Hacettepe Üniversitesi
Bilgisayar Mühendisliği Bölümü
BİL431 Koşut İşlem
1. Ara sınav 25-11-2011

Soru 1. "Hypercube" topolojisiyle bağlı dağıtılmış bellekli MIMD türü bir bilgisayarda en uzak iki nokta arasında 1000 byte uzunluğunda bir ileti göndermek istiyoruz. Bağlantı teknolojisi olarak, 10Gbit/sn hızında (Kanal aktarım sığası – "Channel bandwidth") linkler kullanıldığını varsayarak 64 düğümden oluşan bir ağ için aşağıdaki soruları yanıtlayınız. (İleti yönlendiricilerde harcanan zamanı yok varsayınız.)

a) "Store-and-forward routing" yöntemiyle ileti aktarımı ne kadar sürer?
b) "Cut-through routing" ("wormhole routing") yöntemiyle ileti aktarımı ne kadar sürer? "Flit" (ileti parçacıkları) boyunu 10 byte olarak alınız.

Soru 2. "MPI_Barrier" işlevini, iki boyutlu "Torus" topolojisi için tasarlayınız. Tasarımınızı çizim üzerinde anlatınız.

Soru 3. "MPI_Barrier" işlevini, iki boyutlu "Torus" topolojisi için, diğer MPI işlevlerini kullanarak C programlama dili ile kodlayınız.

Kullanabileceğiniz MPI işlevleri:

```
int MPI_Init(int *argc, char ***argv)
int MPI_Finalize()
int MPI_Comm_size(MPI_Comm comm, int *size)
int MPI_Comm_rank(MPI_Comm comm, int *rank)


int MPI_Send(void *buf, int count, MPI_Datatype datatype,int dest,
                    int tag, MPI_Comm comm)
int MPI_Recv(void *buf, int count, MPI_Datatype datatype,int source,
                    int tag, MPI_Comm comm, MPI_Status *status)
int MPI_Sendrecv(void *sendbuf, int sendcount,MPI_Datatype
                    senddatatype, int dest, int sendtag,void *recvbuf,
                    int recvcount, MPI_Datatype recvdatatype,int
                    source, int recvtag, MPI_Comm comm,MPI_Status
                    *status)

int MPI_Get_count(MPI_Status *status, MPI_Datatype datatype,int
                    *count)

int MPI_Cart_create(MPI_Comm comm_old, int ndims, int *dims, int
                    *periods, int reorder, MPI_Comm *comm_cart)
int MPI_Cart_rank(MPI_Comm comm_cart, int *coords, int *rank)
int MPI_Cart_coord(MPI_Comm comm_cart, int rank, int maxdims,  int
                    *coords)
int MPI_Cart_shift(MPI_Comm comm_cart, int dir, int s_step, int
                    *rank_source, int *rank_dest)
```

`int MPI_Barrier(MPI_Comm comm)`
The only argument of MPI_Barrier is the communicator that defines the group of processes that are synchronized. The call to MPI_Barrier returns only after all the processes in the group have called this function.

Kullanabileceğiniz değişmezler: **`MPI_COMM_WORLD,MPI_ANY_SOURCE,MPI_ANY_TAG`**