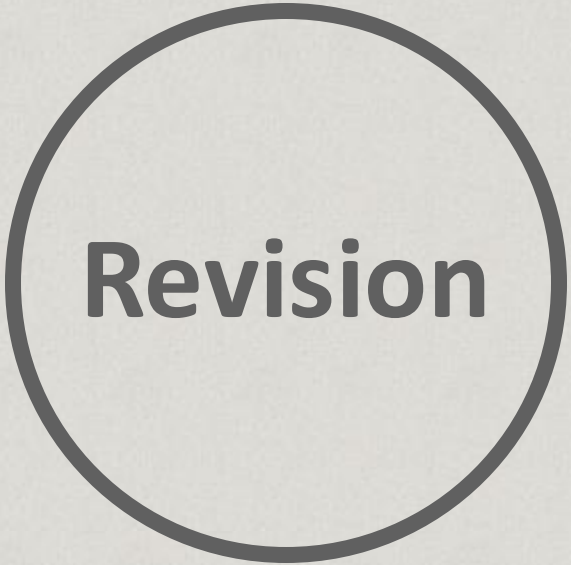
Distributed Computing

Introduction to MPI

Mahmoud Badry, mma18@fayoum.edu.eg

AGENDA

Mahmoud Badry, mma18@fayoum.edu.eg



Revising what we have taken so far.

Mahmoud Badry, mma18@fayoum.edu.eg

Revision

- Introduction to DC.
- Thread level vs. Parallel level.
  - Assignment 1: Refreshing grid using separate thread.
- Network programming (Socket programming – URL processing)
  - Assignment 2:
    - Modifying socket programming example.
    - Parsing URL components using Regex.

Mahmoud Badry, mma18@fayoum.edu.eg

Revision cont.

- Serialization.
- Synchronization.
  - Advanced threading (Sleeping Barber)
- NETTY framework. (Online video)
  - Assignment 3: Create a multiplayer network game.
- RMI.
  - Assignment 4: Small FTP server and client.
- MVC & RMI.

Mahmoud Badry, mma18@fayoum.edu.eg

Message passing programming

Mahmoud Badry, mma18@fayoum.edu.eg

Mesasge Passing Programming

- The **message-passing programming** model is based on the **abstraction** of a **parallel computer** with a **distributed address** space where each processor has a **local** memory to which it has exclusive **access**. There is no **global** memory.
- To **transfer** data from the **local memory** of one **processor A** to the **local memory** of another **processor B**, A must **send** a **message** containing the **data** to B, and B must receive the data in a **buffer** in its local memory.
- There are **no assumptions** on the **topology**, Instead, it is **assumed** that **each processor** can send a **message** to any other **processor**.

Mahmoud Badry, mma18@fayoum.edu.eg

Mesasge Passing Programming cont.

- A message-passing **program** is **executed** by a set of **processes** where each **process** has its own **local data**. Usually, one **process** is executed on one **processor** or **core** of the execution platform.
- The processes uses a communication library to exchange data between each other.
- These communication libraries contain a **point to point** and **global (broadcast) operations**.
- By far the most popular portable communication library is **MPI** (Message-Passing Interface) and **PVM** (Parallel Virtual Machine)).

Mahmoud Badry, mma18@fayoum.edu.eg

An introduction to Message passing interface

Mahmoud Badry, mma18@fayoum.edu.eg

Introduction to MPI

- The **Message-Passing Interface** (MPI) is a **standardization** of a message-passing **library** interface specification. MPI defines the **syntax** and **semantics** of library **routines** for **standard communication** patterns.
- Language bindings for C, C++, Fortran-77, and Fortran-95 are supported. Other languages implementations are available right now. We will focus on MPI with C to understand the basic concepts of MPI.
- There are two versions of the MPI standard:
  - MPI-1: Standard communication operations that work on static process.
  - MPI-2: Extends MPI-1 provides additional support for dynamic process management, and parallel I/O.

Mahmoud Badry, mma18@fayoum.edu.eg

MPI implementations

- Different MPI libraries can use different implementations, possibly using specific optimizations for specific hardware platforms. For the programmer, MPI provides a standard interface, thus ensuring the portability of MPI programs:
  - MPICH: https://www.mpich.org/
  - OpenMPI: http://www.open-mpi.org/
  - MESH-MPI.
  - MPI.NET: https://www.microsoft.com/en-us/download/details.aspx?id=56727
  - mpiJava: http://www.hpjava.org/mpiJava.html

Mahmoud Badry, mma18@fayoum.edu.eg

Blocking vs non-blocking operation

- **Blocking operation:** An MPI communication operation is blocking, if return of control to the calling process indicates that all resources, such as buffers, specified in the call can be reused, e.g., for other operations.
- **Non-blocking operation:** An MPI communication operation is non-blocking, if the corresponding call may return before all effects of the operation are completed and before the resources used by the call can be reused.

Mahmoud Badry, mma18@fayoum.edu.eg

Synchronous vs. asynchronous communications

- Synchronous communication: The communication between a sending process and a receiving process is performed such that the communication operation does not complete before both processes have started their communication operation.
- Asynchronous communication: Using asynchronous communication, the sender can execute its communication operation without any coordination with the receiving process.

Mahmoud Badry, mma18@fayoum.edu.eg

First MPI program

Mahmoud Badry, mma18@fayoum.edu.eg

Communicators

- In MPI, all communication operations are executed using a **communicator**. A communicator represents a communication domain which is essentially a set of processes that exchange messages between each other.
- The MPI default communicator MPI COMM WORLD is used for the communication. This communicator captures all processes executing a parallel program

Mahmoud Badry, mma18@fayoum.edu.eg

First MPI example

```
#include <stdio.h>
#include <string.h>
#include "mpi.h"

int main (int argc, char *argv[]) {
   int my_rank, p, tag=0;
   char msg [20];
   MPI_Status status;

   MPI_Init (&argc, &argv);
   MPI_Comm_rank (MPI_COMM_WORLD, &my_rank);
   MPI_Comm_size (MPI_COMM_WORLD, &p);
   if (my_rank == 0) {
      strcpy (msg, "Hello ");
      MPI_Send (msg, strlen(msg)+1, MPI_CHAR, 1, tag, MPI_COMM_WORLD);
   }
   if (my_rank == 1)
      MPI_Recv (msg, 20, MPI_CHAR, 0, tag, MPI_COMM_WORLD, &status);
   MPI_Finalize ();
}
```


Mahmoud Badry, mma18@fayoum.edu.eg

QUESTIONS

Mahmoud Badry, mma18@fayoum.edu.eg

THANK YOU!

Mahmoud Badry, mma18@fayoum.edu.eg