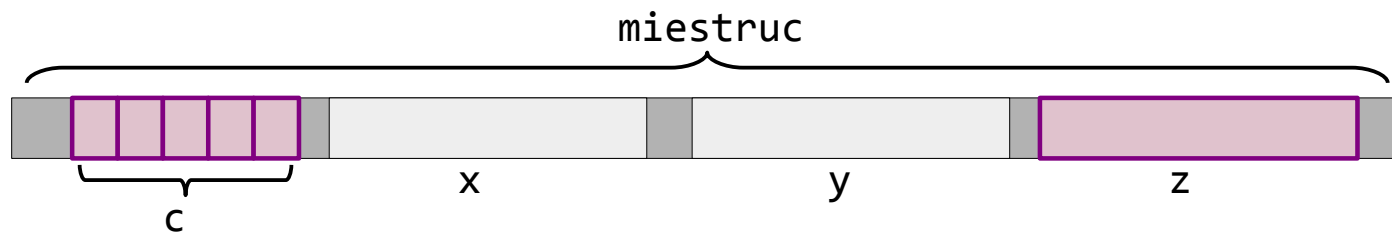


Tipus de Dades Derivats Irregulars: Exemple

```
MPI_Type_struct(count, lens, displs, types, newtype)
```

Crea un tipus de dades heterogeni (p.e. un struct de C)

```
struct {  
    char c[5];  
    double x,y,z;  
} miestruc;  
MPI_Datatype types[2] = {MPI_CHAR,MPI_DOUBLE}, newtype;  
int lengths[2] = { 5, 1 }; /* nomes volem enviar c i z */  
MPI_Aint ad1,ad2,ad3,displs[2];  
  
MPI_Get_address(&miestruc, &ad1);  
MPI_Get_address(&miestruc.c[0], &ad2);  
MPI_Get_address(&miestruc.z, &ad3);  
displs[0] = ad2 - ad1;  
displs[1] = ad3 - ad1;  
MPI_Type_struct(2, lengths, displs, types, &newtype);  
MPI_Type_commit(&newtype);
```

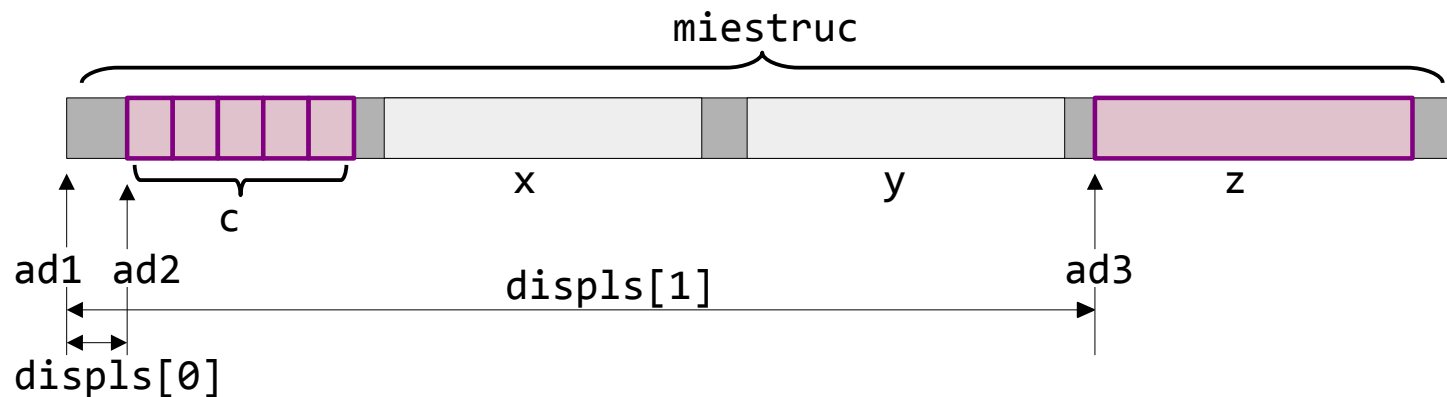


Tipus de Dades Derivats Irregulars: Exemple

```
MPI_Type_struct(count, lens, displs, types, newtype)
```

Crea un tipus de dades heterogeni (p.e. un struct de C)

```
struct {  
    char c[5];  
    double x,y,z;  
} miestruc;  
  
MPI_Datatype types[2] = {MPI_CHAR,MPI_DOUBLE}, newtype;  
int lengths[2] = { 5, 1 }; /* nomes volem enviar c i z */  
MPI_Aint ad1,ad2,ad3,displs[2];  
  
MPI_Get_address(&miestruc, &ad1);  
MPI_Get_address(&miestruc.c[0], &ad2);  
MPI_Get_address(&miestruc.z, &ad3);  
displs[0] = ad2 - ad1;  
displs[1] = ad3 - ad1;  
MPI_Type_struct(2, lengths, displs, types, &newtype);  
MPI_Type_commit(&newtype);
```



Tipus de Dades Derivats Irreg.: Exemple (cont.)

- A continuació, s'usaria el tipus de dades per enviar/rebre. Per exemple:

```
if (my_rank == 0) {  
    MPI_Send(&miestruc, 1, newtype, 1, 0, comm);  
} else {  
    MPI_Recv(&miestruc, 1, newtype, 0, 0, comm, &status);  
}
```

- Finalment, es destruiria el tipus de dades:

```
MPI_Type_free(&newtype);
```