

Lab 12: Multi Processor Systems

Persönliche Lernkontrolle

- Welche der folgenden Aussagen sind richtig (mehrere sind möglich):

 - Ein Thread kann mehrere Prozesse beinhalten
 - Mehrere Threads teilen sich denselben Adressbereich im Speicher
 - Ein Prozess besteht mindestens aus einem Thread (LWP)
 - Mehrere Prozesse teilen sich denselben Adressbereich im Speicher
 - Ein Debugger kann neue Programme ausführen, sowie sich an bereits laufende Programme anhängen
- Beschreiben Sie den Ablauf wie eine Quelldatei in eine ausführbare Datei übersetzt wird:

Der Sourcecode wird mittels compiler in eine ausführbare Datei konvertiert (compiliert)
- Wie heisst der Befehlsaufruf des Compilers und des Debuggers unter Unix (Solaris):

Mittels GCC Compiler compilieren:

```
cc -g -o [output] [source]
cc -g -o hello hello.c
```

-g Compile for debugging

-o Object file [Output]

dbx [prg] -p [ProcessID]

dbx ./prime -p 2900
- Wie viele CPUs stehen Ihnen auf Ihrem System zur Verfügung? Nennen Sie auch, wie Sie diesen Wert ermittelt haben:

4 CPUs, z.B. mittels mpstat:

```
$ mpstat 5
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
0 2 0 0 327 104 43 0 0 0 0 28 0 0 0 100
1 2 0 0 54 5 46 0 0 0 0 25 0 0 0 100
2 2 0 0 64 11 48 0 0 0 0 24 0 0 0 100
3 2 0 0 61 4 53 0 0 0 0 25 0 0 0 100
```
- Was müssen Sie beachten wenn Sie rechenintensive Threads (oder Prozesse erzeugen) erzeugen, im Bezug auf die Anzahl der zur Verfügung stehenden CPUs?
(Hinweis: $n > m$, $n = m$, $n < m$)

Man sollte nicht mehr Threads / Prozesse als CPUs ausführen. So kann ein Thread / Prozess einen CPU optimal auslasten und muss nicht um die Ressourcen „kämpfen“ (warten).