

```

1         preceded by the number of such entries */
2
3     MPI_Datatype Zparticles, Ztype;
4
5     MPI_Aint zdisp[1000]
6     int zblock[1000], i, j, k;
7     int zzblock[2] = {1,1};
8     MPI_Datatype zztype[2];
9     MPI_Aint      zzdisp[2];
10
11     j=0;
12     for (i=0; i < 1000; i++)
13         if (particle[i].index==0)
14             {
15                 for (k=i+1; (k < 1000)&&(particle[k].index == 0) ; k++);
16                 zdisp[j] = i;
17                 zblock[j] = k-i;
18                 j++;
19                 i = k;
20             }
21     MPI_Type_indexed( j, zblock, zdisp, Particletype, &Zparticles);
22     /* Zparticles describe particles with class zero, using
23        their absolute addresses*/
24
25     /* prepend particle count */
26     MPI_Address(&j, zzdisp);
27     zzdisp[1] = MPI_BOTTOM;
28     zztype[0] = MPI_INT;
29     zztype[1] = Zparticles;
30     MPI_Type_struct(2, zzblock, zzdisp, zztype, &Ztype);
31
32     MPI_Type_commit( &Ztype);
33     MPI_Send( MPI_BOTTOM, 1, Ztype, dest, tag, comm);
34
35
36 Example 4.19 Handling of unions.
37
38     union {
39         int      ival;
40         float    fval;
41     } u[1000]
42
43     int      utype;
44
45     /* All entries of u have identical type; variable
46        utype keeps track of their current type */
47
48     MPI_Datatype  type[2];

```