

# CC41B Auxiliar 01

Profesor: L. Mateu  
Auxiliar: M. Leyton

9 de agosto de 2004

## 1. Diagnóstico C

### 1.1. Elección Aleatoria

El siguiente programa permite elegir uno de dos enteros pseudo-aleatoriamente.

```
int randomChoose(int a, int b){  
    if(time(0)%2) return a;  
    return b;  
}
```

```
main(){  
    int a=1,b=2;  
    int c=randomChoose(a,b);  
}
```

Modifique este programa para que acepte cualquier estructura de C.

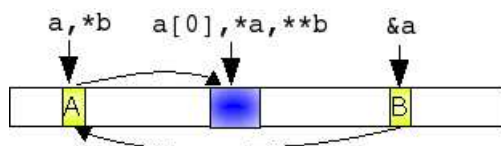
### 1.2. Swap

El siguiente código para hacer swap no funciona. Explique por qué razón, e implemente una versión correcta.

```
void swapMalo(int a, int b){  
    int tmp=a;  
    a=b;  
    b=tmp;  
}
```

## 2. Repaso C

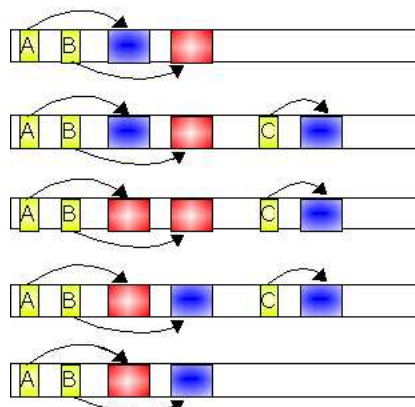
### 2.1. Punteros



Sopa de Punteros

La siguiente figura muestra los estados de punteros al utilizar una función de swap:

Swap Contenido



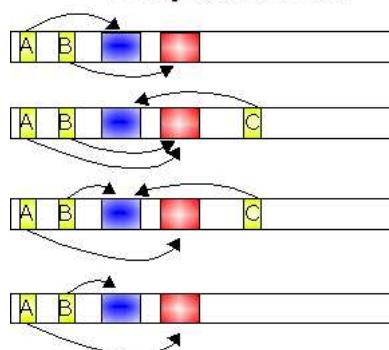
Un intento fallido de implementar este swap resultó en:

```
void swapGenericoMalo(void *a, void *b){  
    void *tmp=a;  
    a=b;  
    b=tmp;  
}
```

¿Por qué razón no funciona? Corrija este swap genérico, de tal manera que el método de swap opere con cualquier estructura de C.

Comente porque no es posible implementar una función que realice swap de referencias. ¿Que alternativa propone?

Swap Referencias



## 2.2. Estructuras

Las estructuras en C puede ser definidas de varias maneras. La tradicional consiste en crear una estructura y definirla como un tipo:

```
typedef struct Coord{
    int x,y,z;
} Coord;
```

Para utilizar estas estructuras se debe crear una variable asociada. Por ejemplo, para realizar un swap de estas variables se debe:

```
Coord coordA,coordB;
swapGenerico((void *)&coordA,
             (void *)&coordB, sizeof(Coord));
```

Escriba la llamada a la función de swap genérico, utilizando las siguientes declaraciones de variables:

```
Coord *coordC=
    (Coord *)malloc(sizeof(Coord));
Coord coordD[1];
```

## 3. Soluciones

### 3.1. Diagnóstico

```
int randomChoose(int a, int b){
    if(time(0)%2)
        return a;
    return b;
}

void * genericRandomChoose(void *a,
                           void *b){
    if(time(0)%2)
        return a;
    else
        return b;
}

main(){
    int a=1;
    int b=2;

    /* Alternativas
    void *c=genericRandomChoose(&a,&b);
    printf("selected:%d\n", *(int *)c);

    int *c=
        (int *)genericRandomChoose(&a,&b);
    printf("selected:%d\n", *c);

    */

    int c=
        *(int *)genericRandomChoose(&a,&b);
    printf("selected:%d\n", c);
}
```

### 3.2. Problemas

```
#define swapReferencias(x, y) { int tmp = x; x = y; y = tmp }

void swapMalo(int a, int b){
    int tmp=a; a=b; b=tmp;
}

void swapInt(int *a, int *b){
    int tmp=*a; *a=*b; *b=tmp;
}

void swapGenericoMalo(void *a, void *b){
    void *tmp=a;
    a=b;
    b=tmp;
}

void * my_memcpy(void *dest, void*src, int size){

    char *dest_char= (char *)dest;
    char *src_char= (char *)src;

    while(size--){
        *dest_char++=*src_char++;
    }

    return dest;
}

void swapGenerico(void *a, void *b, int size){

    char tmp[size];
    //tmp[0]=*(char *)a;
    //tmp[1]=*(char *)a+1;
    //...
    //tmp[size-1]=*(char *)a+size-1;

    //mejor utilizar memcpy
    my_memcpy(tmp,a,size);
    my_memcpy(a,b,size);
    my_memcpy(b,tmp, size);
}

typedef struct Coord{
    int x,y,z;
} Coord;

main(){
    int a=a, b=b;
    printf("%c %c\n",a,b);
    swapInt(&a,&b);
    printf("%c %c\n",a,b);

    Coord coordA,coordB;
    swapGenerico((void *)&coordA,
                 (void *)&coordB, sizeof(Coord));

    Coord *coordC=
        (Coord *)malloc(sizeof(Coord));
    Coord coordD[1];

    swapGenerico((void *)coordC,
                 (void *)coordD, sizeof(Coord));
}
```