INFORMATICA I

Arreglos o " **vectores** "

Ing.Juan Carlos Cuttitta

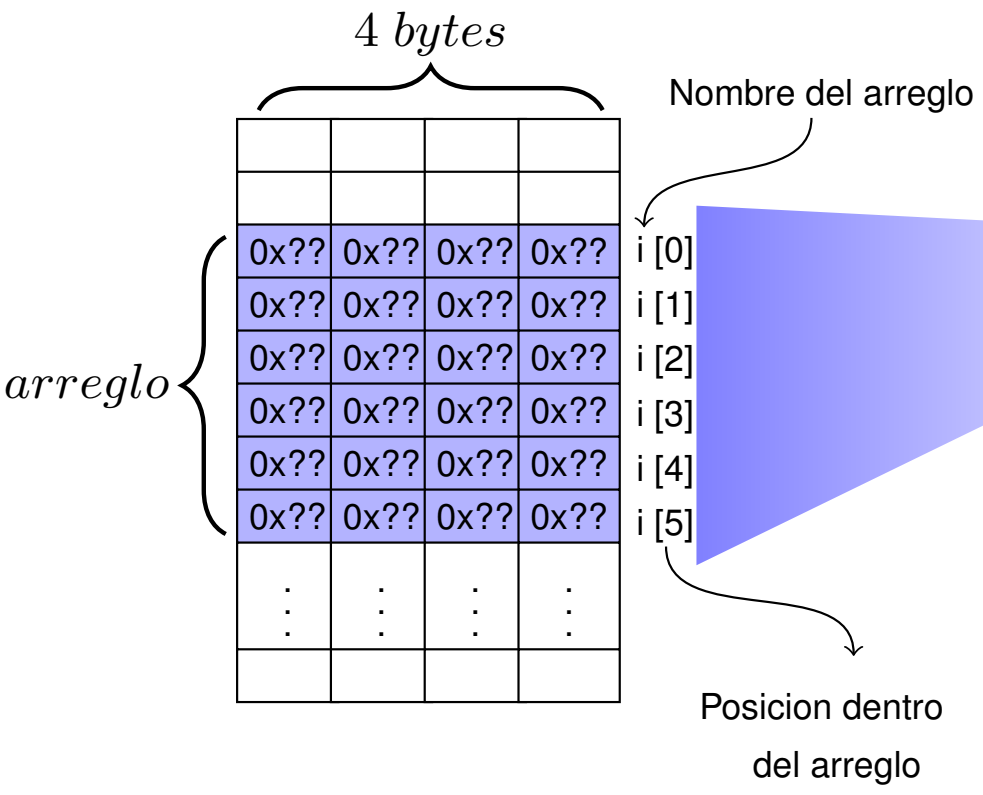
Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Departamento de Ingeniería Electrónica

24 de mayo de 2020

Declaración y disposición en memoria

Arquitectura X86-32 bits

Disposición de la variable i en memoria



Código en programa fuente

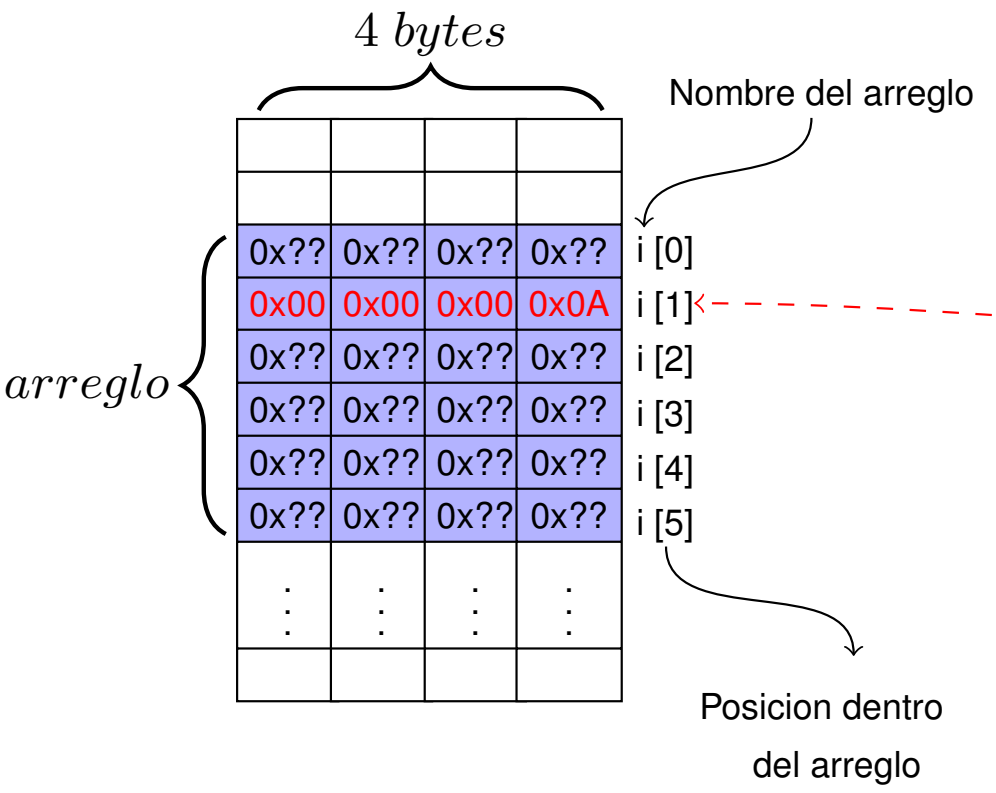
```
1   int main ()
2   {
3     int i[6];
4     .......
5     i[1]=10;
6     .......
7   }
```

Declara arreglo de 6 enteros

Acceso al contenido

Arquitectura X86-32 bits

Disposición de la variable i en memoria



Código en programa fuente

```
1   int main ()
2   {
3     int i[6];
4     .......
5     i[1]=10;
6     .......
7   }
```

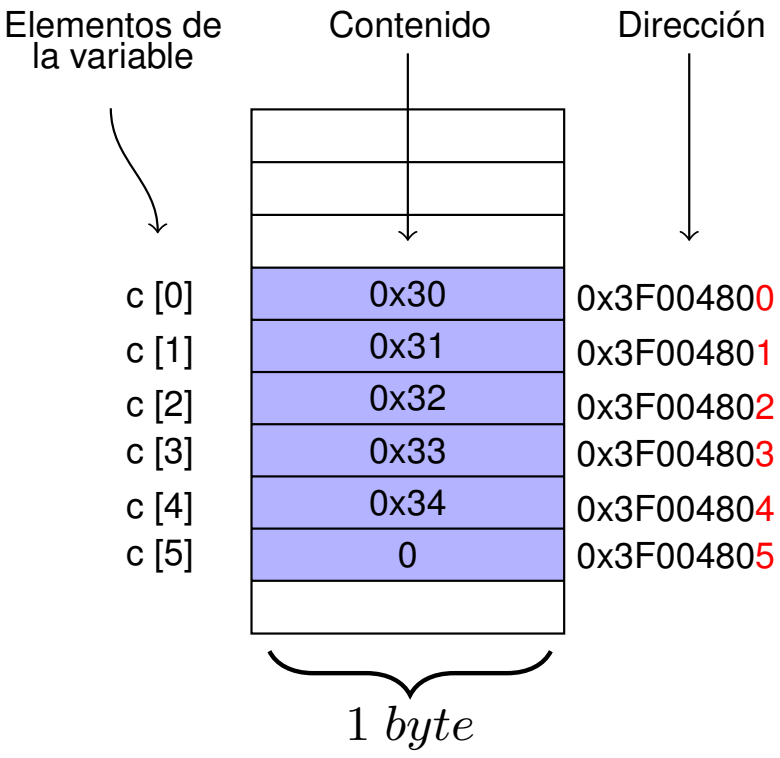
Asigna el valor 10 al elemento 1 del arreglo

Direción no es lo mismo que orden de elemento

Declaración en código

unsigned char c[6]

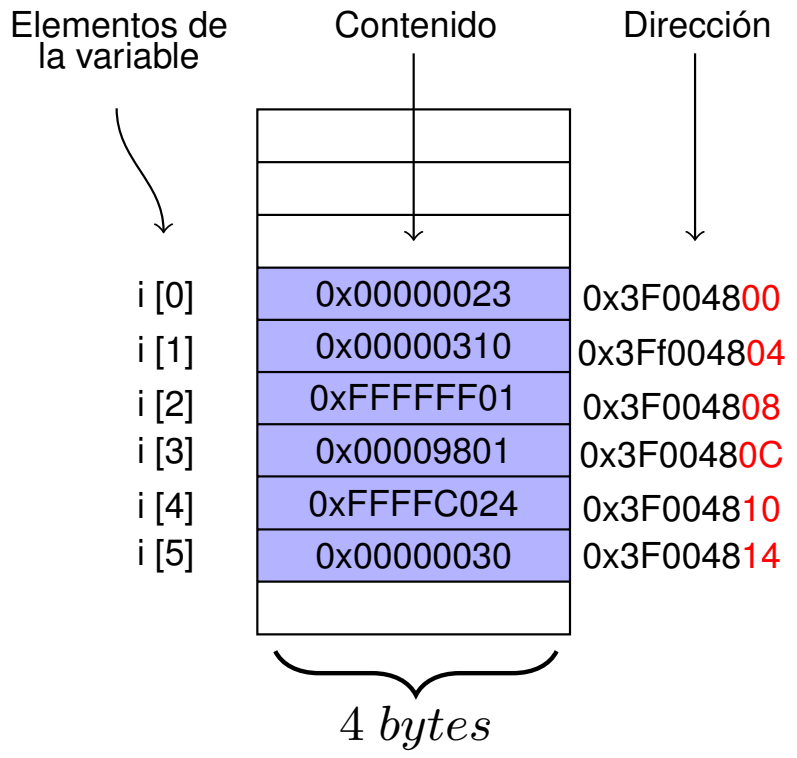
Cardena de caracteres



Declaración en código

int c[6]

Arreglo de enteros

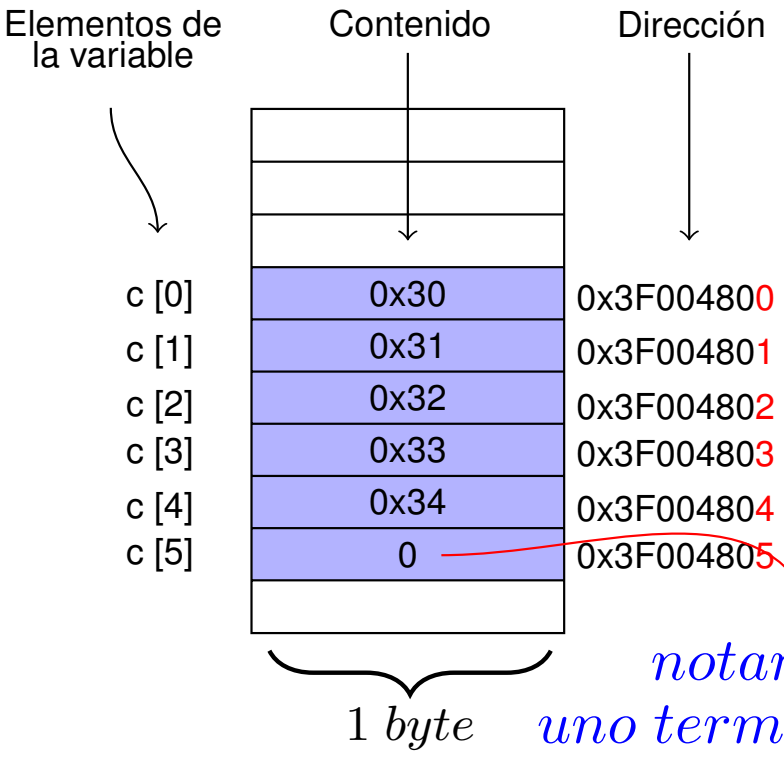


Direción no es lo mismo que orden de elemento

Declaración en código

unsigned char c[6]

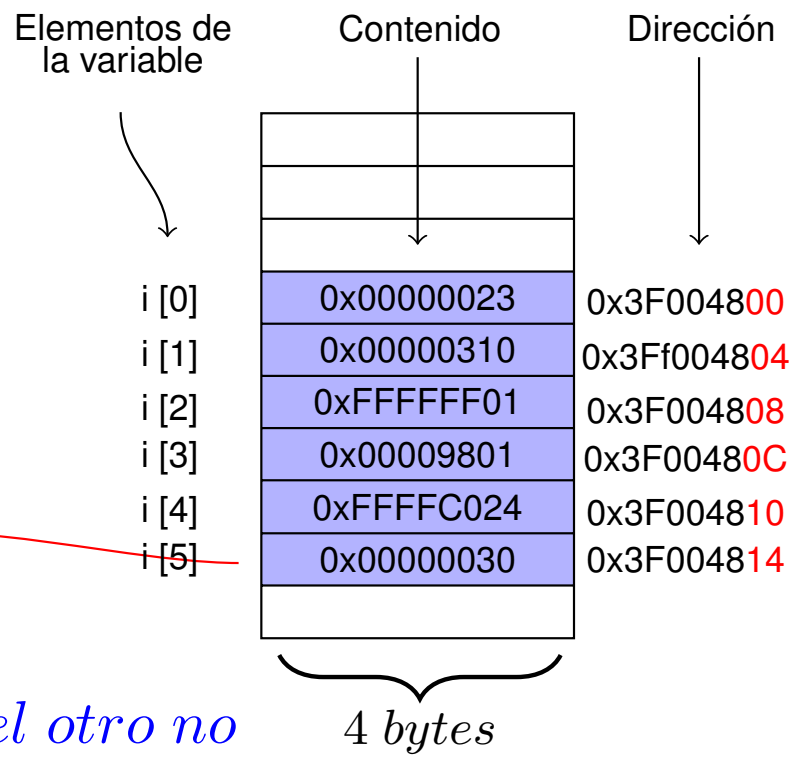
Cardena de caracteres



Declaración en código

int c[6]

Arreglo de enteros



notar la diferencia

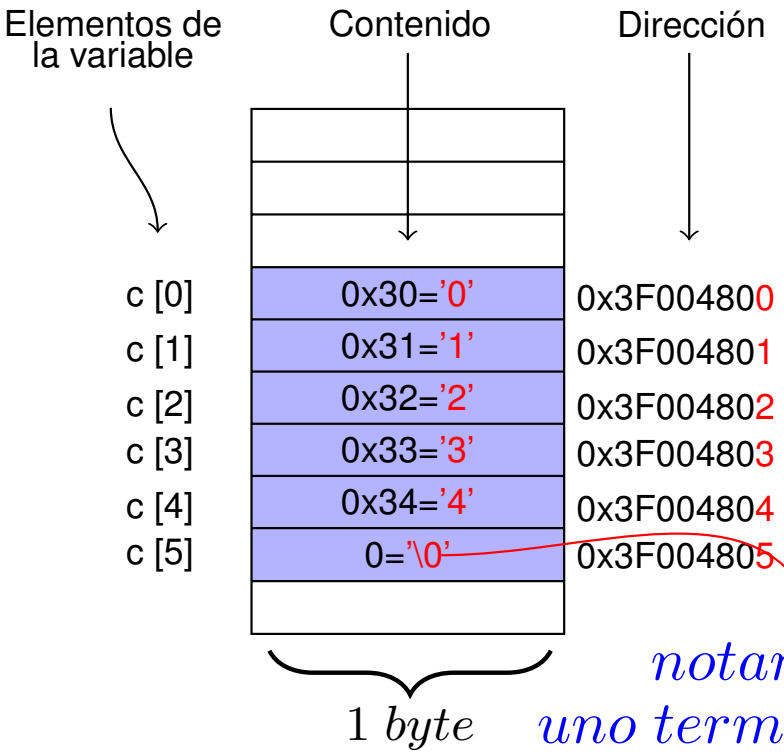
uno termina con '\0' y el otro no

Direción no es lo mismo que orden de elemento

Declaración en código

unsigned char c[6]

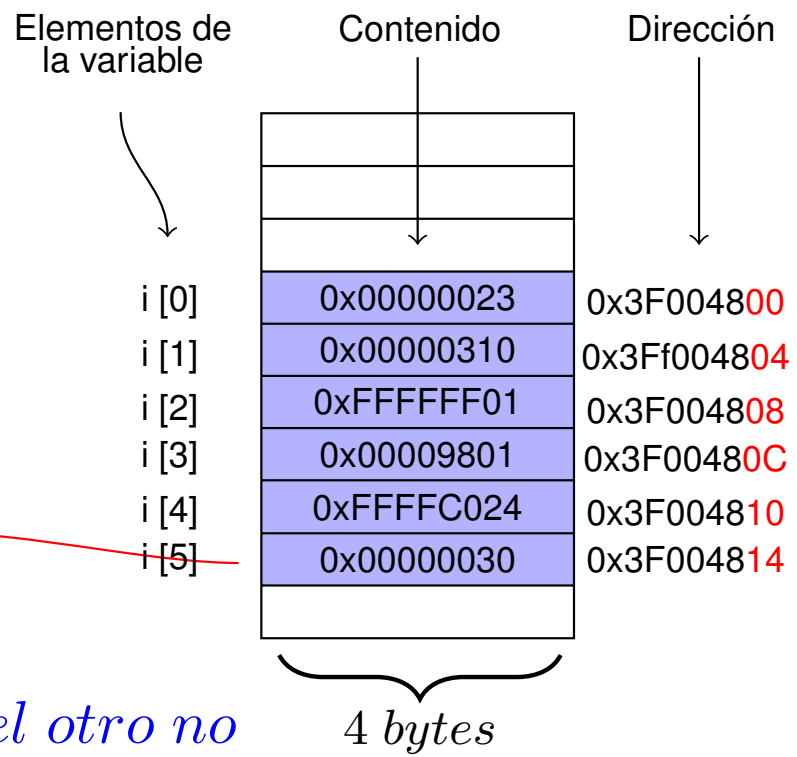
Cardena de caracteres



Declaración en código

int c[6]

Arreglo de enteros



notar la diferencia

uno termina con '\0' y el otro no

Un caso especial de arreglo

Arreglo de caracteres

Se trata de un tipo muy usual de dato que llamamos cadena o string (del inglés).
Se inicializa de los siguientes modos:

```
1 /*Una forma de inicializar un areglo de caracteres con una
      cadena*/
2
3 char cad []= "Hola mundo!";
4
5 /* Otra forma...*/
6
7 char cad []={'H','o','l','a',' ','m','u','n','d','o','!','\0'};
```

Un caso especial de arreglo

Arreglo de caracteres

Se trata de un tipo muy usual de dato que llamamos cadena o string (del inglés).
Se inicializa de los siguientes modos:

```
1 /*Una forma de inicializar un areglo de caracteres con una
      cadena*/
2
3 char cad []= "Hola mundo!";
4
5 /* Otra forma...*/
6
7 char cad []={'H','o','l','a',' ','m','u','n','d','o','!','\0'};
```

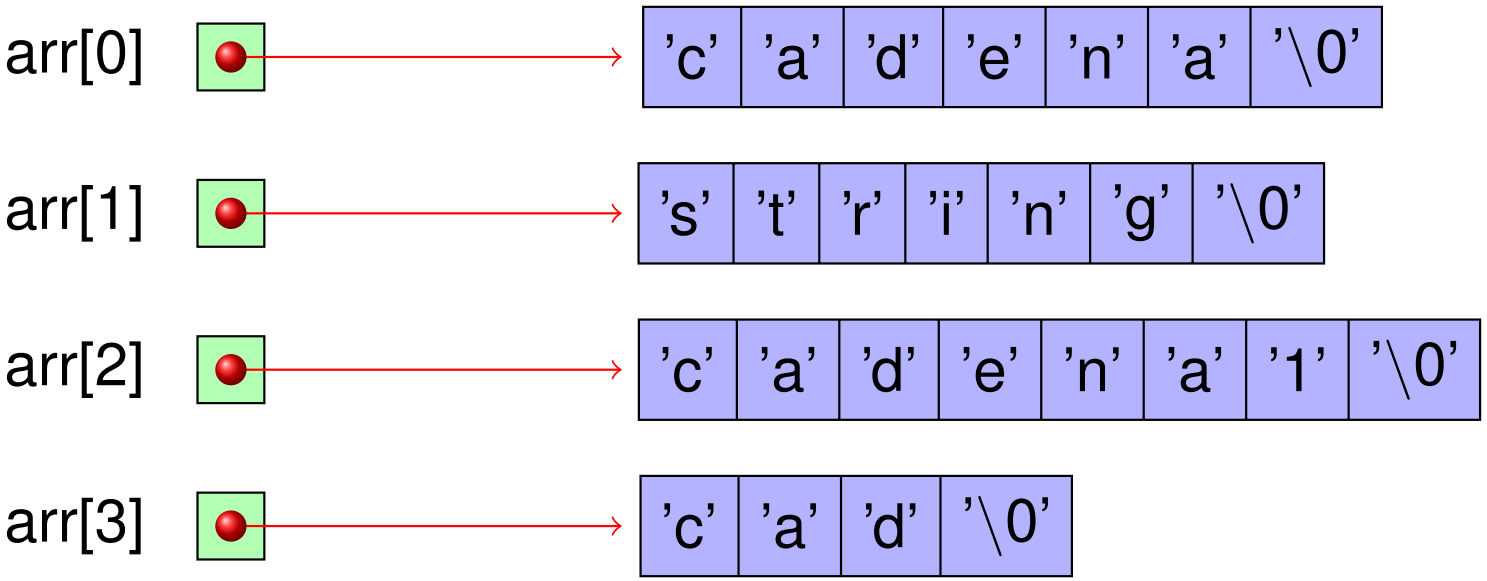
En código....

```
1   /* Utiliza una lista de inicialización para
       inicializar el arreglo arr.*/
2   int arr[8] = {23,0,-669,-1,995,1277,90,-9000};
3   /* La cantidad de elementos en la línea anterior
       es redundante. Puede hacerse lo mismo de la
       siguiente forma*/
4   int arr[] = {23,0,-669,-1,995,1277,90,-9000};
5   /*Si lo vamos a inicializar con el mismo valor
       para todos los elementos, la forma adecuada
       es la siguiente*/
6
7   int arr[8], i;
8   for (i = 0 ; i < 8 ; i++)
9   {
10    arr[i] = 0;
11  }
```

Arreglos de punteros

```
1       char *arr[4]={"cadena","string","cadena1","cad"};
```

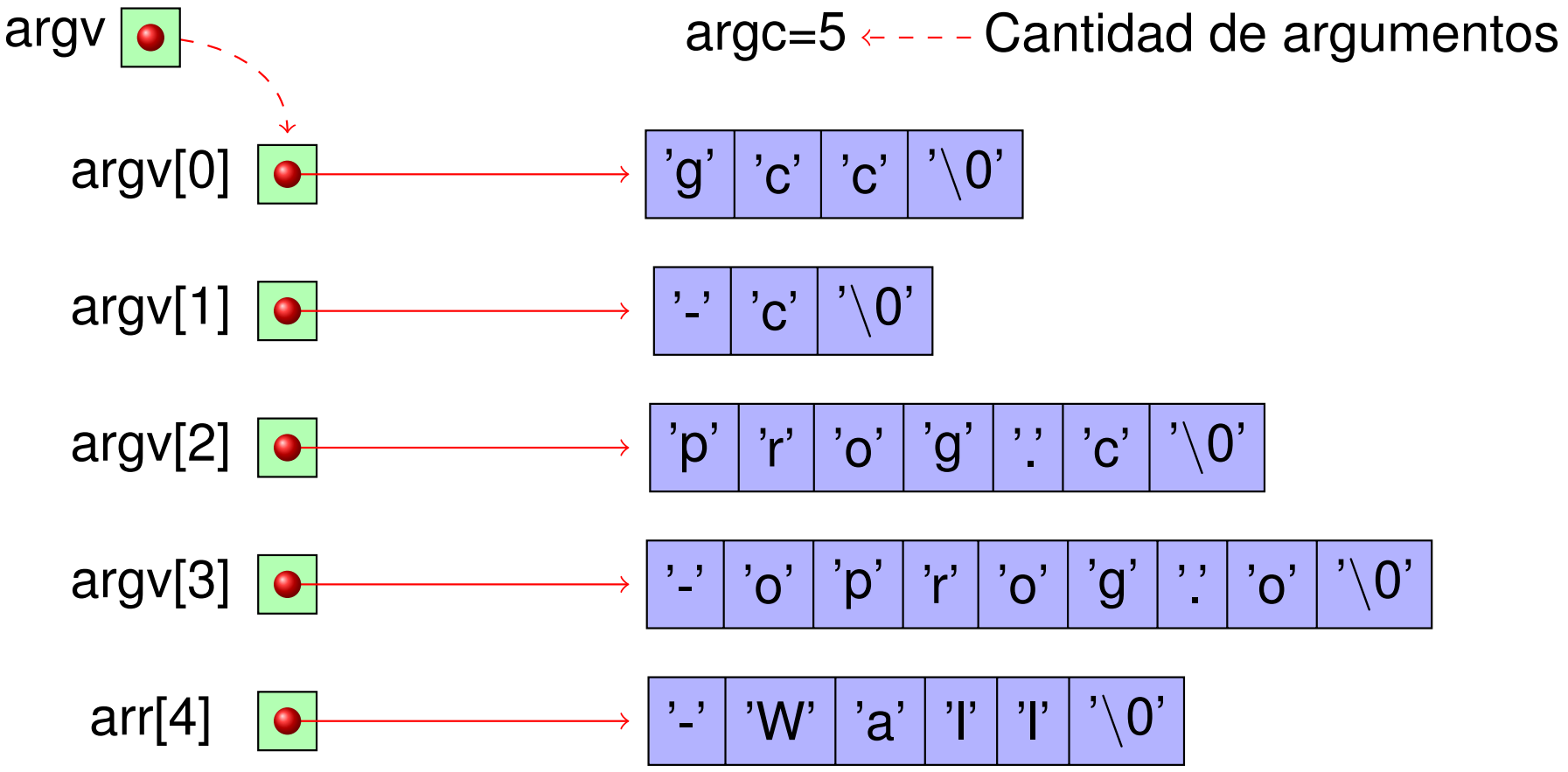
Genera esta disposición en memoria



main, la función multifacética

Ejemplo con función conocida *gcc -c prog.c -oprog.o -Wall*

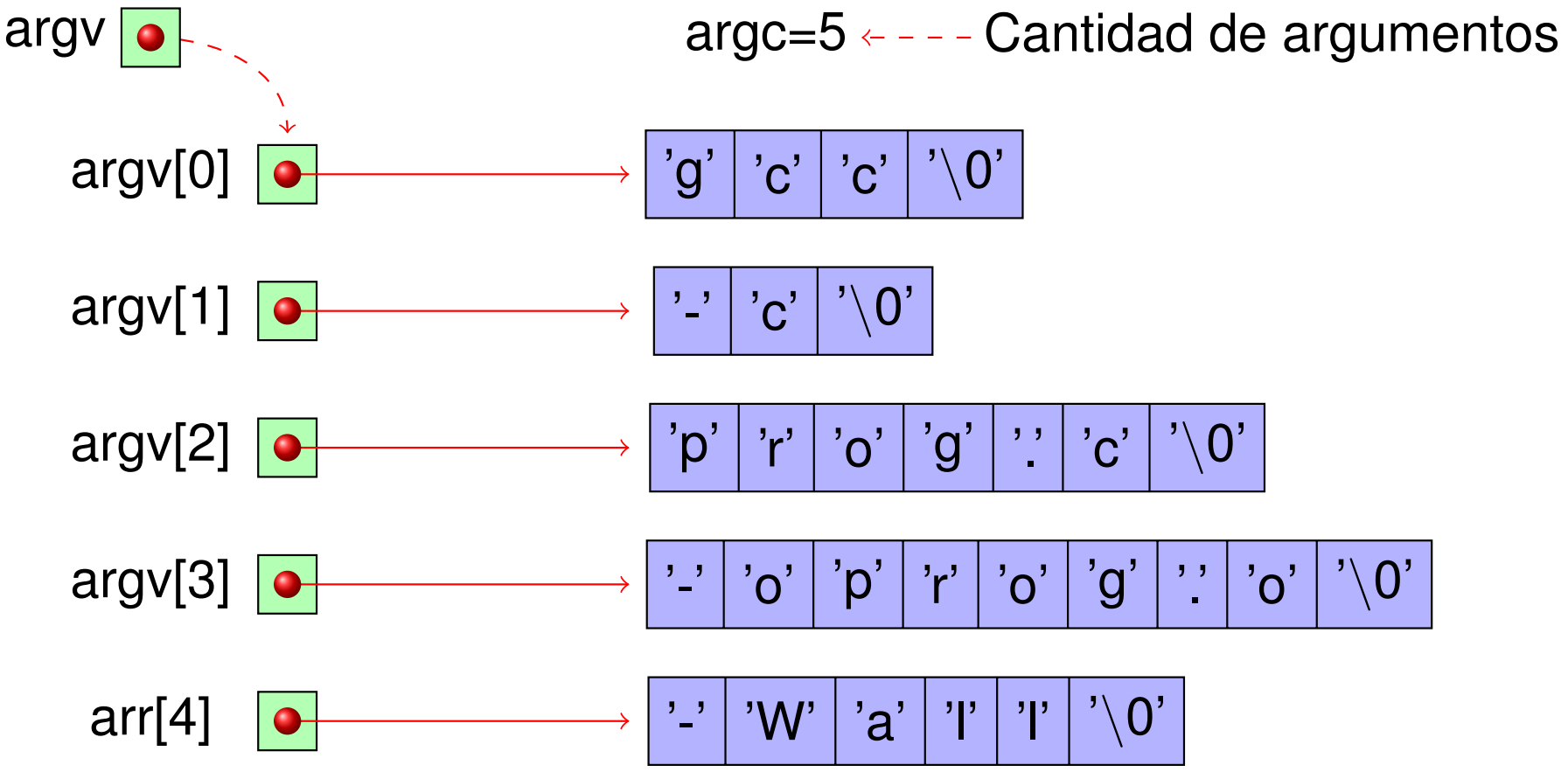
int main (int argc,char *argv[])



main, la función multifacética

Ejemplo con función conocida *gcc -c prog.c -oprog.o -Wall*

int main (int argc,char **argv)

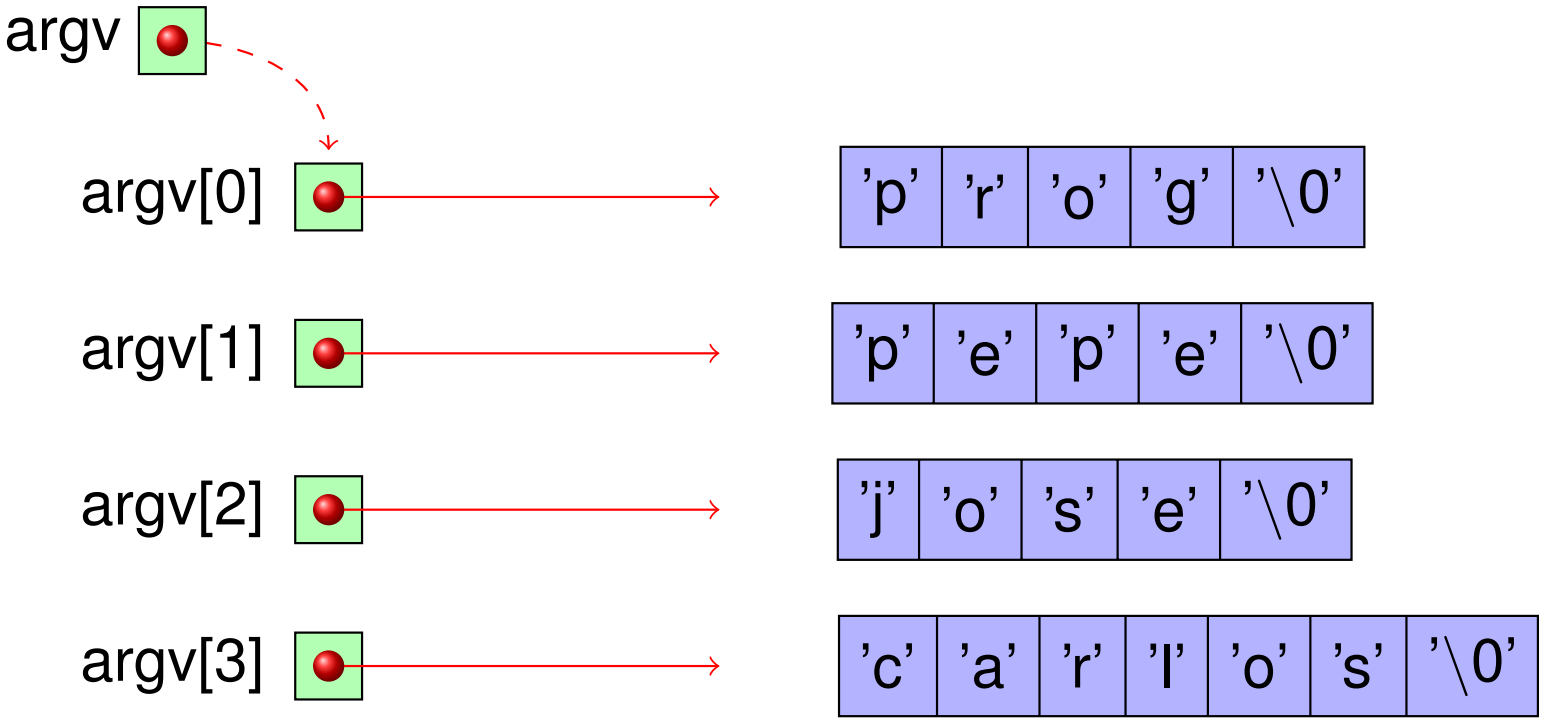


Ejemplo de *array de direcciones*

./prog *pepe jose carlos*

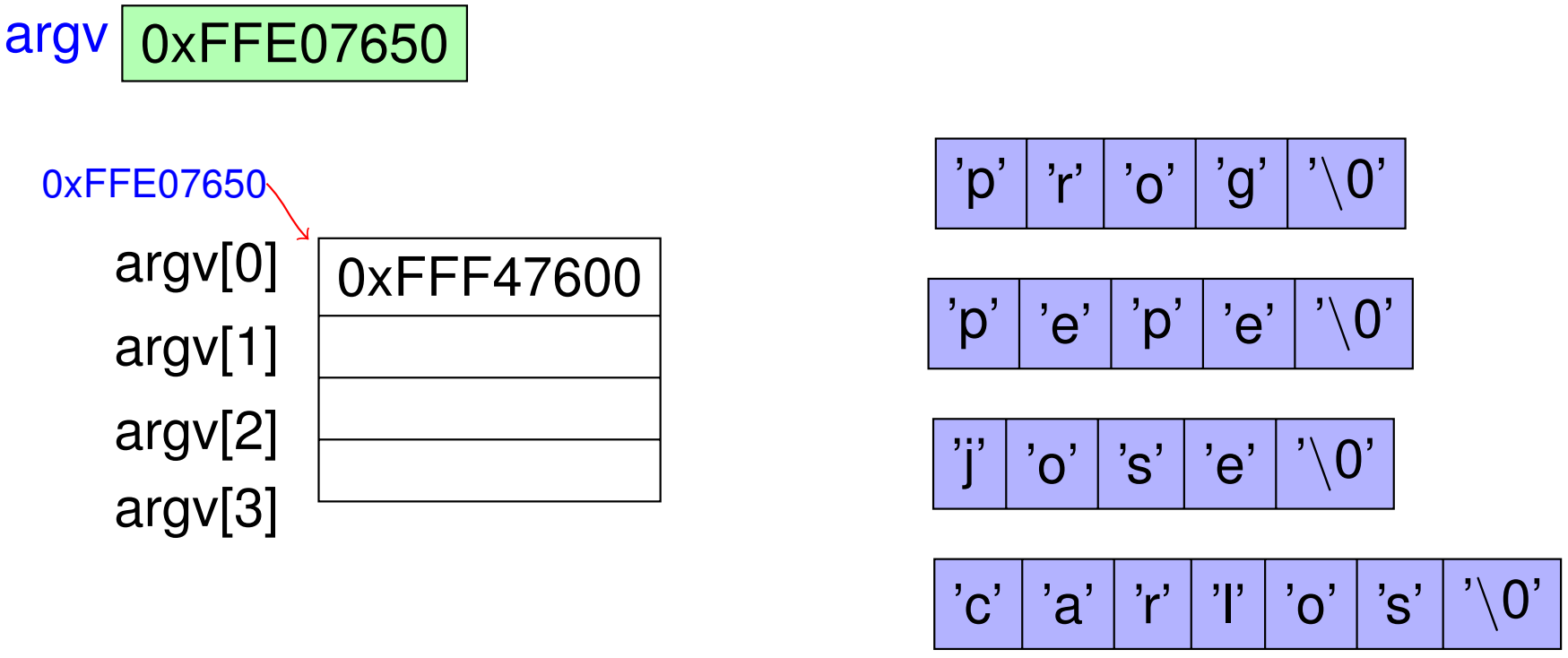
Los argumentos del programa son:

- pepe
- jose
- carlos



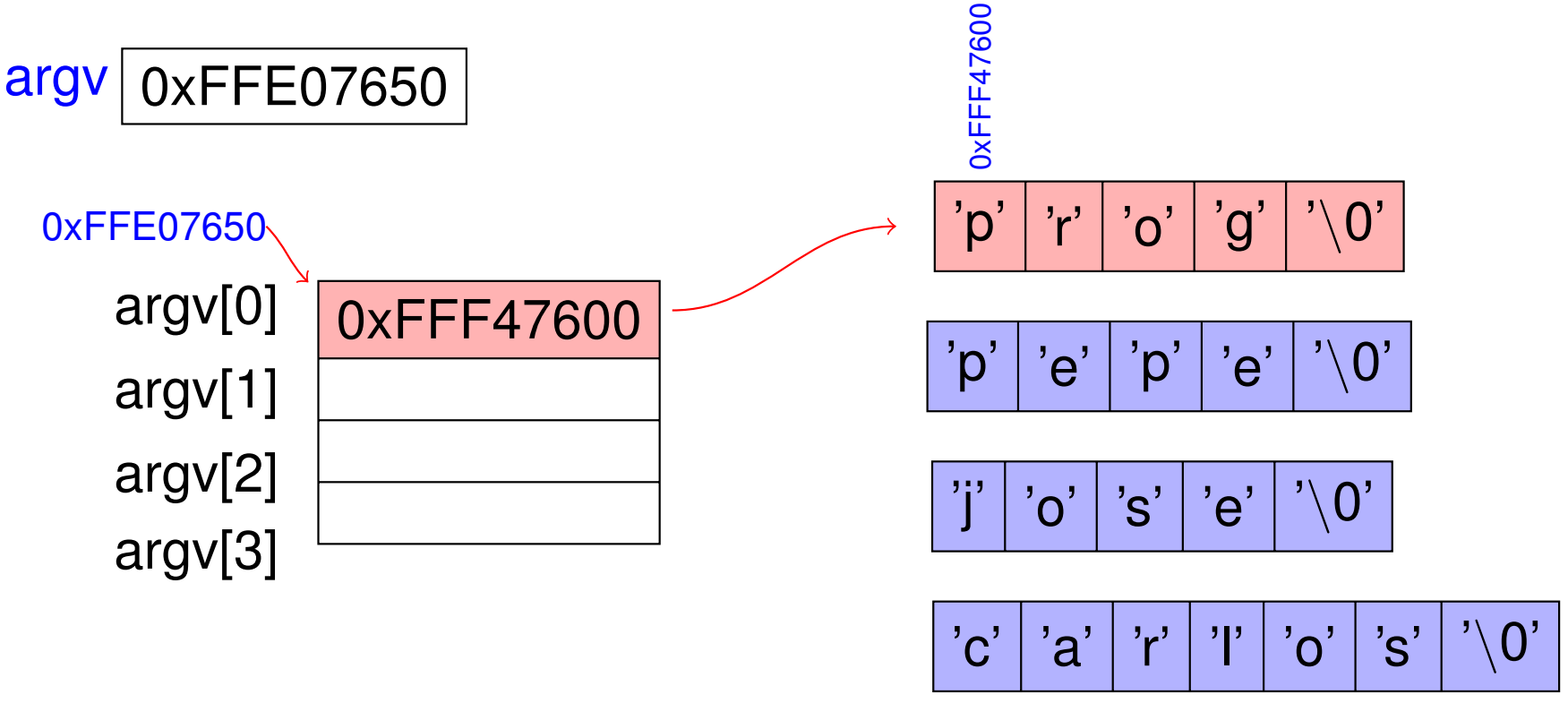
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



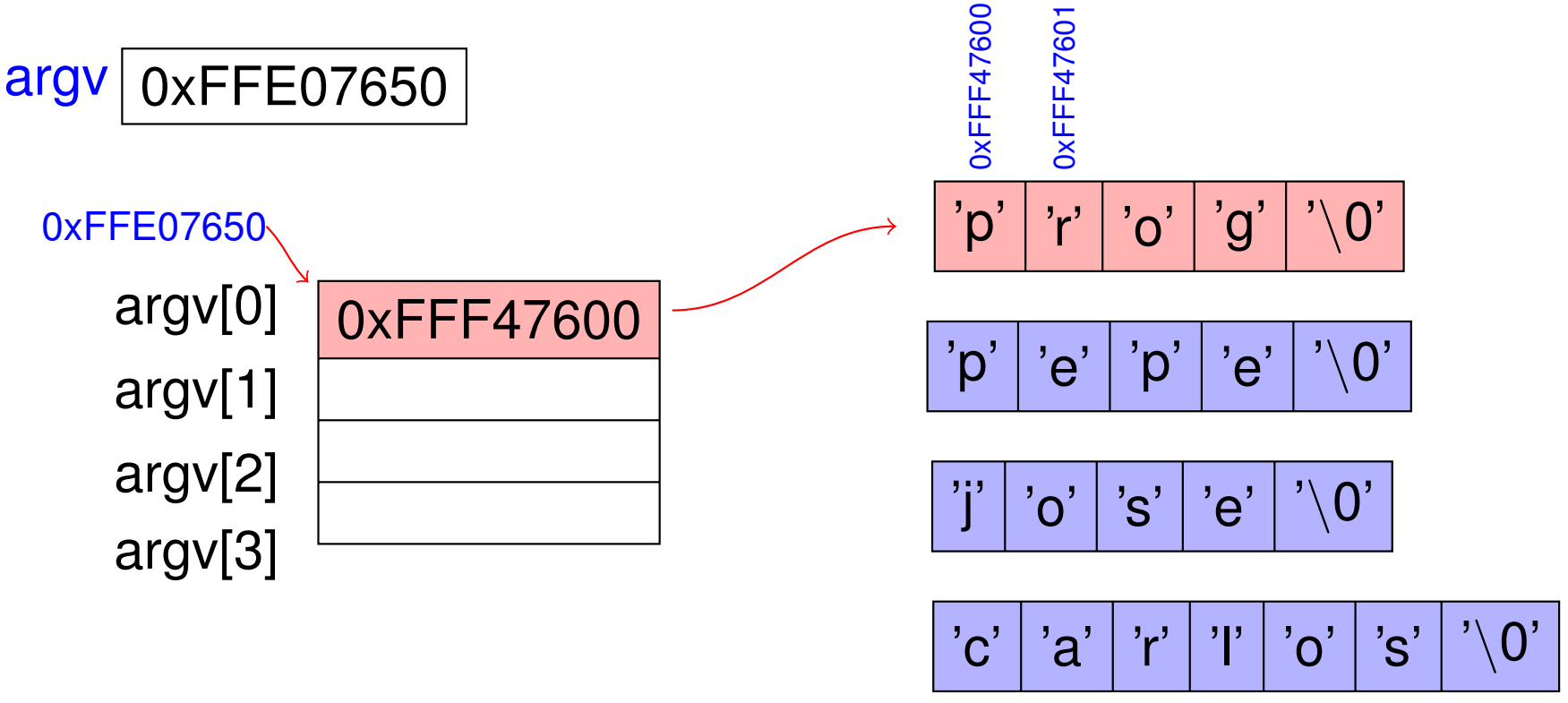
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



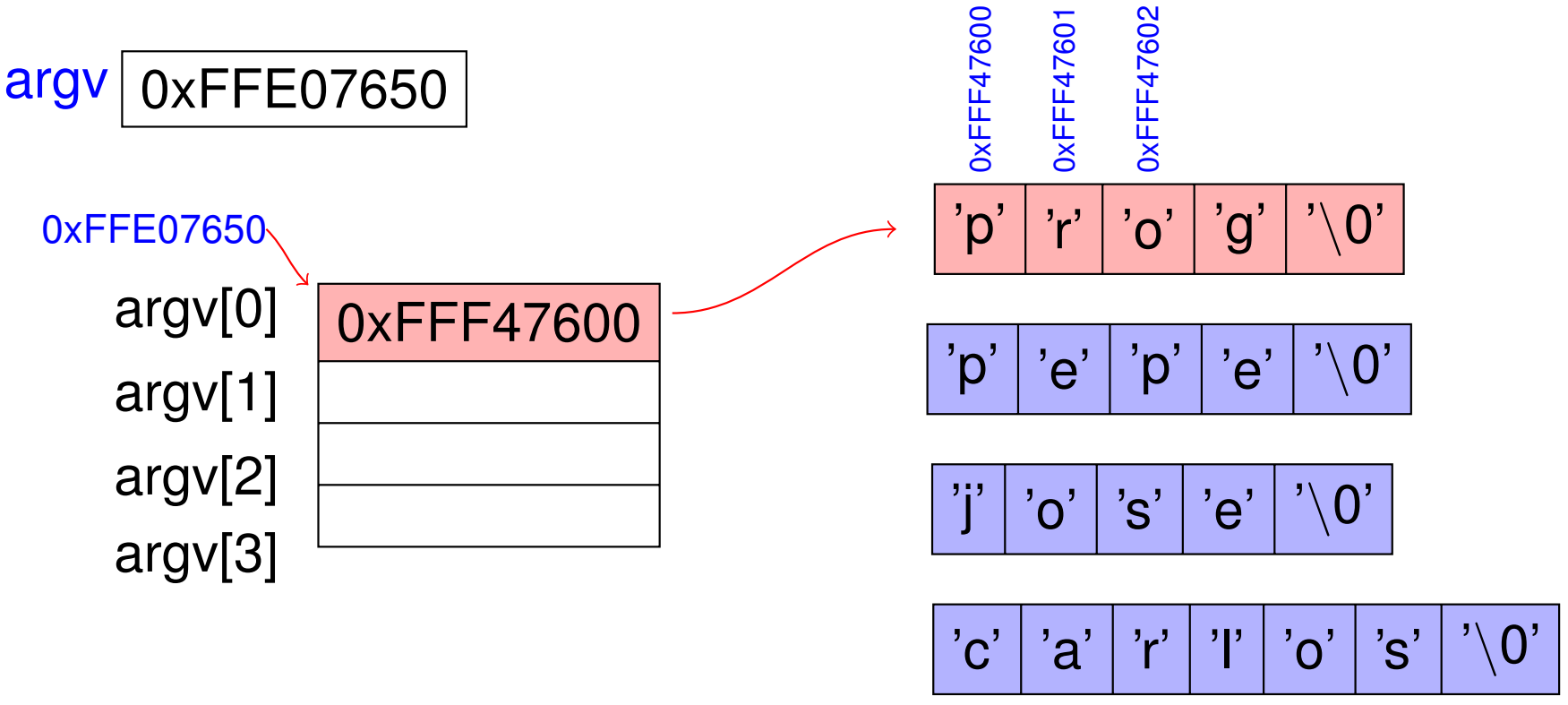
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



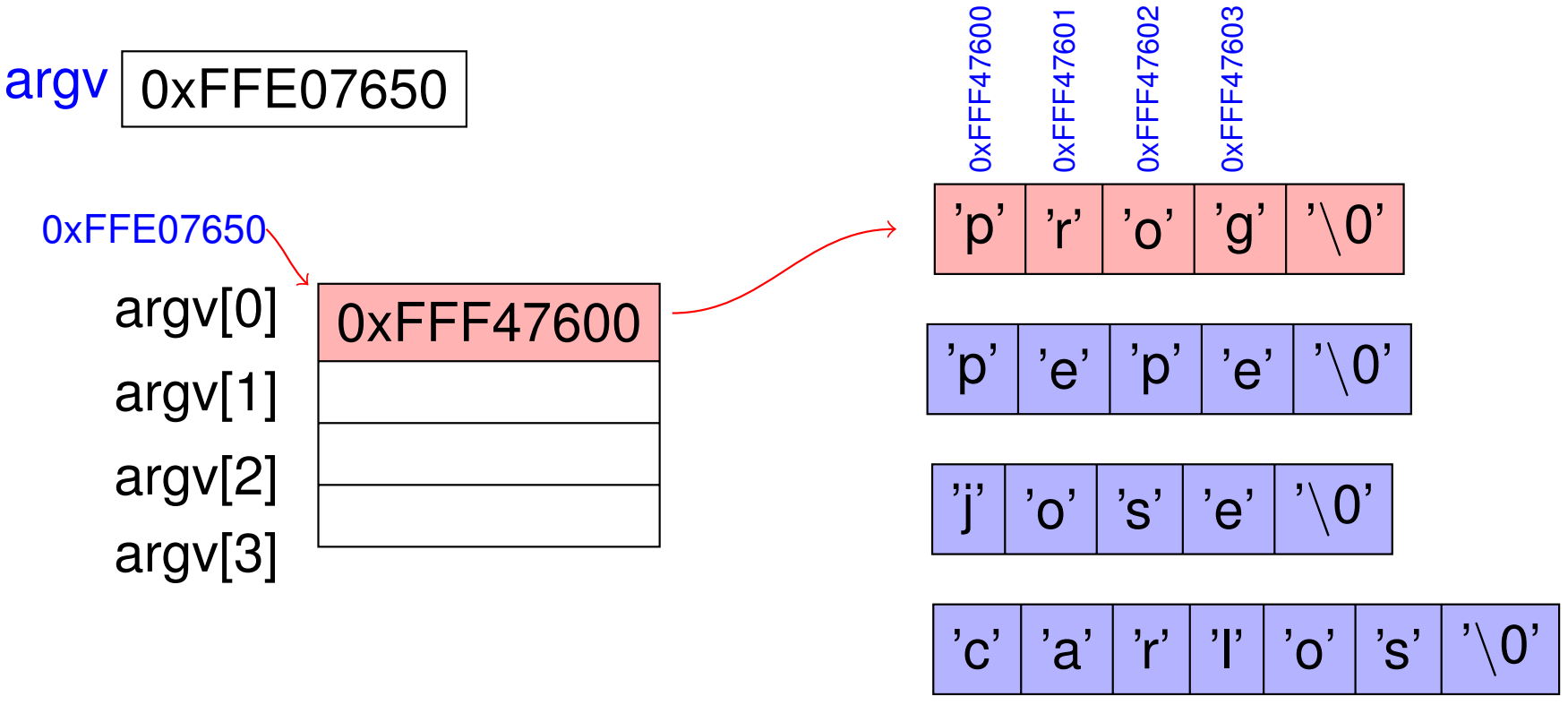
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



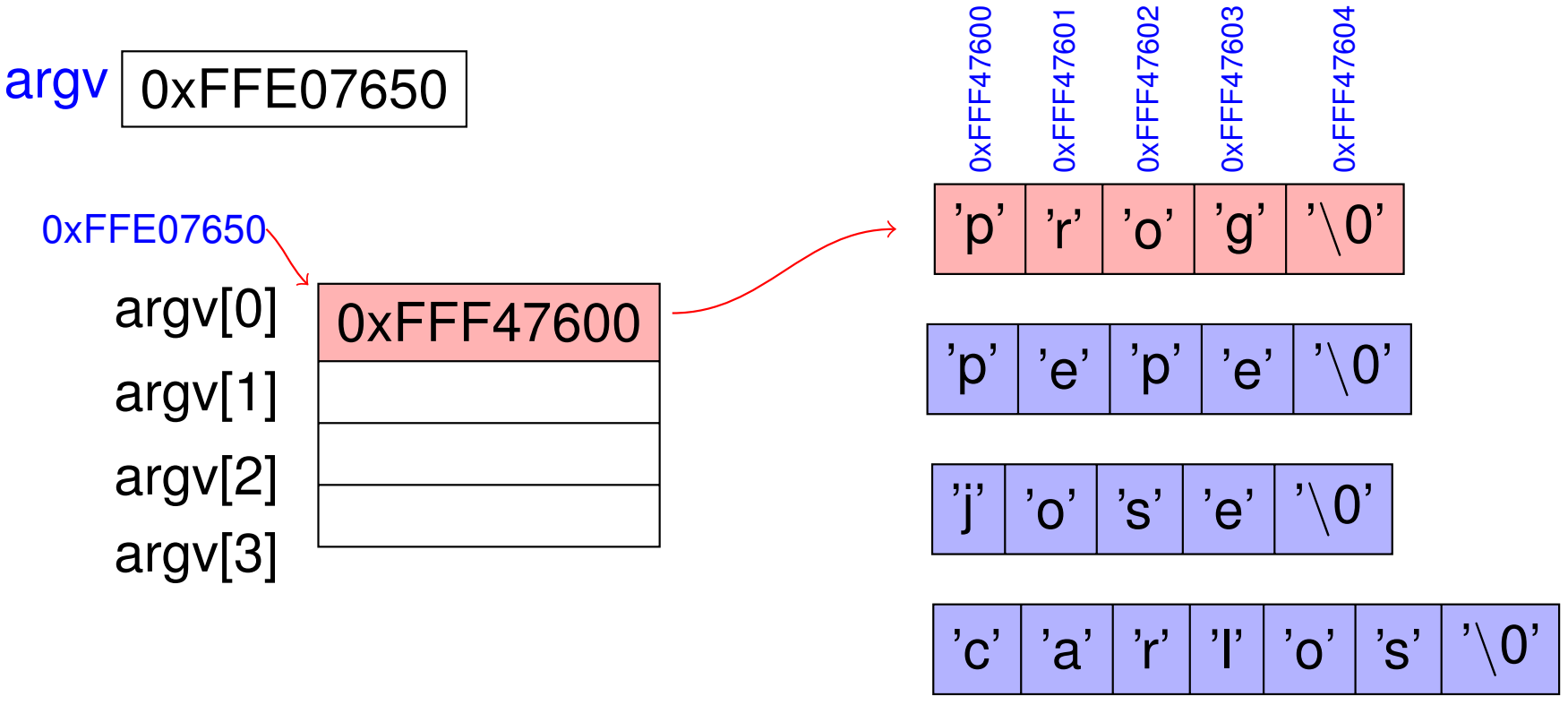
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



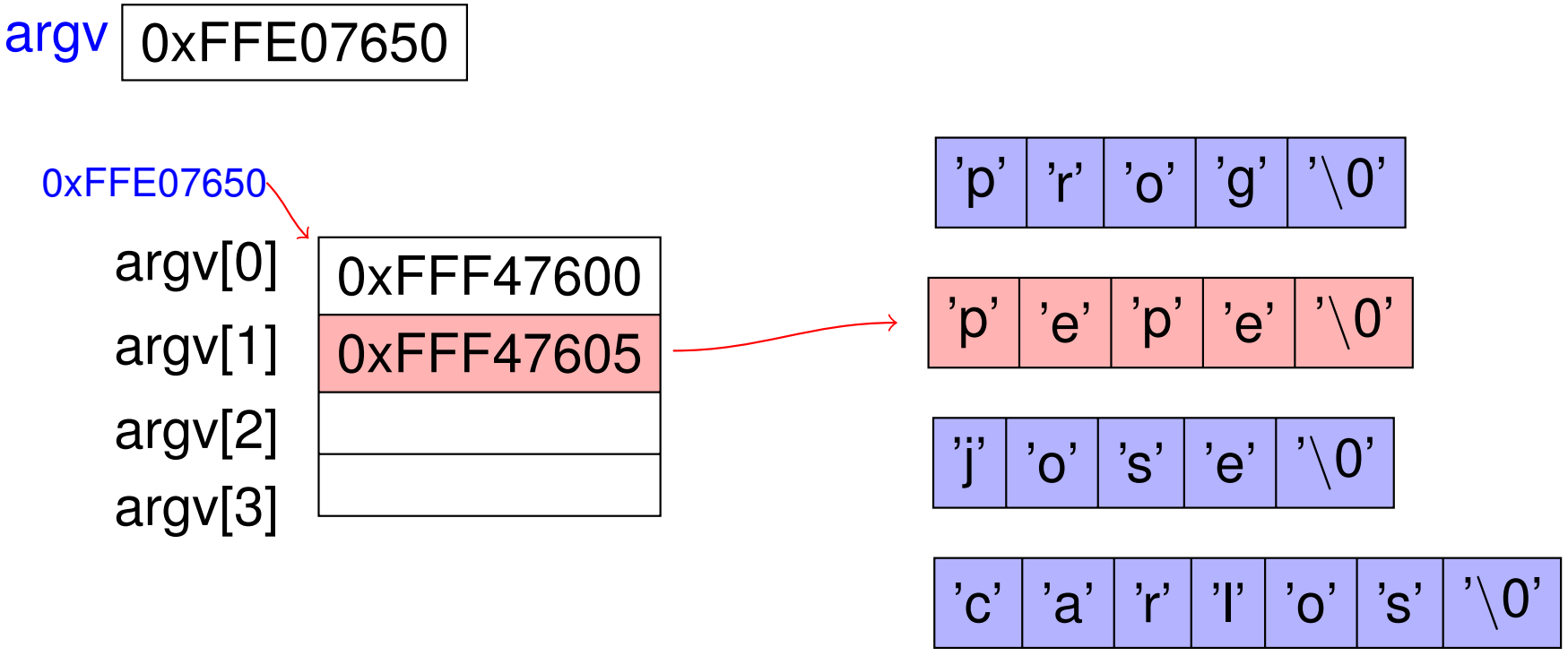
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



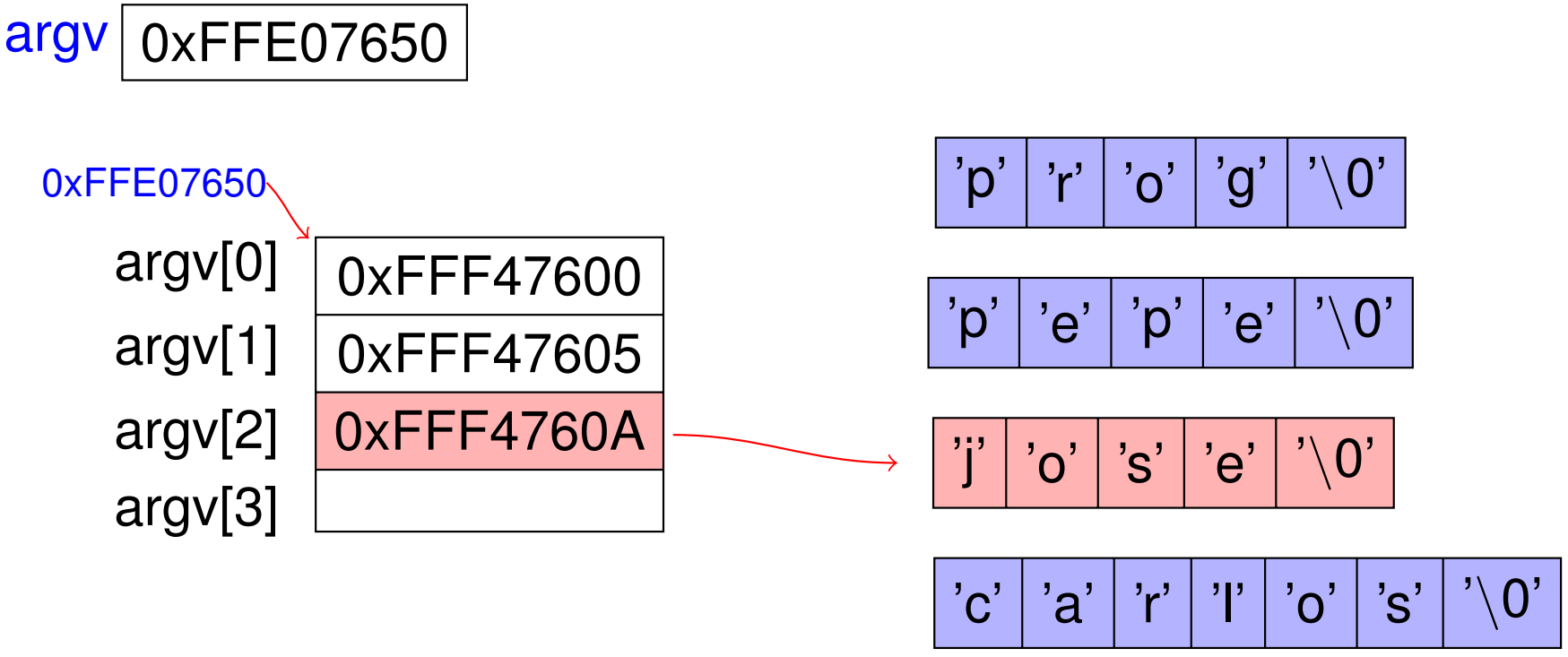
Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



Ejemplo de *array de direcciones*

Arquitectura X86-32 bits



Ejemplo de *array de direcciones*

Arquitectura X86-32 bits