

---

Dieses Übungsblatt soll Ihnen die Möglichkeit geben, Ihre ersten Schritte in der Programmierung mit dem MPI zu machen. Die erworbenen Fertigkeiten werden auf späteren Übungsblätter für komplexere Aufgaben benötigt werden.

## 1 Das erste MPI-Programm (150 Punkte)

Erstellen sie ein MPI-Programm `timempi` in C, welches eine ähnliche Ausgabe erzeugt wie das parallel gestartete Skript von Übungsblatt 3. Dabei sind folgende Vorgaben zu beachten:

- Die Prozesse mit den Rängen 1 bis n sollen den String `HOSTNAME: TIMESTAMP` bei sich erzeugen und als String per MPI an den Prozess mit Rang 0 senden, welcher die komplette Ausgabe übernimmt. (Tipp: Sehen Sie sich die Funktionen `gethostname()` und `gettimeofday()` an.)
- Die Ausgabe soll nach Rang der Prozesse geordnet erfolgen.
- Die Prozesse sollen alle erst beenden, wenn die Ausgabe komplett erfolgt ist. Das Programm ist falsch, wenn ein Prozess zu früh beenden könnte!
- Direkt vor dem Beenden soll jeder Prozess einen Text ausgeben: „Rang X beendet jetzt!“ (Tipp: Dazu können Sie `MPI_Barrier()` verwenden.)
- Das Programm muss mit beliebig vielen Prozessen lauffähig sein.

Die Ausgabe könnte wie folgt aussehen:

```
west1: 2012-11-14 13:15:57.968558
west2: 2012-11-14 13:15:57.968557
Rang 0 beendet jetzt!
Rang 1 beendet jetzt!
Rang 2 beendet jetzt!
```

## 2 Ergebnisse sammeln im MPI-Programm (30 Punkte)

Erweitern Sie Ihr MPI-Programm `timempi` zu `timempi2` um folgende Funktion:

- Direkt nach der Ausgabe der empfangenen Strings soll der Prozess mit Rang 0 noch den kleinsten Mikrosekunden-Anteil aller Prozesse ausgeben.  
(Tipp: `MPI_Reduce()` bietet sich an.)

Die Ausgabe könnte wie folgt aussehen:

```
west1: 2012-11-14 13:15:57.968558
west2: 2012-11-14 13:15:57.968557
968557
Rang 0 beendet jetzt!
Rang 1 beendet jetzt!
Rang 2 beendet jetzt!
```

## Abgabe

Als Abgabe erwarten wir ein gemäß den Vorgaben benanntes komprimiertes Archiv (`.tar.gz`), das ein gemäß den Vorgaben benanntes Verzeichnis mit folgendem Inhalt enthält:

- Die Quellen der C-Programme `timempi.c` und `timempi2.c`.
- Ein Makefile derart, dass `make timempi`, `make timempi2`, `make clean` und `make` erwartungsgemäße Binärdateien erzeugen bzw. löschen. `make` soll dabei alle Binärdateien auf einmal erzeugen.
- **Keine** Binärdateien.

Senden Sie das Archiv per Mail an: `hr-abgabe@wr.informatik.uni-hamburg.de`.

## Rückmeldung (+ 5-10 Punkte)

Bearbeitungszeit			
Schwierigkeit	<input type="checkbox"/> zu leicht	<input type="checkbox"/> genau richtig	<input type="checkbox"/> zu schwer
Lehrreich	<input type="checkbox"/> wenig	<input type="checkbox"/> etwas	<input type="checkbox"/> sehr
Verständlichkeit	<input type="checkbox"/> großteils unklar	<input type="checkbox"/> teilweise unklar	<input type="checkbox"/> verständlich
Kommentar			