

PUNTEROS DOBLES EN C

XIII CUIEET – MASPALOMAS

Sidonio Pérez Matilla
EUITI de Eibar
UPV

PUNTEROS DOBLES EN C

- INTRODUCCIÓN.
- DEFINICIONES DE PUNTERO y PUNTERO DOBLE.
- MAPA DE MEMORIA DE UN PROGRAMA CON PUNTEROS.
- FUNCIONES DE USUARIO Y PUNTEROS.
- EJEMPLO 1.
 - CÓDIGO FUENTE.
 - RESULTADOS.
 - MAPA DE MEMORIA.
- EJEMPLO 2.
 - RESULTADOS.

PUNTEROS DOBLES EN C

INTRODUCCIÓN

Un ejemplo sencillo de un concepto del lenguaje C difícil de entender:

LOS PUNTEROS DOBLES

PUNTEROS DOBLES EN C

DEFINICIÓN DE PUNTERO

Variable cuyos posibles valores son direcciones de otras variables.

DEFINICIÓN DE PUNTERO DOBLE

Variable cuyos posibles valores son direcciones de otras variables puntero.

```
int x=5, y[50], *p1=&y[0], **p2=&p1;
```

PUNTEROS DOBLES EN C

MAPA DE MEMORIA DE UN PROGRAMA CON PUNTEROS

Una tabla donde se relacionan identificadores, direcciones y contenidos

```
int x=5, y[50], *p1=&y[0], **p2=&p1;
```

PUNTEROS DOBLES EN C

DIRECCION	CONTENIDO	IDENTIFICADOR
...
65000	5	x
65002		Y[0]
...
65100		Y[49]
65102	65002	p1
65104	65102	p2
...

PUNTEROS DOBLES EN C

FUNCIONES DE USUARIO Y PUNTEROS

Las funciones de usuario pueden usar punteros:

- En el valor devuelto; en los PF; en variables locales o globales:

```
int *f1(int *A, int B, int C)    // interface
{
    int *p1, ...;
    ...
    return (p1);
}
```

Analiza una tabla de enteros de BxC siendo A su dirección de comienzo y devuelve la dirección del máximo.

PUNTEROS DOBLES EN C

FUNCIONES DE USUARIO Y PUNTEROS

Las funciones que devuelven varios valores usan PF por referencia para “devolver” los resultados que no pueden ser devueltos con un return:

```
float f2(int *A, int B, int C, int *D, int *E) // D y E: dir de var int
{
    int *p1, *p2, f, c, acu;
    ...
    *D = *p1;           // un int
    *E = *p2;           // un int
    return ((float)acu/B*C); // un float
}
```

Analiza una tabla de enteros de BxC siendo A su dirección de comienzo y devuelve el Vmed, y el máximo y mínimo a través de D y E.

PUNTEROS DOBLES EN C

FUNCIONES DE USUARIO Y PUNTEROS

Cuando una función de usuario ha de devolver direcciones a través de PF pasados por referencia estos se definen como punteros dobles:

```
int *f3(int *A, int B, int C, int **D) // D: dir de una var *int
{
    int *p1, ...;
    ...
    *D = expresión;    // una dirección
    return (p1);
}
```

Analiza una tabla de enteros de BxC siendo A su dirección de comienzo y devuelve la dirección del máximo, y también la dirección del mínimo a través de D.

PUNTEROS DOBLES EN C

EJEMPLO 1

Escribir un programa en C que intercambie entre sí las vocales primeras y últimas de dos cadenas.

Hola, Adios --> HAlo, odias

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
#include <stdio.h>
#include <conio.h>
int main()
{
    /* ..... PARTE DECLARATIVA ..... */
    char s1[80], s2[80], *p11, *p12, *p21, *p22;
    char *f1(char *, char **);           // prototipo
    void f2(char *, char *);             // prototipo
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
/* ..... PARTE EJECUTIVA ..... */  
clrscr();           // limpia pantalla usuario  
  
// 1: Leer dos cadenas desde el teclado:  
printf("\n\tTeclea una cadena: ");   gets(s1);  
printf("\n\tTeclea otra cadena: ");  gets(s2);
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
// 2: visualizar las cadenas tecleadas:  
puts("\n\n\tCadenas iniciales: ");  
printf("\ts1: %s \n\tts2: %s", s1, s2);
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

// 3: Buscar direcciones de las vocales
primera y última de cada cadena:

```
p11 = f1(s1, &p12); // llamada función
```

```
p21 = f1(s2, &p22); // llamada función
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
// 4: Intercambiar primeras vocales s1-s2:  
f2(p11, p21);    // llamada a función
```

```
// 5: Intercambiar últimas vocales s1-s2:  
f2(p12, p22);    // llamada a función
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
// 6: visualizar las cadenas tratadas:  
puts("\n\n\tCadenas finales: ");  
printf("\ts1: %s \n\tts2: %s", s1, s2);
```


PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
printf("\n\tPulsa tecla para terminar ... ");  
getch();           // pantalla usuario  
return 0;  
}                  // del main()
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
/*-----  
Analiza cadena y devuelve dirs primera y última vocal o del  
\0 si no hay vocales:  
DEVUELVE: dirección de la primera vocal o del \0.  
A: dirección de comienzo de la cadena.  
B: dir *char para devolver dir última vocal o del \0  
-----*/  
char *f1(char *A, char **B) // interface  
{ // código  
    char *p1, *p1voc, *puvoc;
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
// búsqueda de la primera vocal o del \0
for (p1=A; *p1 != '\0'; p1++)
{
    if (*p1=='a' || *p1=='e' || *p1=='i' || *p1=='o' || *p1=='u'
        || *p1=='A' || *p1=='E' || *p1=='I' || *p1=='O' || *p1=='U')
    {
        break;
    }
}
p1voc = p1; // dir primera vocal o \0
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
// búsqueda de la última vocal o del \0
for (p1=A; *p1 != '\0'; p1++)
{
    ;
}
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
for (p1--; p1 >= A; p1--)
{
    if (*p1=='a' || *p1=='e' || *p1=='i' || *p1=='o' || *p1=='u'
        || *p1=='A' || *p1=='E' || *p1=='I' || *p1=='O' || *p1=='U')
    {
        break;
    }
}
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
if (*p1voc == '\0')
{
    puvoc = p1voc;           // A sin vocales
}
else
{
    puvoc = p1;             // A con vocal/es
}
```

PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
*B = puvoc; // Devolver dir ult vocal  
return (p1voc); // Devolver dir pri vocal  
}
```

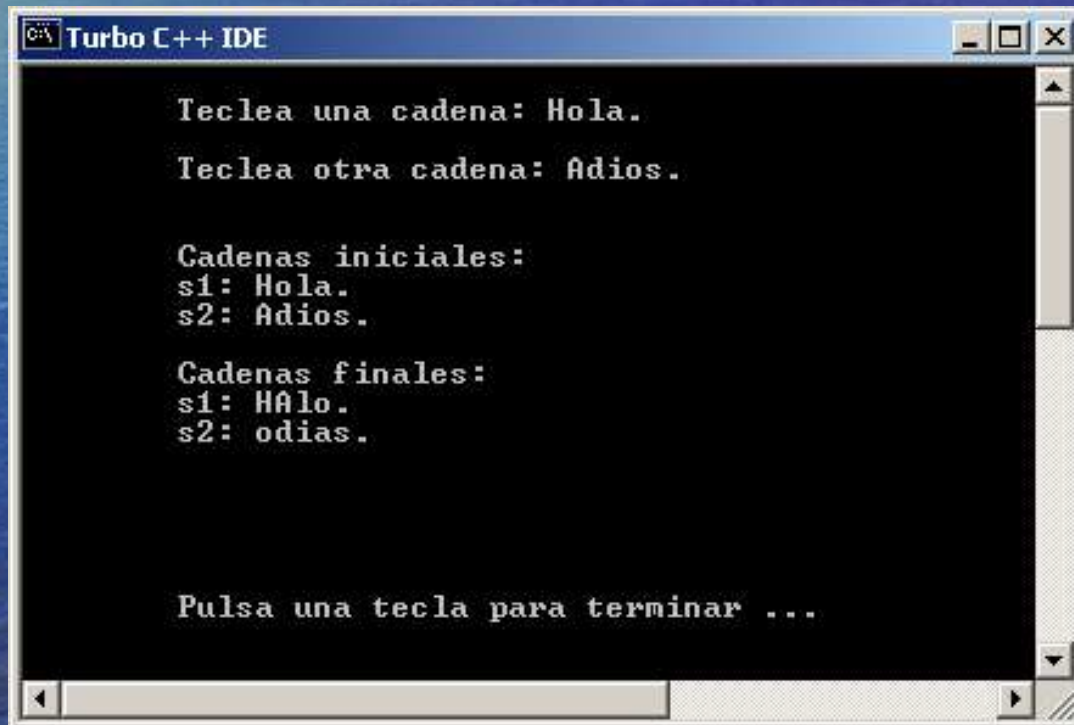
PUNTEROS DOBLES EN C

CÓDIGO FUENTE DEL EJEMPLO 1

```
/* Intercambia 2 char's dados por sus dirs A y B */  
void f2(char *A, char *B)    // interface  
{                            // código  
    char aux;  
    aux = *A;    *A = *B;    *B = aux;  
}
```


PUNTEROS DOBLES EN C

RESULTADOS DEL EJEMPLO 1



The screenshot shows a Turbo C++ IDE window with a black background and white text. The text is as follows:

```
Turbo C++ IDE

Tecllea una cadena: Hola.
Tecllea otra cadena: Adios.

Cadenas iniciales:
s1: Hola.
s2: Adios.

Cadenas finales:
s1: HAllo.
s2: odias.

Pulsa una tecla para terminar ...
```

PUNTEROS DOBLES EN C

RESULTADOS DEL EJEMPLO 1

Combinaciones de vocales en s1 y s2:

0: sin vocales. 1: una vocal. 2+: dos o mas vocales.

0 0	0 1	0 2+
1 0	1 1	1 2+
2+ 0	2+ 1	2+ 2+

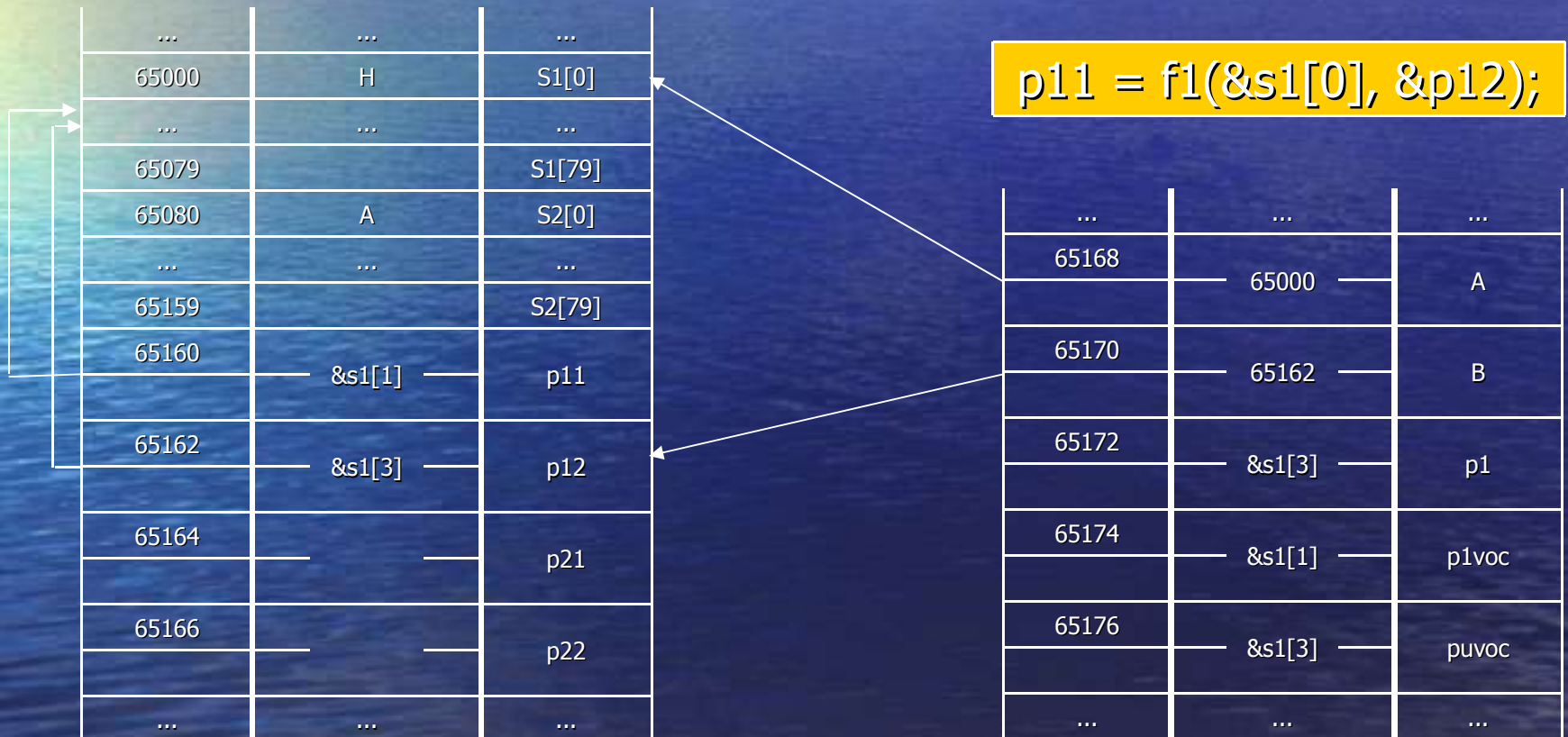
PUNTEROS DOBLES EN C

RESULTADOS DEL EJEMPLO 1

s1 y s2	f1(...)	f2(...)	s2 y s2
Hola\0...	p11=&s1[1] -> 'o' p12=&s1[3] -> 'a'	p11=&s1[1] -> 'A' p21=&s2[0] -> 'o'	HAlo\0...
Adios\0...	p21=&s2[0] -> 'A' p22=&s2[3] -> 'o'	p12=&s1[3] -> 'o' p22=&s2[3] -> 'a'	odias\0...

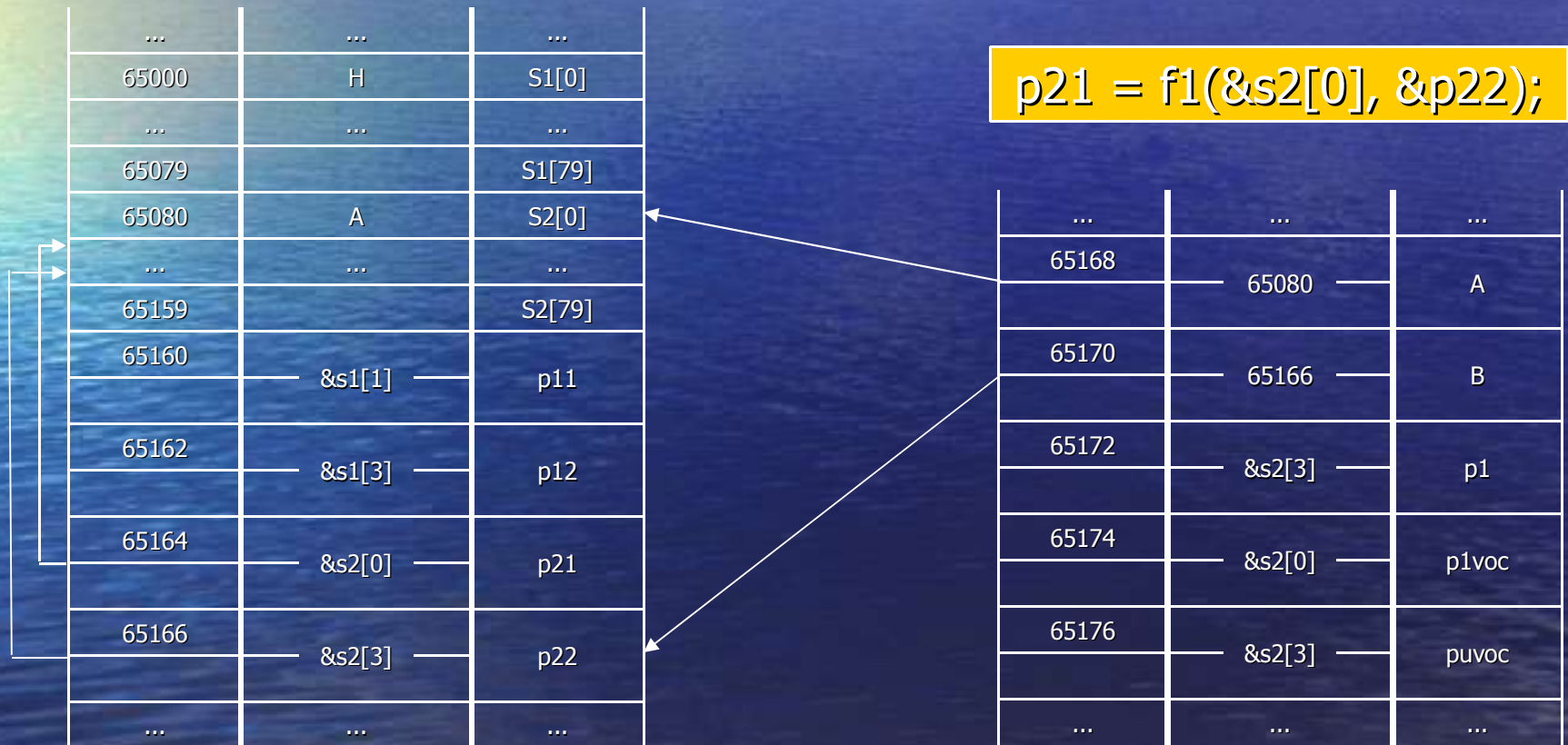
PUNTEROS DOBLES EN C

MAPA DE MEMORIA DEL EJEMPLO 1



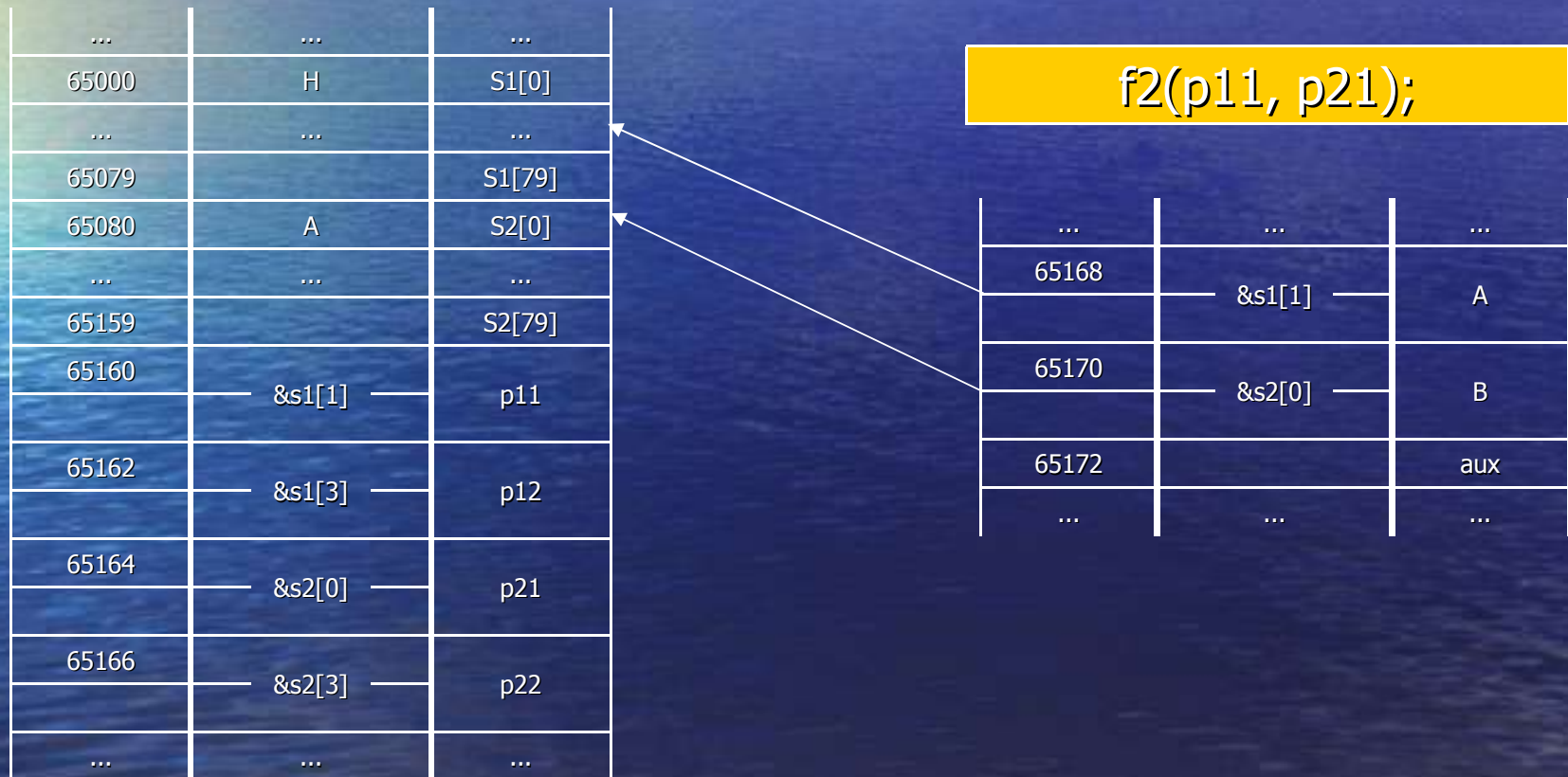
PUNTEROS DOBLES EN C

MAPA DE MEMORIA DEL EJEMPLO 1



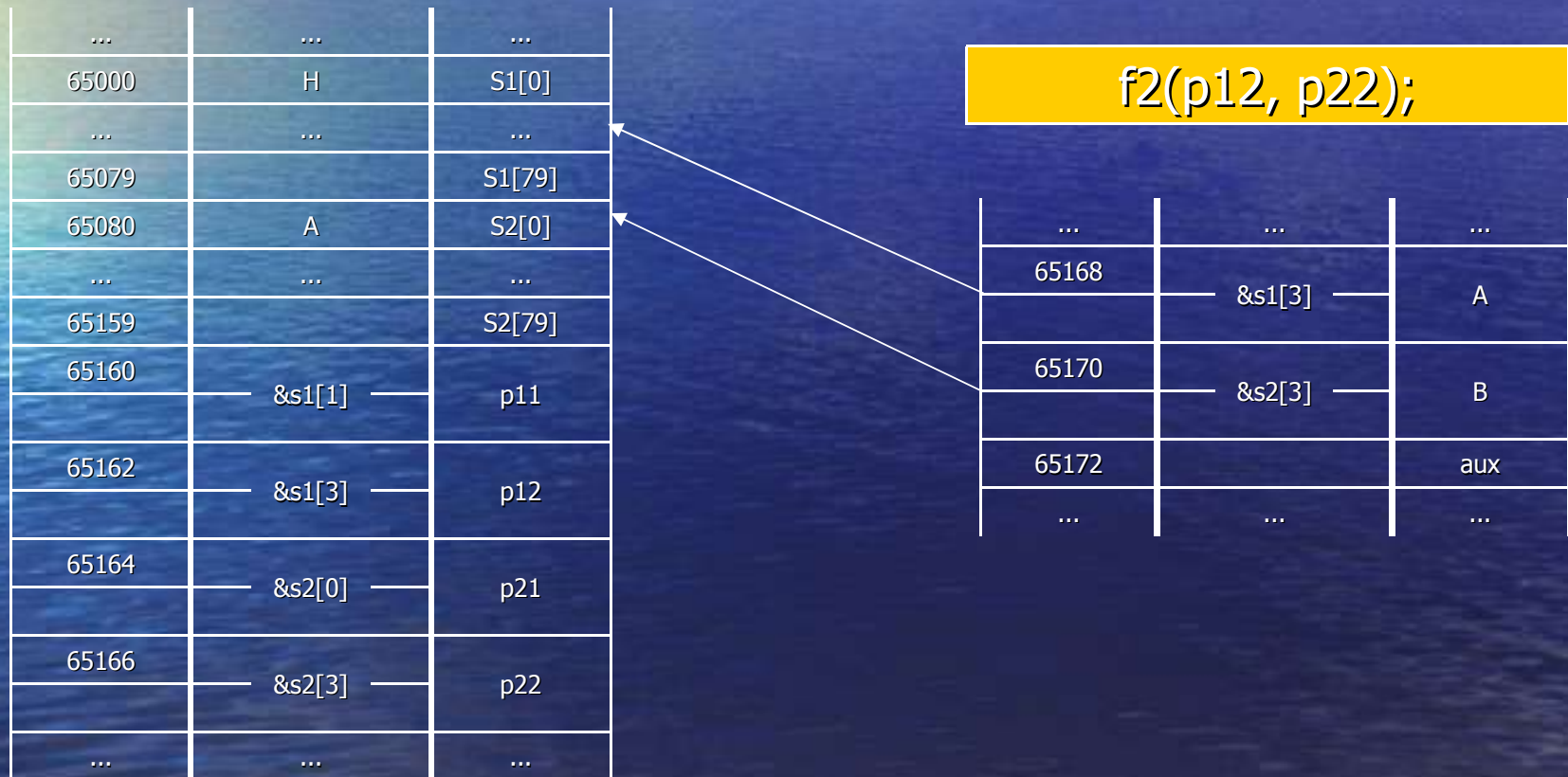
PUNTEROS DOBLES EN C

MAPA DE MEMORIA DEL EJEMPLO 1



PUNTEROS DOBLES EN C

MAPA DE MEMORIA DEL EJEMPLO 1



PUNTEROS DOBLES EN C

EJEMPLO 2:

Escribir un programa en C que permita intercambiar entre si los valores máximos y mínimos de dos tablas de enteros.

PUNTEROS DOBLES EN C

RESULTADOS DEL EJEMPLO 2

```
Turbo C++ IDE
Contenido inicial de las tablas:
Tabla A:
1      2      3      4
3      5      466   -6
7      8      9      0
7      6      5      4

Fin ... pulsa S para seguir ...
Tabla B:
333    4      5      -66
4      3      2      5

Fin ... pulsa S para seguir ... _
```

```
Turbo C++ IDE
Contenido final de las tablas:
Tabla A:
1      2      3      4
3      5      333   -66
7      8      9      0
7      6      5      4

Fin ... pulsa S para seguir ...
Tabla B:
466    4      5      -6
4      3      2      5

Fin ... pulsa S para seguir ...
```

**MUCHAS
GRACIAS
POR
SU
ATENCIÓN**