Prof. Dr. Franz Schweiggert
Michael Mattes
Institut für Angewandte Informationsverarbeitung

Systemnahe Software (II) im Sommersemester 2009

Abgabetermin: 29. April 2009

Der angegebene Abgabetermin bezieht sich auf das Datum der Übungen, in denen dieses Blatt besprochen wird. Um die Übungspunkte zu erhalten, müssen Sie spätestens an diesem Datum um 16:00 Ihrem Tutor oder Ihrer Tutorin das Blatt vorgestellt haben.

Tipps zu jeder Übungsaufgabe gibt es (nur) in den Übungen.

1 Verwendung von *man* (8 Punkte)

In dieser Vorlesung werden Sie sehr oft die Manpages verschiedener Befehle und Funktionsaufrufe konsultieren müssen. Zur Wiederholung sollen Sie sich daher noch einmal kurz mit dem Manpage-System vertraut machen. Die deutsche Wikipedia liefert hierzu unter `http://de.wikipedia.org/wiki/Manpage` einige Informationen, die englischsprachige Ausgabe ist sogar noch informativer.

Konkret sollen Sie mindestens folgende Fragen beantworten können:

- Wozu werden Manpages in Abschnitte (engl. Sections) eingeteilt, und welcher Abschnitt ist für diese Vorlesung besonders wichtig?
- Wie kann die Manpage zum Systemaufruf *kill()* (nicht die zum Shellkommando kill!) unter Linux angezeigt werden? Wie geht dies bei den Sun-Rechnern des IAI, insbesondere bei Ihrem bevorzugten Login-Rechner?
- Wie kann man in einer angezeigten Manpage navigieren? Wie funktioniert die Suche nach einem Wort in der Manpage?
- Wie ist eine Manpage strukturiert?

2 Verletzt fork() Grundregeln von C? (6 Punkte)

Der Systemaufruf *fork()* wurde in der Vorlesung besprochen. Erklären Sie Ihrem Tutor, warum dieser Aufruf laut Manpage zwei Werte zurückliefert, obwohl eine Funktion doch immer nur einen Wert zurückliefern kann. Welche Bedeutung haben die jeweiligen Rückgabewerte?

3 Prozesse zählen (6 Punkte)

Schauen Sie sich folgendes Codestück an und überlegen Sie sich, wie viele Prozesse im Lauf der Verarbeitung entstehen. Wie sähe es denn aus, wenn dort statt des Wertes 5 ein anderer Wert n stünde? Begründen Sie Ihre Antworten. **Führen Sie dieses Codestück jedoch unter keinen Umständen aus!**
Codestück 1:

```
for(int i=0; i<5; ++i) fork();
```

4 Pointerarithmetik (10 Punkte)

Zuletzt sollen Sie sich zur Wiederholung noch einmal mit Zeigern (Pointern) beschäftigen und dazu etwas programmieren.

Ihr Bekannter Jürgen Fliesenmann, der Wetterexperte, bittet Sie um einen kleinen Gefallen. Er möchte Wetter-Messdaten mit Hilfe eines C-Programms auswerten. Da Jürgen auch etwas Ahnung von der C-Programmierung hat, konnte er schonmal die Datenstrukturen festlegen und an seine Dateien anpassen.

Ein Messdatenpunkt besteht aus folgenden Angaben:

- der Messzeitpunkt (als Unix-Timestamp, d.h. die Anzahl der Sekunden seit dem 01.01.1970 00:00 GMT), eine *4 Byte große positive Ganzzahl*
- die gemessene Temperatur (in Grad Celsius), ein *4 Byte großer Gleitkommawert*
- die Nummer der Messstation, eine *2 Byte große Ganzzahl*
- eine Kurzbezeichnung für die Messstation (bestehend aus zwei Zeichen), also insgesamt auch *2 Byte groß*

Die Gesamtgröße einer Messung beträgt also 12 Byte. Da Sie nicht wissen, wie viele Datensätze jeweils verarbeitet werden sollen, beschließen Sie, eine dynamische Datenstruktur zu verwenden. Und zwar eine verkettete Liste, bei der jeder Datensatz auf seinen Nachfolger zeigt.

In C wäre eine passende Datenstruktur dafür gegeben durch

```
struct messung {
        unsigned int zeit;
        float temp;
        short station;
        char kurz[2];
        struct messung *next;
}
```

Schreiben Sie nun ein Programm, welches Messungen aus der Datei “messung.dat” liest, daraus eine verkettete Liste aller Messungen aufbaut und diese auf *stdout* ausgibt. Da die Anzahl der Messungen dem Programm unbekannt ist, müssen Sie für jede Messung dynamisch Speicher allokieren. Sie benötigen mindestens die System-Calls *open*, *read* und *malloc*. Achten Sie auf passende Fehlerbehandlung!

Die Datei ist eine Binärdatei, sie enthält also einfach 12 Byte pro Messung. Sie können dem Aufruf von *read* direkt einen Zeiger auf ein `struct messung` übergeben. *Beachten Sie, dass der Zeiger auf die nachfolgende Messung nicht in der Datei enthalten ist. Diesen müssen Sie selbst korrekt setzen!* Als Ausgabeformat wäre z.B. folgendes denkbar:

```
Messung: Zeitstempel 1150495868; Temperatur 24.01; Station UT (824)
Messung: Zeitstempel 1150589261; Temperatur 0.02; Station HM (336)
Messung: Zeitstempel 1150670067; Temperatur 14.98; Station ZU (1010)
...
```

Die oben angegebenen Werte entsprechen übrigens auch den Werten in der Datei messung.dat auf der Vorlesungs-Webseite. *Dort finden Sie auch Tipps zur Aufgabe.*

Hinweise

Falls Sie nur wenig Erfahrung mit UNIX-ähnlichen Betriebssystemen haben, empfehlen wir Ihnen, sich einmal zuhause ein Linux-System näher anzuschauen. Beispielsweise gibt es von der Linux-Distribution Ubuntu eine sog. Live-CD, die Sie ohne Änderung an Ihrem PC verwenden und auf Wunsch sogar direkt von dieser CD auf Ihrem PC installieren können. Ubuntu ist unter `http://www.ubuntu.com/getubuntu/download` verfügbar.

Die Manpages zu den Systemaufrufen gibt es in einem Debian- oder Ubuntu-System unter dem Paketnamen *manpages-dev*. Weitere wichtige Pakete sind *gcc* und *make*.

Viel Erfolg!