

## Parallel Processing

### Project 1

turnin code: pp\_p1

Write a version of mpiexec and any supporting programs that you need to run MPI programs which use your own implementation of MPI. Your mpiexec should be able to run non-MPI programs as well. It should compile and link as p1.

p1 will find valid hostnames in a file named hostnames.

This name may be overridden by using the -f cmd-line arg.

Also support -n to specify the number of ranks, which defaults to 1.

If the file named hostnames is not in the local directory, and the -f option is not present, all ranks are to be started on the local host.

If the arg to -n specifies more ranks than in the hostnames file, then loop in that file.

All ranks should have these 3 variables in their environments:

PP_MPI_RANK	# rank of the process
PP_MPI_SIZE	# number of ranks started
PP_MPI_HOST_PORT	# hostname:9999 ; hostname is host where p1 is run

The value of the PORT portion of HOST\_PORT will change in later projects.

Make sure that stdout/stderr for ranks are printed on the controlling

tty for p1. Note that fork and exec of ssh will route output back to the controlling tty, and you have all pids from forks, so that you can wait for all ranks to terminate at the end, so, this is an attractive approach, but you may write you own (proxy) server code if you prefer.

Use turnin to submit a tar file of a directory with makefile and all source code to build your p1.

After un-tarring your tar file, I will simply run:

```
rm -f p1
```

```
rm -f *.o
```

```
make
```

to build p1 before testing.