Prof. Franz Schweiggert
Christian Ehrhardt
Abteilung Angewandte Informationsverarbeitung

9. Mai 2006
Blatt 3

Systemnahe Software (SS 2006)

Abgabetermin 16.Mai.2006 2006

Fork (10 Punkte)

Kleine Beispiele (3 Punkte)

Auf der Homepage der Vorlesung gibt es unter Sonstiges zwei kleine Beispiele zu `fork(2)`. Schaut Euch diese Beispiele an und überlegt Euch, was sie jeweils für eine Ausgabe produzieren. Beim zweiten Beispiel sollten dabei verschiedene Werte (0 und 1) für das Makro `OPENFIRST` getestet werden (natürlich wird auch ein Datei `INPUT` benötigt). Begründet warum die Ausgaben jeweils so sind, wie sie sind! Warum unterscheiden sich die Ausgaben beim zweiten Beispiel je nach dem welcher Wert für `OPENFIRST` verwendet wird? Testet eine sich daraus gegebenenfalls ergebende Vermutung auch mit größeren Eingabedateien.
Eure Begründung solltet Ihr Eurem Tutor schriftlich per Mail rechtzeitig vor dem Tutorium zukommen lassen.

Parallelisierter GGT (7 Punkte)

Gegeben ist eine Datei, deren Name auf der Kommandozeile an Euer Programm übergeben wird (auf der Homepage gibt es passende Beispiele). Diese Datei besteht ausschließlich aus einer Folge von Integer-Zahlen. Jede Integerzahl besteht aus genau vier Bytes und die Datei enthält eine gerade Anzahl von Integern. Eure Aufgabe ist es, nacheinander immer genau 8 Bytes (also zwei Integerzahlen) aus der Datei einzulesen, ihren größten gemeinsamen Teiler(ggt) zu berechnen und das Ergebnis menschenlesbar auf die Standardausgabe auszugeben.
Damit das ganze nicht so langweilig wird, sollen mehrere Prozesse gleichzeitig an der ggt-Berechnung arbeiten (die Anzahl wird auf der Kommandozeile angegeben). Mit einem einzigen Zahlenpaar beschäftigt sich natürlich immer nur ein einziger Prozeß, weitere Prozesse können sich aber zeitgleich mit anderen Zahlenpaaren beschäftigen. Natürlich soll für jedes Zahlenpaar in der Datei nur einmal der ggt berechnet werden, die Datei muß also (gleichmäßig) unter den Prozessen aufgeteilt werden.

Hinweise

- Das Zahlenformat in der Eingabedatei ist offenbar nicht menschenlesbar.
- Der erste Prozeß sollte zunächst genau so viele Kinder erzeugen, daß insgesamt die auf der Kommandozeile angegebene Anzahl an Prozessen zusammen kommt.
- Jeder Prozeß bekommt eine Nummer und die Eingabedatei wird gleichmäßig unter den erzeugten Prozessen aufgeteilt. Dazu muß vorher die Dateigröße bestimmt werden.
- Aus dem zweiten Beispiel aus dem ersten Teil ergibt sich, daß man den Zeitpunkt, an dem die Datei geöffnet wird, richtig (relativ zu `fork`) wählen muß.
- Alle Prozesse lesen anschließend, in 8 Byte Blöcken ihren Teil der Eingabedatei.
- Um eine Vermischung der Ausgabe von verschiedenen Prozessen zu vermeiden, empfiehlt es sich, die vollständige Ausgabe für ein Zahlenpaar mit einer einzigen `write`-Operation zu erledigen. Das liefert zwar streng genommen keine Garantie, hilft aber in den aller meisten praktischen Fällen.