

Altre Funzioni MPI

```
double MPI_Wtime(void)
```

La chiamata ritorna il tempo in secondi trascorso da una data nel passato. Il tempo e' relativo al processo.

```
{
    double starttime, endtime;
    starttime = MPI_Wtime();
    .... stuff to be timed ...
    endtime   = MPI_Wtime();
    printf("That took %f seconds\n",endtime-starttime);
}
```

Altre Funzioni MPI

```
double MPI_Wtick(void)
```

La funzione ritorna la risoluzione in secondi della funzione MPI_Wtime.

```
int MPI_Abort( MPI_Comm comm, int errorcode )
```

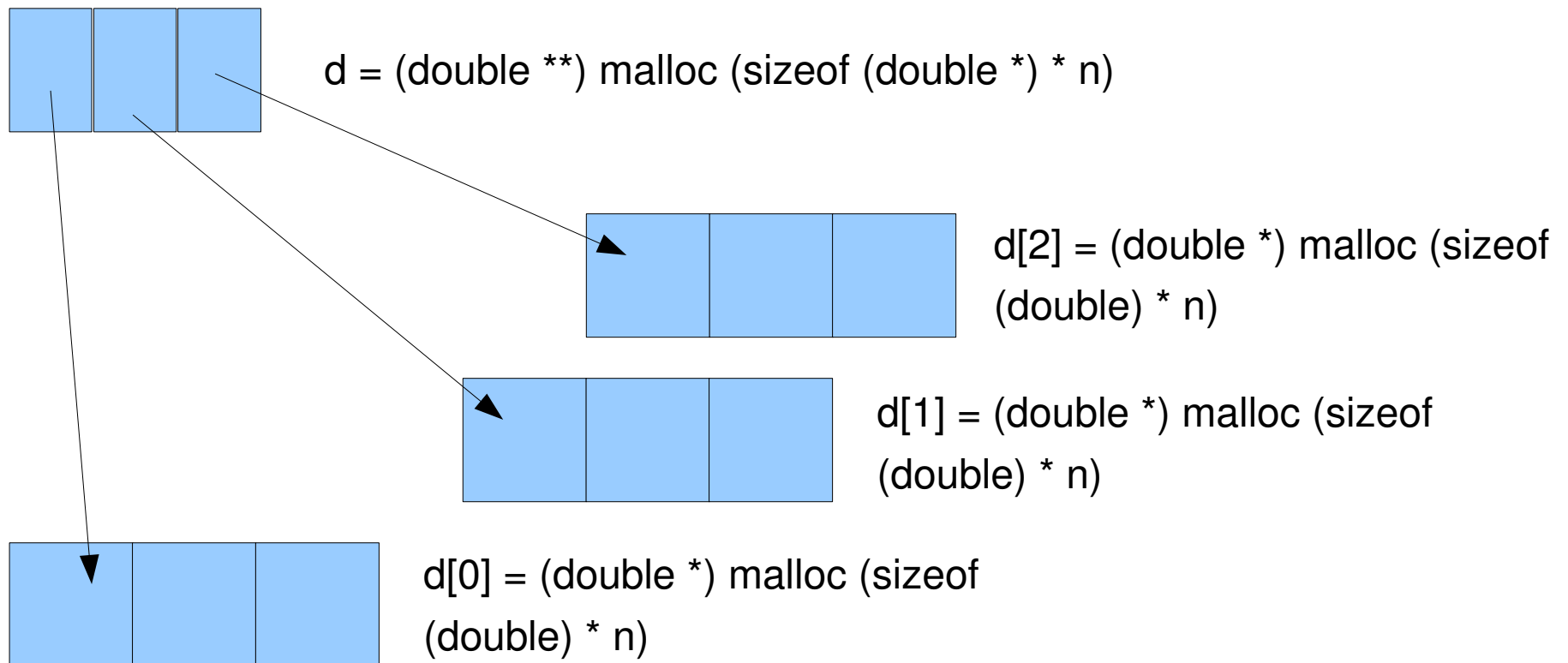
Termina tutti i processi appartenenti al comunicatore comm

Esercizio

Esercizio: scrivere un programma che dato N da linea di comando allochi dinamicamente spazio necessario a contenere una matrice di double **NxN**

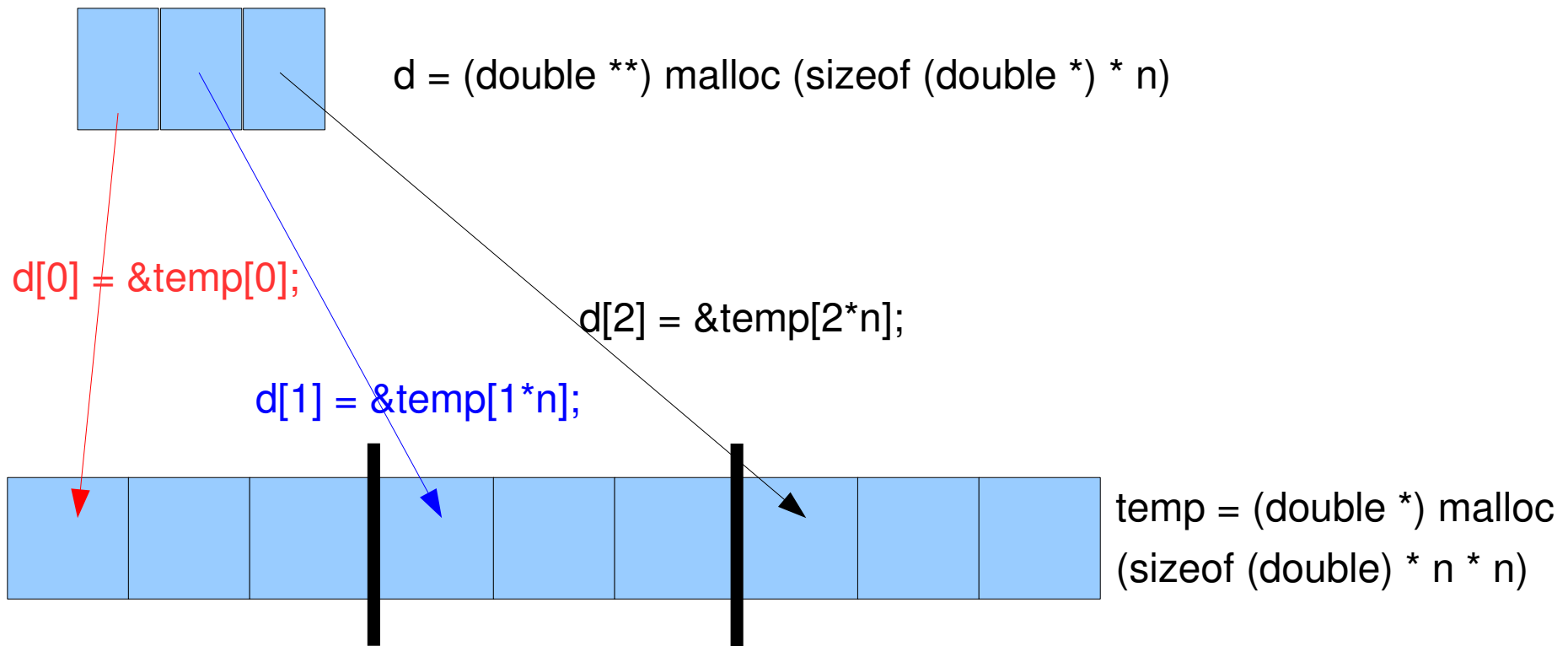
Allocazione dinamica della memoria

Vedi alloc.c (primo)



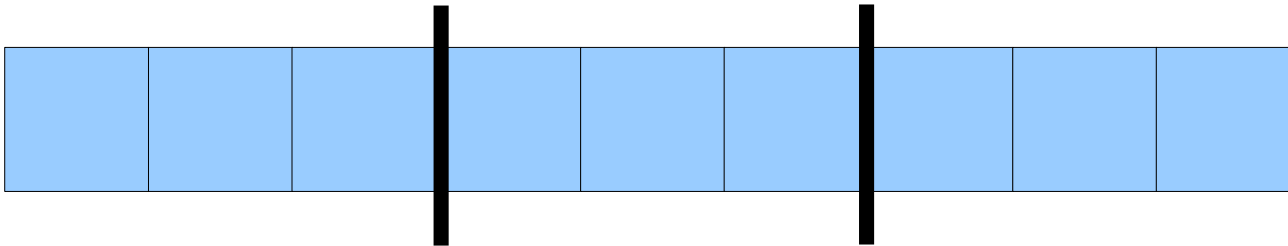
Allocazione dinamica della memoria

Vedi alloc.c (secondo)



Allocazione dinamica della memoria

Vedi alloc.c (terzo)



```
d = (double *) malloc (sizeof (double) * n * n)
```

Cenni sulla funzione `alloca` e su `valgrind`

Esercizi

Esercizio: adesso provate a fare il send e receive di una matrice tra due processi usando i diversi metodi di allocazione dinamica visti. Calcolate anche i tempi per il trasferimento dei dati necessari nei tre casi.