

## Vorwort

Die Studierenden lernen die Techniken zur Programmierung paralleler Abläufe auf Ein- und Multiprozessorsystemen sowie in heterogenen, verteilten Systemen kennen und sind danach in der Lage, selbstständig Lösungen für vorgegebene Problemstellungen zu erarbeiten. Die Programme werden in der Programmiersprache *C* oder *Java* unter dem Betriebssystem UNIX realisiert.

Heutige Problemstellungen (Wetter- oder Erdbebenvorhersagen, *Crashtest*-Simulationen, Entwicklung neuer Medikamente oder chemischer Verbindungen, Suchmaschinen von Web-Dienstleistern, ...) erfordern enorme Rechen- und Speicherleistungen, die im Allgemeinen nur mit Hochleistungsrechnern, Multiprozessorsystemen oder in Rechnernetzen erbracht werden können. Damit solche Umgebungen effizient genutzt werden können, müssen die Programmierer parallelisierbare Programme erstellen.

Im ersten Teil der Lehrveranstaltung werden deshalb einige grundlegende Verfahren zur Programmierung paralleler Abläufe auf Ein- und Multiprozessorsystemen vorgestellt, während sich der zweite Teil der Veranstaltung mit den Mechanismen zur Programmierung paralleler Abläufe in heterogenen, verteilten Systemen beschäftigt. Zusätzlich zu diesem Skript sollten auch entsprechende Fachbücher, die *Manual Pages* und vergleichbare Dokumente zu den einzelnen Themen gelesen und benutzt werden.

Zunächst werden einige UNIX-spezifische Systemaufrufe zur Generierung von Prozessen und zur Interprozesskommunikation vorgestellt. Nachdem die Teilnehmerinnen und Teilnehmer der Veranstaltung die Programmierung paralleler Prozesse kennen gelernt haben, wird im Kapitel *Threads* die Programmierung paralleler Abläufe innerhalb eines Prozesses vermittelt. Der zweite Teil der Lehrveranstaltung beginnt mit der sehr einfachen *Socket*-Schnittstelle, mit deren Hilfe Programme in heterogenen, verteilten Systemen implementiert werden können. *Sockets* werden sowohl von UNIX als auch von Windows 98/NT/2000/XP/... unterstützt. Zum Schluss wird die Programmierumgebung *Message Passing Interface* (MPI) vorgestellt, die speziell im technisch-wissenschaftlichen Bereich verbreitet ist. Alle in dieser Lehrveranstaltung erstellten Programme **müssen meinen Programmierrichtlinien entsprechen**, die Sie unter den URL's

[http://www.hs-fulda.de/~gross/c\\_richt.pdf](http://www.hs-fulda.de/~gross/c_richt.pdf) bzw.

[http://www.hs-fulda.de/~gross/java\\_richt.pdf](http://www.hs-fulda.de/~gross/java_richt.pdf)

finden. Damit Sie die verschiedenen Verfahren möglichst schnell und gut kennen lernen, erarbeiten Sie selbstständig Lösungen zu den Aufgaben und führen die Ergebnisse im Rahmen der Lehrveranstaltung vor. Falls Sie auch zu Hause arbeiten wollen, können Sie entweder auf Ihrem Rechner **Cygwin**, **Linux** oder **Solaris x86** installieren oder eine **VPN-Verbindung** in das Hochschulnetz benutzen (Sie benötigen einen schnellen Internet-Zugang, z. B. über DSL). *Cygwin*, *Linux* und *Solaris x86* stellen alle Systemaufrufe der Kapitel 2 bis 4 standardmäßig zur Verfügung. Die notwendige Entwicklungsumgebung für das Kapitel 5 kann entweder aus dem Internet geladen werden oder Sie installieren sie selbst aus dem Quellcode. Eine Beschreibung zur Installation und Konfiguration von *Cygwin* und *MPI* unter *Cygwin* finden Sie auf meiner Web-Seite (<http://www.hs-fulda.de/~gross/cygwin/index.htm>) und eine Beschreibung zur Installation der VPN-Software auf der Seite des Rechenzentrums (<http://www.hs-fulda.de/dvz>). Neben der VPN-Software benötigen Sie noch einen *Secure Shell Client* (z. B. OpenSSH oder PuTTY), damit Sie

eine Verbindung zu den UNIX-Rechnern aufbauen können. Wenn Sie die Software installiert und konfiguriert haben, können Sie z. B. eine DSL-Verbindung zu Ihrem Netzanbieter (*Provider*) aufbauen und dann den *VPN-Client* starten. Nachdem Sie sich mit Ihrem Benutzernamen und Passwort in der Hochschule angemeldet haben, können Sie sich mit *ssh* (interaktive Sitzung) oder *sftp* (Dateitransfer) auf einem UNIX-Rechner anmelden. Falls Sie auf dem UNIX-Rechner Programme mit einer grafischen Oberfläche benutzen wollen, benötigen Sie auf Ihrem Rechner einen *X-Server* (z. B. das kommerzielle Produkt *X-Win32* oder ein freies Produkt wie *Mocha X Server* bzw. *Xming*; alternativ können Sie auch den *X-Server* einer Cygwin-Installation benutzen). Eine Beschreibung zur Installation und Konfiguration von *Xming* finden Sie auf meiner Web-Seite (<http://www.hs-fulda.de/~gross/xming/index.htm>)

Wenn Sie auf Ihrem Rechner *Cygwin* installieren, können Sie alle Programme lokal auf Ihrem Rechner entwickeln und sind unabhängig von der Verfügbarkeit der Rechnernetze und der Hochschulrechner. Falls Sie *Xming* benutzen, kann es passieren, dass die VPN-Verbindung beendet wird, wenn Sie zu lange „nachdenken“ (eventuell abhängig vom *Provider* der DSL-Verbindung?). In diesem Fall können alle Änderungen seit der letzten (automatischen) Speicherung des Dokuments verloren gehen, so dass Sie auf keinen Fall vergessen sollten, die letzten Änderungen zu speichern, bevor Sie eine Pause machen.

Ihre Programme müssen unter den Betriebssystemen Linux und Solaris fehlerfrei übersetzt werden und ablauffähig sein.