***BETRIEBSSYSTEME***

***Teil 6 Aufgabe1***  Schreiben Sie eine Prozedur, der als Parameter ein Dateiname (relativ zum aktuellen Verzeichnis) übergeben wird. Durch die Prozedur soll ermittelt werden, ob die angegebene Datei ein Verzeichnis oder eine gewöhnliche Datei ist. Im zweiten Fall soll außerdem festgestellt werden, ob die Datei leer oder nicht lesbar ist.  
Die Ergebnisse der Abfragen sollen jeweils (über stdout) angezeigt werden. Auch der Fall, dass versehentlich kein Parameter übergeben wurde, soll berücksichtigt werden.

***if test $# -eq 0***

***then***

***echo "Parameter vergessen !!"***

***exit***

***fi***

***if test -f $1***

***then***

***if test -s $1***

***then***

***if test -r $1***

***then***

***#echo " Rückkehrcode = $? "***

***echo "$1 ist eine gewöhnliche Datei, ist lesbar und nicht leer"***

***else***

***echo "$1 ist nicht lesbar"***

***fi***

***else***

***echo "$1 ist leer"***

***fi***

***else***

***if test -d $1***

***then***

***echo "$1 ist ein Verzeichnis"***

***fi***

***fi***

***Teil 5 Aufgabe4***

Schreiben Sie eine Prozedur, die es Ihnen ermöglicht, eine Abkürzung in der Datei /glb/studi/short/abk durch die Eingabe zweier Wörter (Prozedurname, Abkürzung) abzufragen. Sichern Sie, dass keine Zeilen angezeigt werden, in denen die gesuchte Abkürzung im Erklärungstext einer anderen Abkürzung vorkommt!

**egrep "^ \*$1[^A-Za-z0-9]" /glb/studi/short/abk**

**if [ $? = 1 ]**

**then echo "Keine Abkürzung vorhanden"**

**fi**

***Teil 5 Aufgabe 1***

Stellen Sie eine Prozedur auf, die ein ausführbares Programm beliebigen Namens zur Abarbeitung bringt! Programmname und die von dem Programm zu verarbeitenden Parameter sind der Prozedur in der Kommandozeile zu übergeben. Die auf stdout erzeugten Ausgaben des Programms sollen wahlweise auf dem Bildschirm oder auf dem Drucker oder in eine Datei beliebigen Namens ausgegeben werden. Das Ausgabemedium und ggf. der Dateiname sind in einem Dialog über die Konsole vom Benutzer zu erfragen.

**echo "d@:$@" # $@ was das programm ausführt und das was wir eingeben**

**echo "d\*:$\*" # es zeigt uns alle Positionsparameter**

**echo "d#:$#" # es zeigt uns die Anzahl der Positionsparameter**

**echo "--------------------------------"**

**cmd=`$@`**

**echo "Ausgabemedium: Drucker oder Datei"**

**read DEST**

**if [ "$DEST" == "lpr" ] # es wird geprüft, ob es durch die Tastatur eingegeben wurde,ob gedruckt wer$**

**then**

**echo "drucken..."**

**echo $cmd | $DEST # hier wird es gedruckt**

**else**

**echo "Ausgabe in eine Datei" #ansonsten wird es in eine Datei geschoben**

**echo $cmd > $DEST # es wird in die Datei geschoben cmd >> was ausgeführt werden soll**

**fi**

***Teil 4 Aufgabe 4***

Geben Sie eine Kommandoprozedur proz4\_5 an, die bei ihrer Ausführung die Zugriffsrechte eines Verzeichnisses und (getrennt davon) die Zugriffsrechte der im Verzeichnis enthaltenen Dateien anzeigt. Der Name des Verzeichnisses ist der Kommandoprozedur als Parameter zu übergeben. Er soll mit einer Pfadangabe relativ zum Homeverzeichnis notiert werden.   
Testen Sie diese Prozedur mit Ihrem Homeverzeichnis, dem Unterverzeichnis uprakt und dessen Unterverzeicchnis text. Welche Ergebnisse erhalten Sie, wenn Sie im Unterverzeichnis uprakt das read- bzw. execute-Recht (vorübergehend) entfernen?

**echo "Dateirechte$1"**

**ls -l$1**

**echo "Verzeishnisrechte$1"**

**ls -ld$1**

**echo "working directory$2"**

**pwd$2**

***Teil 6 Aufgabe 2***

 Ergänzen Sie die Prozedur in der Weise, dass die Datei, sofern dies sinnvoll ist, wahlweise auf Drucker bzw. Bildschirm ausgegeben wird. Bei der Menügestaltung soll berücksichtigt werden, dass im Falle einer "falschen Eingabe" (weder Drucker noch Bildschirm ausgewählt) das Menü erneut ausgegeben wird und die Eingabe wiederholt werden kann.

**if test $# -eq 0**

**then**

**echo "Parameter vergessen !!"**

**exit**

**fi**

**if test -f $1**

**then**

**if test -s $1**

**then**

**if test -r $1**

**then**

**#echo " Rückkehrcode = $? "**

**echo "$1 ist eine gewöhnliche Datei, ist lesbar und nicht leer"**

**echo "Ausgabemedium: Drucker oder Bildschirm (d oder b)"**

**echo '(d... drucken) / (b...bildschirm)'**

**read ausgabe**

**while [ test $ausgabe != d && $ausgabe != b ]**

**do**

**echo d oder b eingeben!**

**read ausgabe**

**done**

**if test $ausgabe == d**

**then**

**echo "'$1'" wird gedruckt...**

**lp $1**

**else**

**if test $ausgabe == b**

**then**

**echo Ausgabe von "'$1'":**

**cat $1**

**else**

**echo Programm beendet**

**fi**

**fi**

**else**

**echo "$1 ist nicht lesbar"**

**fi**

**else**

**echo "$1 ist leer"**

**fi**

**else**

**if test -d $1**

**then**

**#echo "$1 existiert nicht oder ist kein Dateiname"**

**#echo "Rückkehrcode = $? "**

**echo "$1 ist ein Verzeichnis"**

**fi**

**fi**

***Teil 7 Aufgabe 1***

Schreiben Sie eine Kommandoprozedur dir zur Ausgabe des Inhalts eines Verzeichnisses. Der Verzeichnisname (absolut oder relativ zum aktuellen Verzeichnis) ist der Prozedur als Parameter zu übergeben.  
Der Ausgabe soll ein Kopf mit dem Verzeichnisnamen vorangestellt werden. Wenn kein Parameter übergeben wird, soll das aktuelle Verzeichnis und dessen Inhalt ausgegeben werden. Die Ausgabe soll bildschirmweise erfolgen (falls weniger als 25 Zeilen auszugeben sind, soll die Ausgabe mittels cat erfolgen, andernfalls mittels less).

**if test $# -eq 0**

**then**

**echo "Parameter vergessen !!"**

**exit**

**fi**

**#echo "Verzeichnisname als Parameter Übergeben"**

**echo $1 Verzeichnisname**

**LINES=`ls $1 | wc -l` # LINES ist die Zahl der Zeilen oder der Einträge**

**if [ $LINES -ge 25 ] # if LINES größer oder gleich 25 ist**

**then ls $1 | less**

**else ls $1 | cat**

**fi**

***Teil 7 Aufgabe 2***

Im Rahmen eines Kunden-Informationssystems sind mit Hilfe einer Prozedur zu einzelnen Kunden, deren Namen als Parameter (2. Parameter und folgende) übergeben werden, ergänzende Informationen (z.B. Geburtsdatum und Anschrift oder Angaben über die Zahlungsmoral) der Kundendatei (1. Parameter) hinzuzufügen.  
Die Zusatzinformationen sollen im Dialog abgefragt und jeweils als zusätzlicher Datensatz eingetragen werden.

Hinweise: Als Beispiel einer Kundendatei können Sie die Datei kunden im Verzeichnis /glb/studi benutzen. Verwenden Sie in der aufzustellenden Prozedur den zeilenorientierten Editor ed.

**kudat=$1**

**shift**

**for i**

**do**

**echo "Geben Sie Zusatzinformationen fuer den Kunden \"$i\" ein:"**

**read zusatz**

**ed $kudat<<%**

**/$i/a**

**$zusatz**

**.**

**-1**

**j**

**w**

**q**

**%**

**done**

***ANWENDUNG***

s73889@ilux150:~> . aufgabe7\_2 kunden Hagen Meier

Geben Sie Zusatzinformationen fuer den Kunden "Hagen" ein:

9.12.1980

613

Hagen, Dagobert 91/0374 Meissen

623

Geben Sie Zusatzinformationen fuer den Kunden "Meier" ein:

12.4.1980

623

Meier, Isolde 91/0676 Dresden

633

s73889@ilux150:~>

***Teil 8 Aufgabe1***

Schreiben Sie eine Shell-Prozedur, die mehrere Dateien aus dem Verzeichnis /glb/studi in ein Unterverzeichnis Ihres Home-Directories kopiert.  
Die Namen der zu kopierenden Dateien sind als Parameter zu übergeben, Ihr Unterverzeichnis soll im Dialog abgefragt werden. Sollten kein bzw. mehr als 9 Parameter an die Prozedur übergeben worden sein, ist eine Fehlermeldung auszugeben und die Prozedur zu beenden. Im Fall, daß eine zu kopierende Datei nicht in /glb/studi enthalten ist, ist ebenfalls eine Fehlermeldung auszugeben.

**echo "Zu kopierenden Dateien: $\*"**

**if test $# -eq 0 -o $# -ge 9**

**then**

**echo "Parameter falsch eingegeben!!"**

**exit**

**fi**

**echo "Wo soll die datei kopiert werden?"**

**read antwort**

**for i**

**do**

**if test -f /glb/studi/$i**

**then**

**cp /glb/studi/$i ~/$antwort**

**else**

**echo "Datei existiert nicht"**

**fi**

**done**

***Teil 8 Aufgabe 2***

Schreiben Sie eine Prozedur, die von allen gewöhnlichen Dateien eines der Prozedur als Parameter zu übergebenden Textverzeichnisses die Anzahl der Wörter ermittelt und die Dateinamen sowie (d.h. gefolgt von ...) die jeweilige Wortanzahl zeilenweise in eine einzurichtende Datei "Wortanalyse" schreibt. Der Name des Textverzeichnisses sei relativ zum aktuellen Verzeichnis.   
Hinweis: Verwenden Sie das find-Kommando!

***cd ~/$1***

***for i***

***do***

***if test -f $i***

***then***

***wc -w $i > wort***

***fi***

***done***

***find ~/uprakt -type f -print > analyse***

***cat wort analyse > wortanalyse***

***Teil 8 Aufgabe 3***

Schreiben Sie eine Prozedur del, der beliebig viele Namen von zu löschenden Dateien (Dateinamen relativ zum aktuellen Verzeichnis) als Parameter übergeben werden sollen. Die Prozedur soll die Dateien aber nicht endgültig löschen, sondern in ein vorher im Homedirectory einzurichtendes Verzeichnis .muell übertragen.  
Nach mehrmaliger Benutzung der Prozedur soll dieses Verzeichnis je nach Bedarf (z.B. am Sitzungsende) mit dem üblichen UNIX-Kommando rm geleert werden. Die Dialog-Abfrage, ob das Verzeichnis geleert werden soll, soll über ein kleines Menü erfolgen.

**echo "Zu löschenden Dateien: $\*"**

**for i**

**do mv $i ~/.muell**

**done**

**cd ~/.muell**

**verz=~/.muell**

**if [ ! "`ls -A $verz`" ];**

**then**

**echo "Verzeichnis leer"**

**else**

**echo "Soll .muell geleert werden? (j/n)"**

**read antwort**

**while test $antwort != "j" -a $antwort != "n" #-a steht für and**

**do**

**echo j oder n eingeben!**

**read antwort**

**done**

**if test $antwort == "j"**

**then echo .muell wird geleert...**

**for i**

**do**

**rm -r ~/.muell/\***

**done**

**else echo .muell wird nicht geleert**

**ls -l ~/.muell**

**fi**

**fi**

***Teil 10 Aufgabe 2***

Übertragen Sie aus der Datei stadt, die im Verzeichnis /glb/studi zu finden ist und alle deutschen Großstädte enthält, den Eintrag mit Ihrer Landeshauptstadt in Ihr eigenes Verzeichnis.

s73889@isys12:~> **awk '/^Dres/{print $1, $2, $3}' /glb/studi/stadt >aufgabe10\_2**

s73889@isys12:~> nano aufgabe10\_2

***Teil 10 Aufgabe 3***

Ermitteln Sie, wieviele Großstädte das Land Baden-Wuerttemberg hat.

s73889@isys12:~> **awk '/Baden/{print $3, $1, $2}' /glb/studi/stadt > aufgabe10\_2**

s73889@isys12:~> nano aufgabe10\_2

s73889@isys12:~> **wc -l aufgabe10\_2**

**10** aufgabe10\_2

***Teil 10 Aufgabe 5***

Ermitteln Sie die Unterschiede zwischen den Programmen unix.f und hunix.f (beide im Verzeichnis /glb/studi) und gleichen Sie unix.f an hunix.f an (in Ihrem text-Verzeichnis). Geben Sie eine Kommandoprozedur an, die die Namen von zwei Texttadeien als Parameter übergeben bekommt und die erste Datei automatisch an die zweite angleicht.

**nano aufgabe10\_5**

**echo "Geben Sie die 2 Dateinamen ein:"**

**diff ~/$1 ~/$2**

***Teil 10 Aufgabe 6***

Schreiben Sie eine Prozedur, die alle die Dateien Ihres Home-Directories (einschließlich aller Unterverzeichnisse) anzeigt, die in den letzten *n* Tagen (Anzahl *n* als Parameter übergeben!) modifiziert worden sind.

**nano aufgabe10\_6**

**echo "Geben Sie die Zahl n ein"**

**read n**

**find ~ -mtime $n**

***ANWENDUNG***

***s73889@ilux150:~> . aufgabe10\_6***

***Geben Sie die Zahl n ein***

***2***

*/u\_nfs4/iw15/s73889/.local/share/baloo/file*

*…………*

***Teil 12 Aufgabe 2***

 Im Verzeichnis /glb/studi finden Sie die Dateien kunden, aktikel und auftraege mit Datensätzen, die folgendermaßen aufgebaut sind:

kunden: Kundenname Kundennummer Wohnort  
artikel: Artikelname Artikelnummer Preis  
auftraege: Kundennummer Auftragsnummer Artikelnummer Stückzahl Liefertermin

Sortieren Sie  
a) die Datei kunden alphabetisch nach Namen  
b) die Datei artikel nach Artikelnummern  
c) die Datei auftraege erstens (erstrangig) nach Kundennummern und zweitens (zweitrangig) nach dem Liefertermin

**s73889@ilux150:~> cp /glb/studi/kunden directory/kunden**

**s73889@ilux150:~> cp /glb/studi/artikel irectory/artikel**

**s73889@ilux150:~> cp /glb/studi/auftraege directory/auftraege**

**s73889@ilux150:~/directory> sort -d +0 -1 kunden #alphabetisch sortiert**

**s73889@ilux150:~/directory> sort +1 -2 artikel**

**s73889@ilux150:~/directory> sort -n +0 -1 +4 auftraege #erstrangig und zweitrangig**

***Teil 12 Aufgabe 3***

Stellen Sie eine Datei kundenauftraege für alle Kunden auf, die einen Auftrag erteilt haben. Sie enthalte: Kundennr., Name, Ort, Artikelnr., Preis, Stückzahl und Auftragsnr..

**s73889@ilux150:~/directory> sort +2 -3 -o skunden kunden**

**s73889@ilux150:~/directory> sort +0 -1 -o sauftraege auftraege**

**s73889@ilux150:~/directory> join -j1 3 -j2 1 -o 1.3 1.1 1.4 2.3 2.4 2.2 skunden sauftraege > kundenauftraege\_1**

**s73889@ilux150:~/directory> sort +1 -2 -o sartikel artikel**

**s73889@ilux150:~/directory> sort +2 -3 -o skundenauftraege1 kundenauftraege\_1**

**s73889@ilux150:~/directory> join -j1 2 -j2 4 -o 1.1 1.2 1.3 1.4 2.3 1.5 1.6 sartikel skundenauftraege > kundenauftraege**

***Teil 12 Aufgabe 4***

Sortieren Sie die Datei kundenauftraege  
a) für den Versandt nach Orten,  
b) für die Abrechnung nach Kunden  
c) für die Lagerhaltung nach Artikelnummern.

**s73889@ilux150:~/directory> sort -d +3 -4 kundenauftraege**

**s73889@ilux150:~/directory> sort -n +4 -5 kundenauftraege**

**s73889@ilux150:~/directory> sort +3 -4 kundenauftraege**

***Teil 12 Aufgabe 5***

 Ordnen Sie die Datei Kundenaufträge so, daß bei (a) die Ortsnamen, bei (b) die Kundennamen und bei (c) die Artikelnummern zuerst ausgegeben werden.

**s73889@ilux150:~/directory> awk '{print $3, $1, $2, $4, $5, $6, $7}' kundenauftraege > kundenauftraege\_ort**

**s73889@ilux150:~/directory> awk '{print $4, $1, $2, $3, $5, $6, $7}' kundenauftraege > kundenauftraege\_nummer**

**s73889@ilux150:~/directory> awk '{print $2, $1, $3, $4, $5, $6, $7}' kundenauftraege > kundenauftraege\_name**

***Teil 12 Aufgabe 6***

Die Datei stadt im Verzeichnis /glb/studi enthält alle deutschen Großstädte. Ermitteln Sie daraus die Städte mit mehr als einer halben Million Einwohner. Geben Sie die gefundenen Datensätze in der Reihenfolge   
Einwohnerzahl - Städtename - Bundesland   
aus und sortieren Sie die Städte der Größe nach!

**s73889@ilux150:~/uprakt> awk '$2 >= 500000 {print $2, $1, $3}' stadt | sort -n +1 -2 > stadt\_1**