



Informationen zur Bearbeitung und Abgabe von Übungsaufgaben im Lehrgebiet Programmierung komplexer Systeme

1 Regeln

Soweit nicht anders angegeben, gelten die folgenden **verbindlichen Regeln** für alle Übungen des Lehrgebiets.

1. Soweit keine Gruppenarbeit explizit für die Veranstaltung erlaubt wurde, werden individuelle und selbstständig erstellte Lösungen zu allen Aufgaben erwartet.
2. Die Übungen werden unter Linux durchgeführt. Die Lösungen müssen auf den Laborrechnern des Labors für parallele und verteilte Systeme bzw. dem WR-Cluster lauffähig sein und wo verlangt Leistungswerte auf diesen Rechnern ermittelt werden.
3. Die zu benutzenden Rechner werden in jeder Veranstaltung angegeben.
4. Legen Sie in ihrem Home-Verzeichnis ein Unterverzeichnis `xyz` zur Veranstaltung `xyz` an, in das Sie alle Unterlagen zu dieser Veranstaltung ablegen. Legen Sie darin zu jeder Übung ein eigenes Unterverzeichnis an, das alle Unterlagen zu der entsprechenden Übung aufnimmt (Source, Object-Dateien, Shell-Skripte, Ausführungsanweisungen, Makefiles, Dokumentation,...). Beispiel für die erste Übung zur Veranstaltung `xyz`: `$HOME/xyz/u1`. Legen Sie in solch einem Unterverzeichnis für jede Aufgabe wiederum ein eigenes Unterverzeichnis an. Beispiel: `$HOME/xyz/u1/a1` für die erste Aufgabe der ersten Übung.
5. In dem Unterverzeichnis zu einer Übung muss vorhanden sein:
 - (a) Eine **README-Datei** mit einer klar verständlichen kurzen Anleitung zum Erzeugen und Ausführen der Anwendung. Geben Sie darin auch klar an, welche Dateien auf welchen Rechnern (ggfs. in welchen Verzeichnissen) vor einem Programmstart verteilt werden müssen und welche Schritte zum Installieren, Übersetzen und Ausführen notwendig sind.
 - (b) Ein **Makefile** zum Erzeugen von ausführbarem Code. Ein Aufruf von `make` muss das Erzeugen aller notwendigen ausführbaren Dateien für diesen Rechner bewirken.
 - (c) Der komplette **kommentierte Source-Code** zur Aufgabe.
 - (d) Sonstige notwendige Dateien zum Übersetzen oder Ausführen.
 - (e) Ggfs. eine Datei mit Ergebnissen etc.
6. Ihre vollständigen, lauffähigen und komplett ausgetesteten Lösungen reichen Sie bis spätestens zum jeweils angegebenen Abgabetermin ein.
7. Dazu kopieren Sie den Ordner `xyz` zur Übung `xyz` aus ihrem Home-Verzeichnis komplett in das Verzeichnis `/home/veranstaltungsname/username/xyz` (`veranstaltungsname` ist der Name der Veranstaltung, `username` ist ihr Benutzername) auf einen Rechner, der in der Veranstaltung bekannt gegeben wird. Üblicherweise ist dies `wr0.wr.inf.h-brs.de`. Mit Ablauf der Abgabefrist um 23:59 Uhr des angegebenen Fristtages wird dieser Ordner (automatisch) zur Bewertung kopiert. Spätere Änderungen an den dortigen Dateien zeigen also keine Wirkung.
8. Liegt bis zum Stichtag (Eingang 23.59 Uhr) keine Lösung bzw. kein erkennbarer Lösungsansatz für eine Aufgabenstellung vor, so wird diese Aufgabe als nicht gelöst gewertet, eine Nachbesserung ist insofern dann auch nicht möglich.

9. Eingereichte Lösungen, die nicht genau das oben beschriebene Format haben oder keine bzw. keine geeignete README/Makefile-Datei enthalten, bekommen den Status "falsch bearbeitet" mit der Möglichkeit zur Nachbesserung (s.u.)
10. Werden an der Lösung Mängel festgestellt, so wird eine Mail an Sie verschickt, in der die Mängel aufgelistet werden und die Frist (üblicherweise 7-14 Tage nach Abschicken der Mail) genannt wird, bis zu der die Nachbesserung vorliegen muss. Für diese Nachfrist gelten obige Regelungen (Stichtag, Format) ebenfalls. Es gibt pro Aufgabe genau eine Nachbesserungsmöglichkeit.
11. Nachbesserungen schicken Sie direkt an die Email-Adresse `rudolf.berrendorf@h-brs.de` als `tgz`-Datei der gesamten Übung. Also keine `zip`-Dateien, auch nicht mehrere Einzeldateien etc. Als Subject-Zeile der Mail geben Sie an: `xyz Uab` für Übung `ab` der Veranstaltung `xyz`. Beispiel zur Erzeugung einer `tgz`-Datei zur ersten Übung:
`cd $HOME/xyz; tar czf u1.tgz u1.`
12. Für Gruppenarbeiten gilt, dass genau ein Person einer Gruppe wie oben beschrieben abgibt. Im README zu einer Aufgabe wird jeweils offensichtlich erkennbar angegeben, wer Teilnehmer dieser Gruppe ist. Sollte (aus Versehen) eine Lösung mehrfach abgegeben werden (als Gruppenlösung markiert und ein weiteres Gruppenmitglied hat in seinem eigenen Ordner eine weitere Lösung), so werde ich nichtdeterministisch eine der Lösungen auswählen.

2 Softwarebibliothek libFHBRS

Für einige Programme benötigen Sie zusätzliche Funktionen (Grafik, Zeitmessung, Zufallszahlen), die auf den Laborrechnern des Labors für parallel und verteilte Systeme und auf dem WR-Cluster schon in Form einer Bibliothek installiert sind. Die Header-Datei liegt unter `/usr/local/include/libFHBRS.h`, die Bibliothek unter `/usr/local/lib/libFHBRS.a`. Wenn Sie Programme auf einem anderen Unix-Rechner testen wollen, müssen Sie diese Software-Bibliothek aus dem Netz laden und anschließend dort erst installieren. Sie finden den Source-Code zu dieser Bibliothek auf der WWW-Seite zur Veranstaltung. Um die Bibliotheksfunktionen nutzen zu können müssen Sie im C-Programm ein `#include <libFHBRS.h>` einfügen und beim Linken zusätzlich angeben: `-L/usr/local/lib -lFHBRS -lX11 -lpthread -lm`.

3 Zeitmessungen

Bei einigen Aufgaben werden Zeitmessungen benötigt, um z.B. die Laufzeit eines Algorithmus zu bestimmen. Dazu bietet die libFHBRS geeignete Zeitmessroutinen an. In der Bibliothek ist dazu folgende Funktion definiert: `double gettime(void)` liefert einen aktuellen Zeitwert, womit durch Differenzbildung der Zeitunterschied (und damit die abgelaufene Zeit) zwischen zwei Aufrufen dieser Funktion gemessen werden kann. Der Overhead für einen Aufruf ist ca. 1-2 Microsekunden, die Auflösung beträgt ca. 1 Mikrosekunde.

Beispiel zur Nutzung:

```
/* Datei: zeit.c */
#include <libFHBRS.h>

int main(int argc, char **argv)
{
    double t0, t1;

    t0 = gettime();
    ...
    t1 = gettime();
    printf("That part of the program took %.6f seconds", t1-t0);
}
```

Übersetzen, Linken und Ausführen mit:

```
prompt> cc -I/usr/local/include zeit.c -o zeit -L/usr/local/lib -lFHBRS -lX11 -lpthread -lm
prompt> ./zeit
```