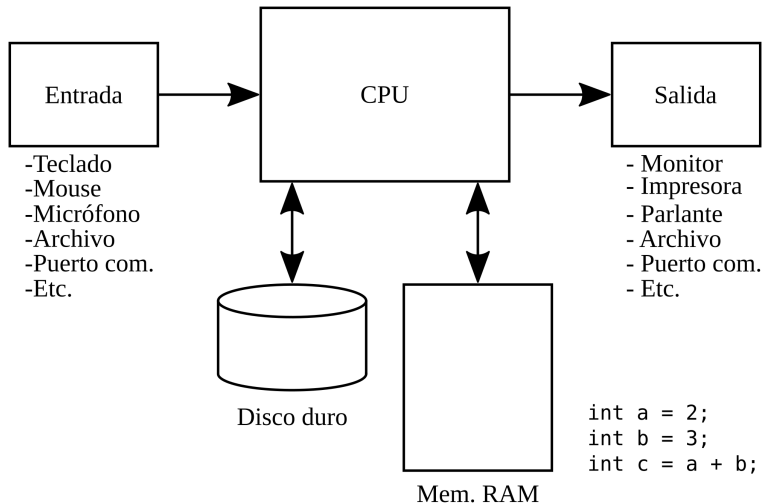
¿Cómo funciona un μ P (PC)

¿Dónde se guarda el programa **durante** su ejecución?



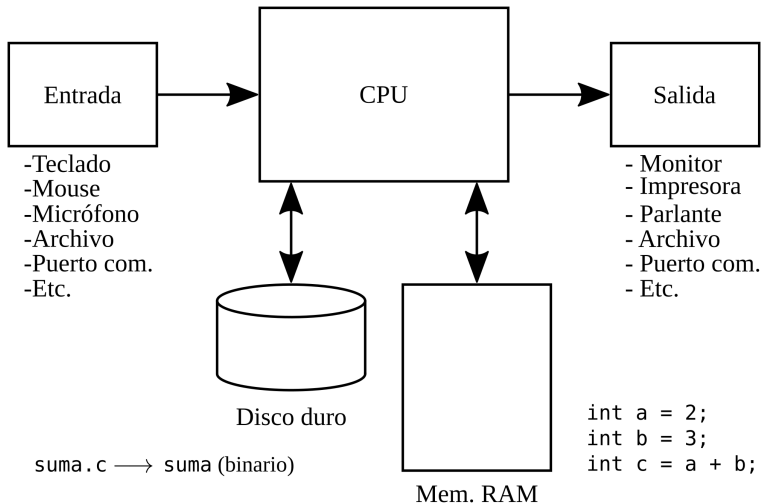
¿Cómo funciona un μ P (PC)

Programa que suma dos números



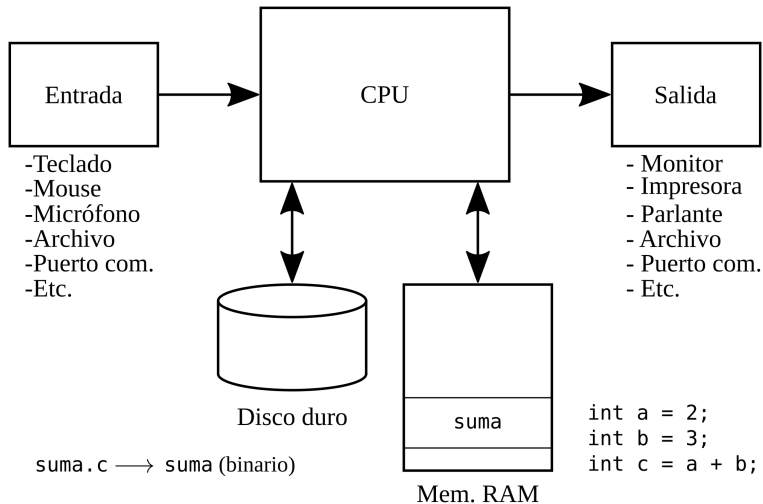
¿Cómo funciona un μ P (PC)

Programa que suma dos números



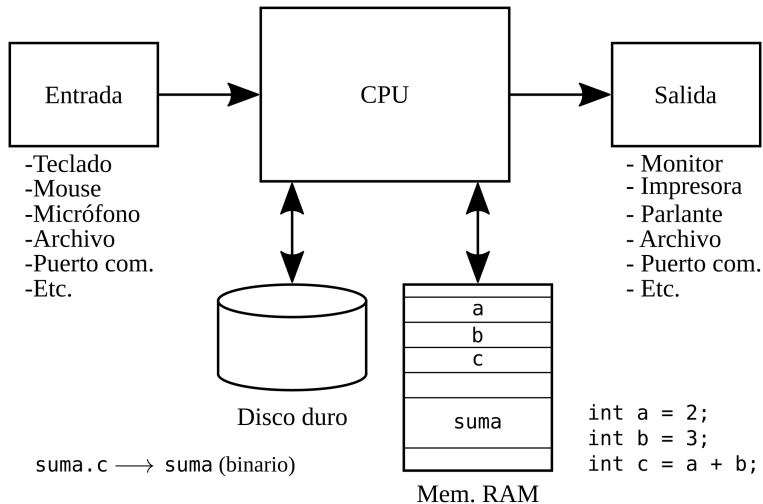
¿Cómo funciona un μ P (PC)

Programa que suma dos números



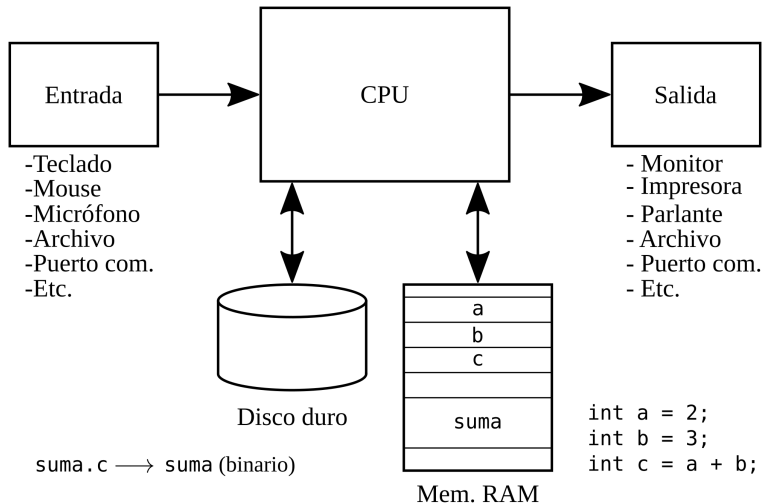
¿Cómo funciona un μ P (PC)

Programa que suma dos números



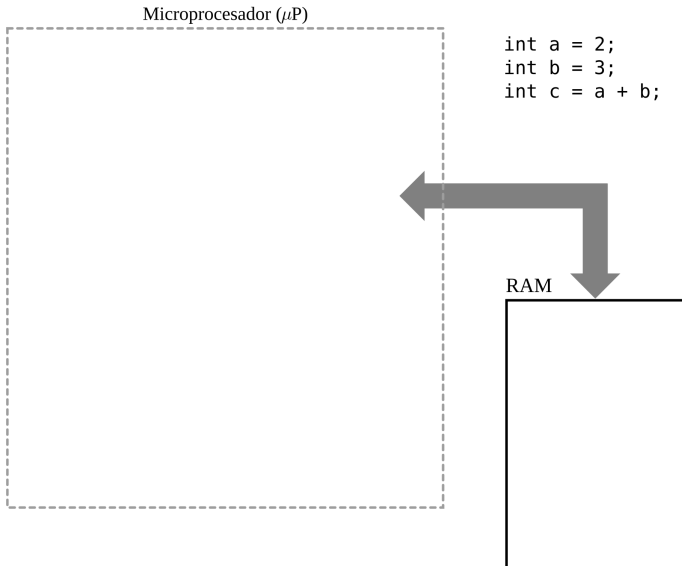
¿Cómo funciona un μ P (PC)

El tráfico de datos está centrado en el μ P



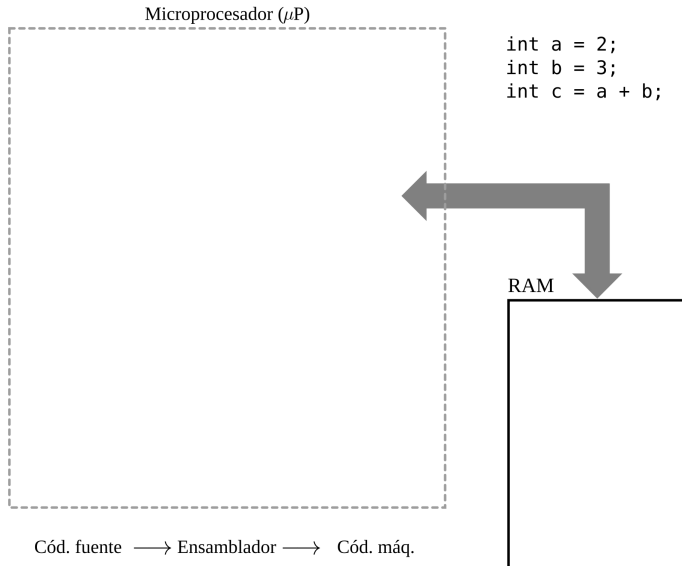
¿Cómo funciona un μ P (PC)

¿Qué formato tiene el programa ejecutable en memoria RAM?



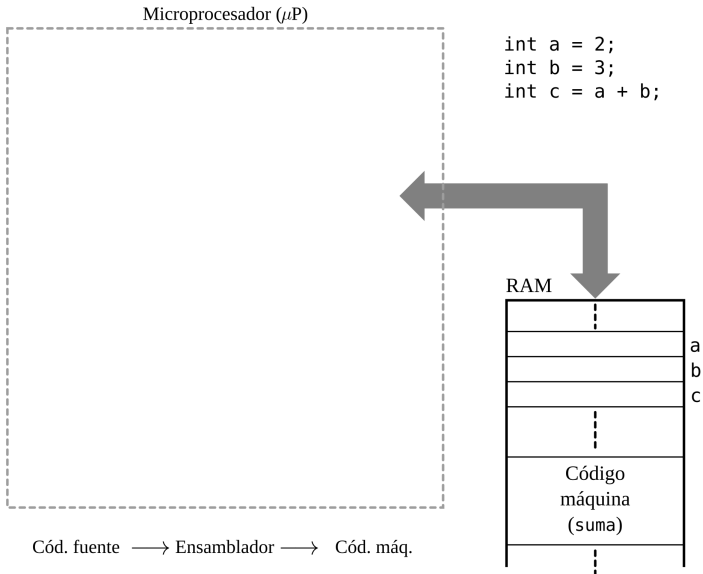
¿Cómo funciona un μ P (PC)

¿Qué formato tiene el programa ejecutable en memoria RAM?



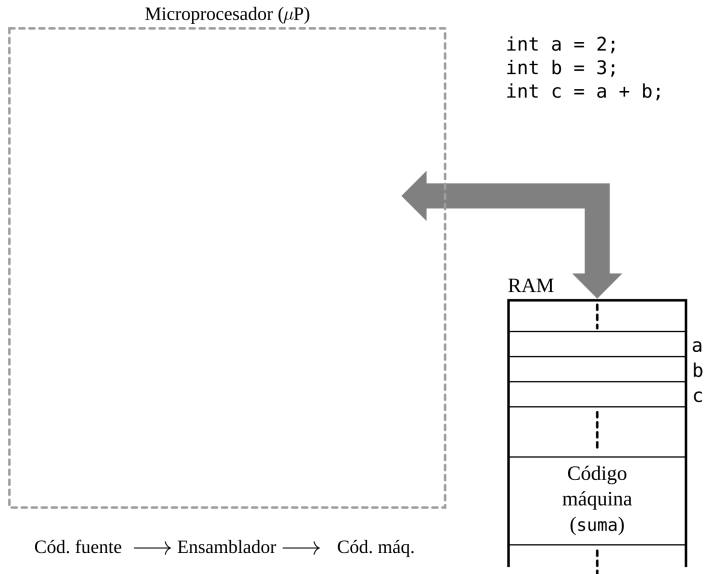
¿Cómo funciona un μ P (PC)

¿Qué formato tiene el programa ejecutable en memoria RAM?



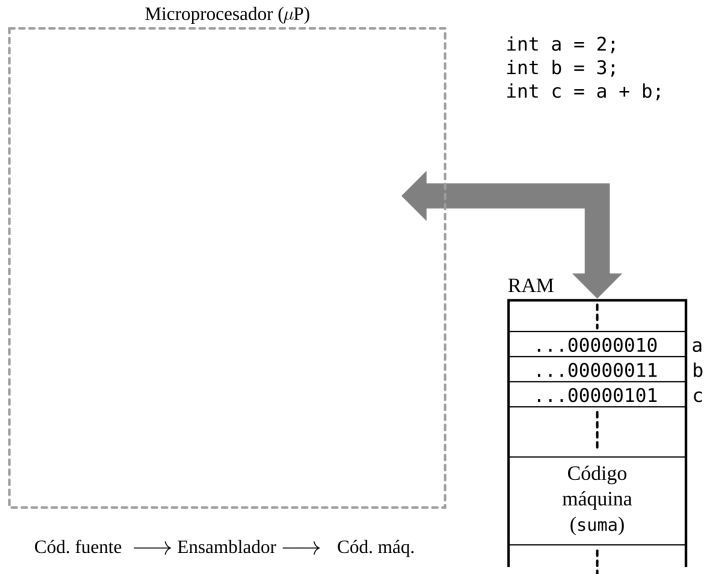
¿Cómo funciona un μ P (PC)

¿Qué formato tienen los datos en memoria RAM?



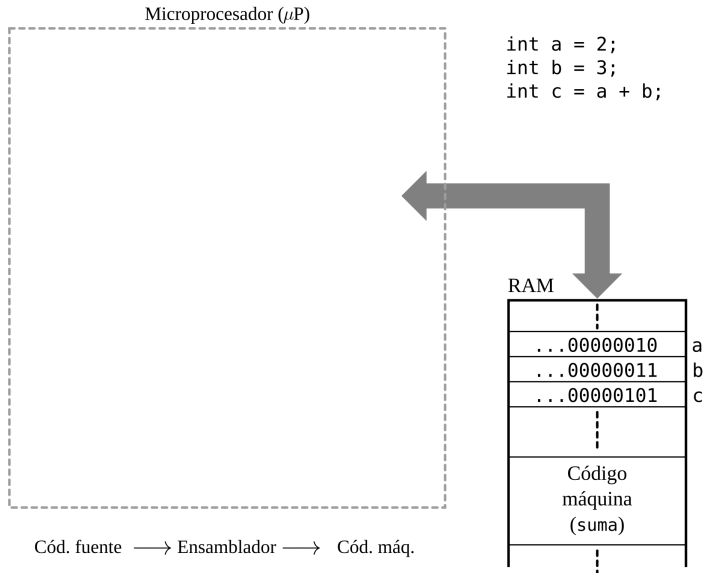
¿Cómo funciona un μ P (PC)

¿Qué formato tienen los datos en memoria RAM?



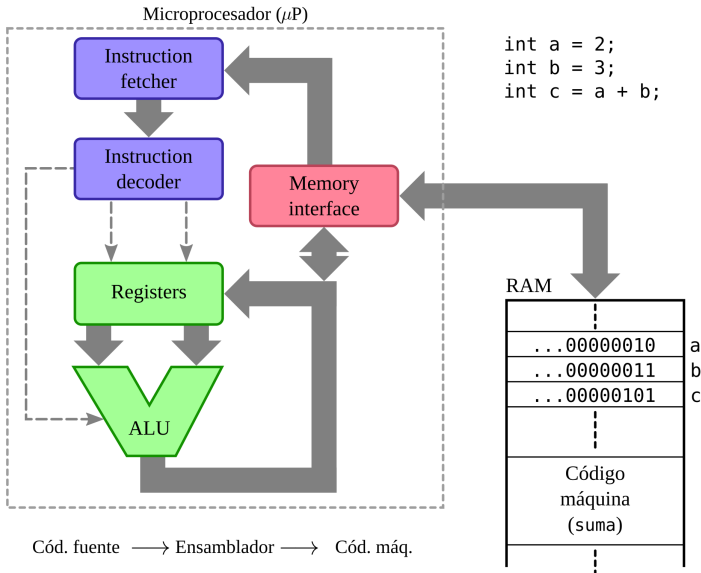
¿Cómo funciona un μ P (PC)

¿Cómo **decodifica** el programa la CPU para procesar los datos?

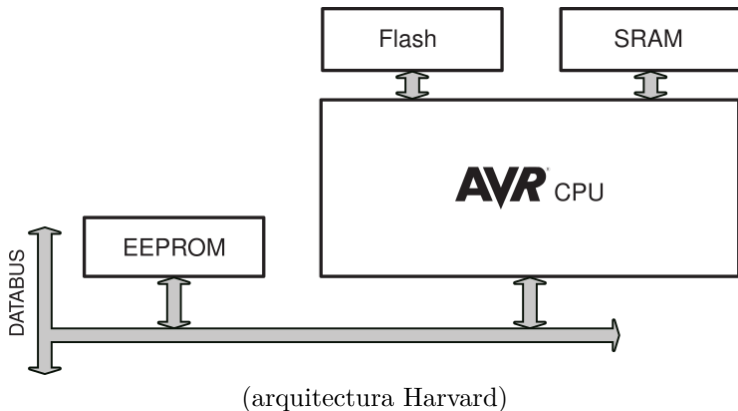


¿Cómo funciona un μP (PC)

¿Cómo **decodifica** el programa la CPU para procesar los datos?



Microcontrolador (μ C) ATmega328



Microcontrolador (μ C) ATmega328

