1. Übung: nebenläufiges Sortieren **Realtime Systems**

Abgabe: 21. April 2008 (für die Gruppen 1 und 3)
28. April 2008 (für die Gruppe 2)

Angaben zur Aufteilung der Gruppen und zu den Terminen siehe: www.tfh-berlin.de/~buchholz

Schreiben Sie ein C-Programm, das ungefähr den folgenden Aufbau besitzt:

```
#define FELDMAX 100

void sortiere(int feld[], int anz); //Funktions-Prototypen
void mische(int fein[], int faus[], int anz);

void sortiere(int feld[], int anz)
{
      ...
}

void mische(int fein[], int faus[], int anz)
{
      ...
}

main()
{
      ...
      int feld[FELDMAX];
      int feldaus[FELDMAX];

// Füllen von feld mit Zufallszahlen mit Werten von 0 bis 999

      sortiere(feld, FELDMAX/2);
      sortiere(feld +  FELDMAX/2, FELDMAX/2);
      mische(feld, feldaus, FELDMAX);

      ...
}
```

Nach dem erfolgreichen Test des entsprechenden Programms folgt das eigentlich Wichtige:

Ein Elternprozess (EP) füllt das Feld mit Zufallszahlen und erzeugt einen Kindprozess (KP). Der EP sortiert die erste Feldhälfte, wartet dann auf das Ende des KP, übernimmt die sortierte Teilliste des Kindprozesses, mischt die beiden sortierten Feldhälften und gibt das Ergebnis aus. Der KP sortiert die zweite Feldhälfte, übergibt diese an den EP und beendet sich.

Verwenden Sie für die Interprozess-Kommunikation (IPC: interprocess communication) den Pipe-Mechanismus (wird in der zweiten Hälfte der Vorlesung RTS näher erläutert). Das geschieht nach folgendem Schema (die für die IPC relevanten Bestandteile sind ***kursiv*** gedruckt!):

EP

```
...
int fd[2];
...
pipe(fd);
// fork ausführen
// sortiere 1. Hälfte
close(fd[1]);
read(fd[0], feld+FELDMAX/2,
     FELDMAXMAX/2*sizeof(int));
close(fd[0]);
// mische
// wait
```

KP

```
...


// sortiere 2. Hälfte
close(fd[0]);
write(fd[1], feld+FELDMAX/2,
      FELDMAX/2*sizeof(int));
close(fd[1]);
```

Achten Sie in Ihrer Anwendung auf den korrekten Bereich der zu übertragenden Werte in feld. Der Aufruf von `fork()` darf erst hinter dem Aufruf von `pipe(...)` erfolgen (warum?)! Das Warten auf die Beendigung des KP darf im EP nich schon z.B. nach dem Sortieren der ersten Feldhälfte erfolgen (warum?).

Wichtige Anforderungen an die Lösung dieser und der folgenden Aufgaben:

- Gruppenarbeit ist für die Aufgaben in RTS nicht zulässig!
- Die Abnahme der Übungen erfolgt spätestens zum angegebenen Endtermin in Ihrem Übungsblock. Kommen Sie bitte nicht erst am Ende des Blockes, da dann eventuell keine Zeit mehr zur Abnahme bleibt.
- Versehen Sie Ihr Listing stets mit Ihrem Namen und der Matrikelnummer.
- Achten Sie auf sinnvolles Einrücken im Quelltext (C-üblich) und eine dem Lesen förderliche Nutzung von Leerzeilen. Die maximale Zeilenlänge ist auf 80 Zeichen zu beschränken.
- Stellen Sie jeder selbst implementierten Funktion einen sinnvollen Header voran, der die Aufgabe der Funktion, verwendete Parameter (Eingangs-, Ausgangs-und Durchgangs-Parameter) sowie mögliche Fehlerreaktionen beschreibt.

Es gilt die folgende Regelung für die Abnahme der Übungsaufgaben:

Die Aufgaben werden von der Lehrkraft, die die Übungen betreut auf den Rechnern im Raum D132 (nicht auf einem Laptop) abgenommen. Zusätzlich senden Sie bitte unbedingt per email an

`buchholz@tfh-berlin.de` bzw.
`dinse@tfh-berlin.de`

mit dem Betreff „RTS Aufgabe N“ (wobei N die Nummer der jeweiligen Aufgabe ist) Ihre jeweiligen Programmquellen (*.c, *.h und Makefile; keine *.o- und ausführbare Dateien) als tar-Datei. Die tar-Datei erzeugen Sie bitte dadurch, dass Sie im entsprechenden Verzeichnis

```
tar –cvf aufgabeN.tar *
```

ausführen, wobei N auch hier die Nummer der jeweiligen Aufgabe ist.
Damit sollte es uns möglich sein festzustellen, ob eventuell Lösungen von anderen Studierenden kopiert worden sind!