

Phys 7411: Message Passing Interface

Karen Tomko
ktomko@osc.edu
Spring 2009

Today's topics

- What is MPI
- MPI Standards and Implementations
- What is a Process
- Hello World

MPI

- Message Passing Interface
- A library and run-time system for developing programs consisting of multiple communicating processes running on 1 or more CPUs.
- Primary Goals
 - Run on distributed memory computer systems
 - Provide portability
 - Provide good performance

MPI Standards

- MPI Forum
 - Participants from:
 - Academia, Government labs, Computer vendors
 - Standards documents
 - <http://www.mpi-forum.org/>
- MPI 1.1 (1992-1994)
 - initial standard
- MPI-2 (1995-1997)
 - parallel I/O, one sided communication, dynamic task creation
- MPI-3 (2008-?)

MPI Implementations

- Open source implementations
 - MPICH / MPICH2
<http://www.mcs.anl.gov/research/projects/mpich2/index.php>
 - OpenMPI
<http://www.open-mpi.org/>
 - MVAPICH / MVAPICH2
<http://mvapich.cse.ohio-state.edu/>
- Vendor implementations
Cray, IBM, SGI,

MPI

- Message Passing Interface
- A library and run-time system for developing programs consisting of multiple communicating processes running on 1 or more CPUs.
- Primary Goals
 - Run on distributed memory computer systems
 - Provide portability
 - Provide good performance

The process concept

- *Program*: a static, algorithmic description, consists of instructions

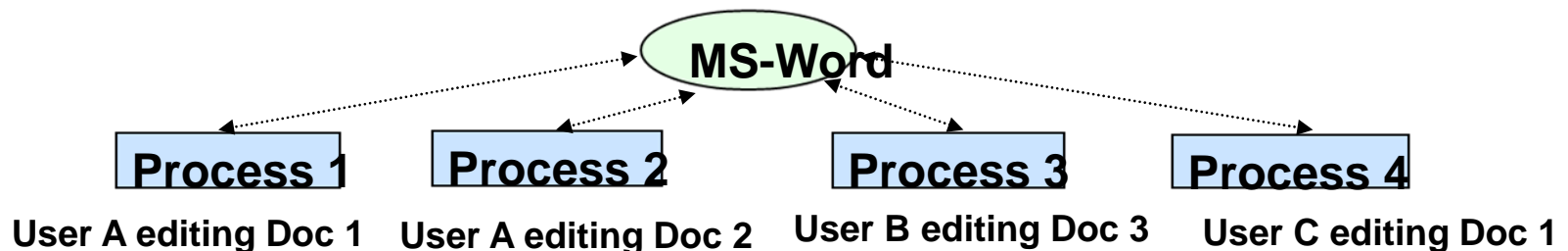
```
- int main() {  
    int i, prod=1;  
    for(i=0; i<100; ++i)  
        prod = prod*i;  
}
```

- *Process*: dynamic, consists of instruction executions

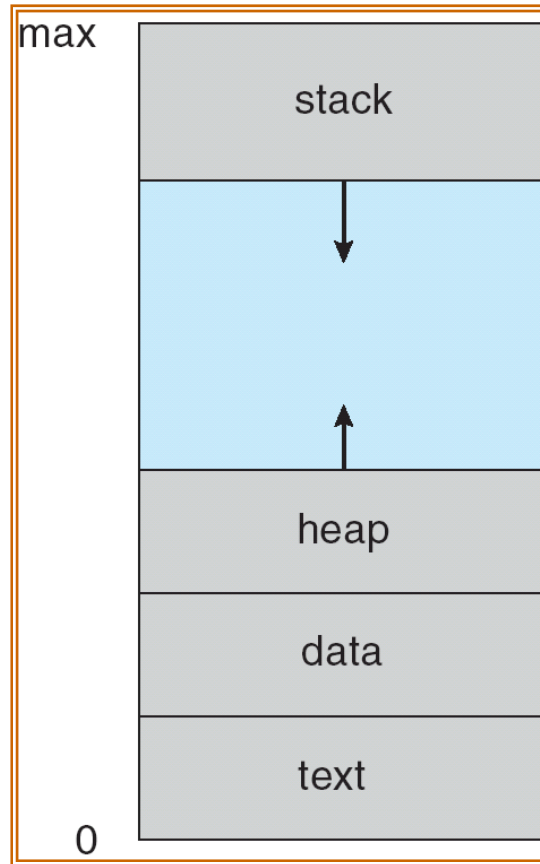
This slide borrowed from Chapter 3 lecture for *Operating Systems: A Design-Oriented Approach* by Charles Crowley – K. Tomko

The Process Concept (cont.)

- A **process** is an **execution of a program**
 - A **dynamic** concept; has a life cycle
 - One **static program**, multiple processes
 - One copy of MS-Word run by 3 different users
 - May run at different time; very likely use different input data
 - Multiple distinct processes



Process in Memory



hello.c

```
#include <mpi.h>
```

```
int main(int argc, char *argv[]) {  
int rank, size;
```

```
MPI_Init(&argc, &argv);  
MPI_Comm_rank(MPI_COMM_WORLD, &rank);  
MPI_Comm_size(MPI_COMM_WORLD, &size);
```

```
printf("I am %d of %d\n", rank, size);
```

```
MPI_Finalize ();  
}
```

hello.c (continued)

- Compile

```
> mpicc -o hello hello.c
```

- Run

```
> mpiexec -np 4 hello
```

```
I am 3 of 4
```

```
I am 0 of 4
```

```
I am 2 of 4
```

```
I am 1 of 4
```

```
>
```