1 Ausführung im Hintergrund

Schreiben Sie eine *Shell*-Prozedur namens *submit*, die so aufgerufen wird:

```
submit anzahl kdo {option}
```

Es soll im Hintergrund nach <*anzahl*> Sekunden das Programm oder Kommando namens <*kdo*> mit sämtlichen (beliebig vielen) Optionen oder Dateinamen ausführen, die in {option} übergeben werden, und die Standard-Ausgabe in eine Datei mit dem Namen <*kdo*>.*stdout* leiten. Anschließend soll es sich am Terminal mit einem Piepston und der Ausgabe "<*kdo*> beendet" melden.

Beispiel:

```
submit 22 wc -w testdatei
```

soll nach 22 Sekunden folgendes ausführen:

```
wc -w testdatei >wc.stdout
echo ^G wc beendet
```

2 Erzeuge Leerdateien

Erstellen Sie eine *Shell*-Prozedur namens *mkfl*, die so aufgerufen wird:

```
mkfl dname [anzahl]
```

Es soll <*anzahl*> leere Dateien namens <*dname*>*1,* <*dname*>*2,* <*dname*>*3,* ... erzeugen. Wenn der Parameter <*anzahl*> fehlt, sollen 5 derartige Dateien erzeugt werden. Hinweise:

Eine Leerdatei kann man durch eine leere Umleitung erzeugen, z. B.:

```
>leerdatei
```

3 Dateianalyse

Schreiben Sie eine *Shell*-Prozedur namens *analyse*, die so aufgerufen wird:

```
analyse fname1 fname2
```

Sie soll die Eingabe aus der Datei <*fname2*> (per Umleitung) entnehmen, zeilenweise analysieren und folgendermaßen behandeln:

- Zeilen, die mit einer Ziffer beginnen, werden verworfen.
- Die anderen Zeilen werden in der Datei <*fname1*> zeilenweise sortiert (mit *sort*) abgespeichert.
- Schließlich wird die Anzahl der Wörter, die insgesamt in den verworfenen Zeilen standen, auf *stdout* ausgegeben.

4 Beispielprozedur Interaktivität

a) Erstellen Sie unter Verwendung der *case*-Konstruktion eine Shell-Prozedur namens "menue", die mit Hilfe der *read*-Anweisung wiederholt die folgenden Kommandos entgegennimmt: *a*, *h*, *l* und *p*. Erst bei Eingabe

eines Datei-Endezeichens beendet sich "menue". Andere Eingaben werden mit einer Fehlermeldung beantwortet, ohne daß "menue" abbricht. Auf die vier korrekten Eingaben reagiert "menue" so:

a: Katalogwechsel zum Vaterkatalog,

h: Katalogwechsel zum Heimatkatalog,

l: Auflisten der Dateien im aktuellen Katalog,

p: Angabe des aktuellen Katalogs.

(Sie brauchen dazu nicht die gesamte Shell-Prozedur aus Teil a zu wiederholen, sondern nur den hinzuzufügenden Teil hinzuschreiben.)

b) Erweitern Sie "menue", indem Sie das Kommando c hinzufügen. Auf das Kommando c hin soll "menue" zunächst zur Eingabe eines Katalognamens auffordern, diesen mit Hilfe einer zweiten read-Anweisung entgegennehmen und ihn zum neuen aktuellen Katalog machen. Falls in der zweiten read-Anweisung ein Datei-Endezeichen eingegeben wird, soll "menue" an dieser Stelle ohne Katalogwechsel abbrechen.

5 Shell-Programmierung **(Klausur: SS 2000)**

Schreiben Sie ein Shell-Programm 'suche', welches analog zu (aber **ohne (!) Verwendung** von) grep oder ähnlichen Dienstprogrammen eine oder mehrere in der Kommandozeile gegebene Dateien hinsichtlich einer gegebenen Zeichenkette durchsucht. Wird die Zeichenkette in einer Zeile gefunden, so ist der Dateiname, ein Doppelpunkt und die Zeile – in der die Zeichenkette vorkommt – auszugeben.

Der Aufruf von suche hat folgende Syntax: `suche <Zeichenkette> <Dateiname1> ...`

Es muss mindestens eine zu suchende Zeichenkette und ein Dateiname angegeben sein, sonst ist eine Fehlermeldung auszugeben.

Achten Sie bitte darauf, die gelesene Zeile auch dann noch korrekt auf dem Bildschirm auszugeben, wenn diese möglicherweise für die Shell relevante Sonderzeichen (z.B. '*') enthält!