



Conceptos de Algoritmos Datos y Programas



CADP – TEMAS

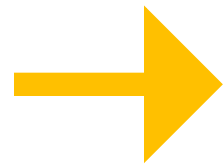


- Alocación de memoria
- Cálculo en la utilización de la memoria

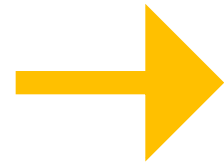
CADP – MEMORIA DE UN PROGRAMA



En rasgos generales la memoria necesaria para la ejecución de un programa puede dividirse en dos.

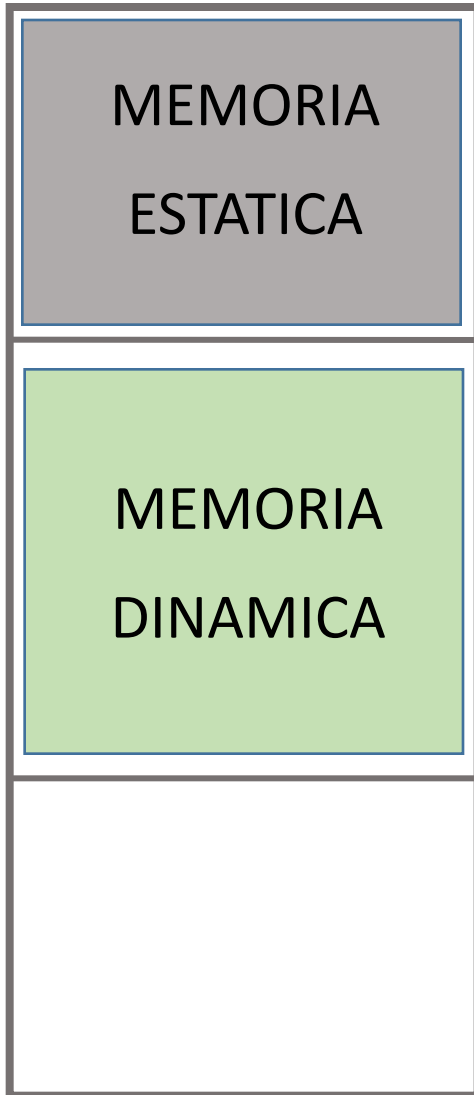


MEMORIA ESTATICA: a modo de simplicidad consideraremos sólo las variables locales, variables globales de programa y constantes.



MEMORIA DINAMICA: a modo de simplicidad consideraremos sólo cuando en la ejecución de un programa se reserva o libera memoria.

*Cómo
trabajamos?*



CADP – MEMORIA DE UN PROGRAMA



Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)

Program uno;

Const

DF = 10;

Type

puntero = ^real;

puntero2 = ^char;

persona = record

nombre:string[20];

dni:integer;

end;

puntPer = ^persona;

Var

p:puntero; q:puntero2;

per: puntPer;

perso:persona;

Begin

End.

CALCULO DE MEMORIA ESTATICA

DF = 4 bytes

p = 4 bytes

q = 4 bytes

per = 4 bytes

perso = 21 + 4 = 25 bytes

41 bytes

CALCULO DE MEMORIA DINAMICA

Como no hay operaciones de new()
ni dispose(), no se modifica la
memoria dinámica

0 bytes

CADP – MEMORIA DE UN PROGRAMA



Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)

```
Program uno;
```

```
Const
```

```
    DF = 10;
```

```
Type
```

```
    puntero = ^real;
```

```
    puntero2 = ^char;
```

```
    persona = record
```

```
        nombre:string[20];
```

```
        dni:integer;
```

```
    end;
```

```
    puntPer = ^persona;
```

```
Var
```

```
    p:puntero; q:puntero2;
```

```
    per: puntPer;
```

```
    perso:persona;
```

```
Begin
```

```
    new(p);
```

```
End.
```

CALCULO DE MEMORIA ESTATICA

DF = 4 bytes

p = 4 bytes

q = 4 bytes

per = 4 bytes

perso = 21 + 4 = 25 bytes

41 bytes

CALCULO DE MEMORIA DINAMICA

Al hacer un new se reserva
memoria para el contenido de p

8 bytes

CADP – MEMORIA DE UN PROGRAMA



```
Program uno;  
Const  
  DF = 10;  
Type  
  puntero = ^real;  
  puntero2 = ^char;  
  persona = record  
    nombre:string[20];  
    dni:integer;  
  end;  
  puntPer = ^persona;  
  
Var  
  p:puntero; q:puntero2;  
  per: puntPer;  
  perso:persona;  
Begin  
  new(p);  
  new(per);  
End.
```

Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)

CALCULO DE MEMORIA ESTATICA

DF = 4 bytes
p = 4 bytes
q = 4 bytes
per = 4 bytes
perso = 21 + 4 = 25 bytes

41 bytes

CALCULO DE MEMORIA DINAMICA

Al hacer un new se reserva
memoria para el contenido de p y
per (8 + 25)

33 bytes

CADP – MEMORIA DE UN PROGRAMA



```
Program uno;  
Const  
  DF = 10;  
Type  
  puntero = ^real;  
  puntero2 = ^char;  
  persona = record  
    nombre:string[20];  
    dni:integer;  
  end;  
  puntPer = ^persona;  
Var  
  p:puntero; q:puntero2;  
  per: puntPer;  
  perso:persona;  
Begin  
  new(p);  
  new(per);  
  read (per^.nombre);  
End.
```

Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)

CALCULO DE MEMORIA ESTATICA

DF = 4 bytes
p = 4 bytes
q = 4 bytes
per = 4 bytes
perso = 21 + 4 = 25 bytes

41 bytes

CALCULO DE MEMORIA DINAMICA

Al hacer un new se reserva memoria para el contenido de p y per (8 + 25), el read NO altera la memoria

33 bytes

CADP – MEMORIA DE UN PROGRAMA



```
Program uno;  
Const  
  DF = 10;  
Type  
  puntero = ^real;  
  puntero2 = ^char;  
  persona = record  
    nombre:string[20];  
    dni:integer;  
  end;  
  puntPer = ^persona;  
  
Var  
  p,p1:puntero; q:puntero2;  
  per: puntPer;  
  perso:persona;  
Begin  
  new(p);  
  p1:= p;  
End.
```

Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)

CALCULO DE MEMORIA ESTATICA

DF = 4 bytes
p,p1 = 8 bytes
q = 4 bytes **45 bytes**
per = 4 bytes
perso = 21 + 4 = 25 bytes

CALCULO DE MEMORIA DINAMICA

Al hacer un new se reserva memoria para el contenido de p, la asignación de direcciones NO altera la memoria **8 bytes**

CADP – MEMORIA DE UN PROGRAMA



Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)

```
Program uno;
```

```
Const
```

```
    DF = 10;
```

```
Type
```

```
    puntero = ^real;
```

```
    puntero2 = ^char;
```

```
    persona = record
```

```
        nombre:string[20];
```

```
        dni:integer;
```

```
    end;
```

```
    puntPer = ^persona;
```

```
Var
```

```
    p:puntero; q:puntero2;
```

```
    per: puntPer;
```

```
    perso:persona;
```

```
Begin
```

```
    new(p);
```

```
    dispose(p);
```

```
End.
```

CALCULO DE MEMORIA ESTATICA

DF = 4 bytes

p = 4 bytes

q = 4 bytes

per = 4 bytes

perso = 21 + 4 = 25 bytes

41 bytes

CALCULO DE MEMORIA DINAMICA

Al hacer un new se reserva memoria para el contenido de p, luego al hacer dispose(p) se libera la memoria **0 bytes**

CADP – MEMORIA DE UN PROGRAMA

```
Program uno;  
Const  
  DF = 10;  
Type  
  puntero = ^real;  
  puntero2 = ^char;  
  persona = record  
    nombre:string[20];  
    dni:integer;  
  end;  
  puntPer = ^persona;  
  
Var  
  p:puntero; q:puntero2;  
  per: puntPer;  
  perso:persona;  
Begin  
  new(p);  
  p:= nil;  
End.
```

Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)



CALCULO DE MEMORIA ESTATICA

DF = 4 bytes
p = 4 bytes
q = 4 bytes
per = 4 bytes
perso = 21 + 4 = 25 bytes

41 bytes

CALCULO DE MEMORIA DINAMICA

Al hacer un new se reserva memoria para el contenido de p, luego al hacer nil NO se libera la memoria

8 bytes

CADP – MEMORIA DE UN PROGRAMA

Program uno;

Type

```
puntero = ^real;  
puntero2 = ^char;  
persona = record  
  nombre:string[20];  
  dni:integer;  
end;
```

```
punPer = ^persona;
```

Var

```
p,p1:puntero;  
per: punPer;
```

Begin

```
  new(per);  
  new(p);  
  p^:= 8;  
  p1:= p;  
  dispose(p1);
```

End.

Tabla de ocupación:

char,	(1 byte)
boolean,	(1 byte)
integer	(4 bytes)
real,	(8 bytes)
string,	(tamaño + 1 byte)
subrango,	(depende el tipo)
registro,	(suma de sus campos)
arreglos	(dimFísica*tipo elemento)
puntero	(4 bytes)



CALCULO DE MEMORIA ESTATICA

p = 4 bytes	
p1 = 4 bytes	12 bytes
per = 4 bytes	

CALCULO DE MEMORIA DINAMICA

new(per) = 25 bytes	
new(p) = 8 bytes	25 bytes
dispose(p1) = -8 bytes	