

Protokoll der Praktikumsdurchführung
BS Praktikum 2 – Gruppe 1
Ivan Morozov und Chandrakant Swaneet Kumar Sahoo
sowie
Kai Bursch und Matthias Nitsche
16.11.2013

Wir haben uns mit einer anderen Gruppe zusammengeschlossen und die Aufgaben gemeinsam zu bearbeiten und Probleme gemeinsam zu lösen. Morozov und Sahoo sowie Nitsche und Bursch haben sich zusammengesetzt.

Aufgabe 1 – Makefile:

Unser makefile lautet:

```
CC=/usr/bin/gcc
CFLAGS= -pthread -g -Wall -lpthread

TARGET = $1

$(TARGET): $(TARGET).c
    $(CC) $(CFLAGS) -o $(TARGET) $(TARGET).c
```

Mit `make critsect02` können wir die Datei erstellen.

Aufgabe 2 - Philosophenproblem:

Protokoll – Und was wir gelernt haben:

- Wir hatten bereits auf einem Mac mit XCode(einer IDE für die C Sprache) einen Tielder Aufgabe bearbeitet. Dort waren alle Funktionen enthalten, bis auf die beiden `get_sticks` und `put_sticks` sowie dem `disp_philo_states`. Bei der Bewältigung der Aufgabe haben wir uns öfters an dem vorgegebenen Beispiel `critsect02.c` sowie an anderen Codebeispielen aus dem pub orientiert. Wir hatten auch Hinweise von der anderen Partnergruppe die fertig war erhalten. Wir haben uns an unserer Vorlage aus XCode orientiert und nach und nach Bestandteile unseres Codes auf der OpenSuse-VM importiert und getestet. Wir haben zunächst die Initialisierung und den Lebenszyklus der Threads/Philosophen sowie den InputLoop mit den Benutzerbefehlen eingefügt. Nachdem wir die Ausgabe der Philosophen(`disp_philo_states`) geschrieben hatten, haben wir uns an dem `get_sticks` und `put_sticks` rangesetzt und unter Verwendung der genannten Threadfunktionen eine funktionierende Simulation der Spesenden Philosophen erstellt.
- Unser makefile enthält auch ein `clean` und `all` was den Vorgang des Compilens erleichtert.
- Wir haben den gcc-Compiler im makefile auch den Befehl `-std=c99` gegeben, damit wir Iterationsvariablen direkt im Kopf der For-Schleifen definieren können und diese daher lokal halten können. Das Default-Verhalten lässt diese (für uns als AI-ler angewohnte) Deklaration nicht zu.
- Es ist nicht klug auf einem Mac mit XCode zu entwickeln, da viele Sachen auf der VM nicht mehr funktionierten. Nächstes mal, werden wir direkt auf openSuse in der VM schreiben und auf der VM schrittweise testend entwickeln.
- Nicht nur haben wir viel über Threads und Synchronisierung in C verstanden, sondern auch über die Organisation einer größeren Gruppe – und dass größere

Gruppe schwerer zu koordinieren sind und nicht unbedingt (für diesen Aufgabenkontext) einen Nutzen gebracht haben.

- Wir haben auch festgestellt, dass das Git-Protokoll zusammen mit Github sehr nützlich als Versionsmanagement für die Gruppenarbeit und die Verteilung war.
- Dennoch heißt das nicht, dass jeder der einen Notepad++ hat, auch unbedingt Änderungen am Code machen und sie (auf GitHub) pushen sollte. Ohne IDE Änderungen zu machen ruft viele Fehler hervor, die nicht vom Änderernden sofort festgestellt werden können. Dies e müssen dann mühselig von Anderen behoben werden. „Don't commit without an IDE.“
- Bezüglich Doxygen haben wir festgestellt, dass es ein komfortables und leistungsstarkes Tool zum automatischen Generieren von HTML-Dokumentationen ist.
- Doxygen unterstützt nur C Code, aber keine Präprozessor-Anweisungen.