

Aufgabe 1: Erläutern Sie detailliert das nachfolgende Kommando (Einzelbefehle, Optionen, Zusammenspiel) (20 Punkte):

```
cd /usr
```

```
cd bin
```

Wechsel ins /usr (absolut), dann von dort nach bin (relativ) – also insgesamt nach /usr/bin

```
ls -l --inode [aeimotu]*.??? |
```

Listen aller Files im Akt. Verzeichnis (ausführlich, mit i-Nr.), die mit einem Kleinbuchstaben a oder e oder i usw. beginnen, dann beliebige Zeichen haben, dann nen . und eine Endung aus 3 Zeichen. Das Ergebnis (stdout) wandert über eine Pipe ins nächste Kommando (stdin):

```
sort -r | tee -a /tmp/t2 |
```

Und wird dort rückwärts zeilenweise sortiert. Das Ergebnis wird in einem Abzweig an eine Datei t2 im Temp-Verzeichnis angehängt.

Und:

```
grep 'e' | less
```

alle Zeilen rausgefiltert, die ein e enthalten. Diese werden dann seitenweise am Bildschirm angezeigt.

Aufgabe 2: Was geben folgende Kommandos am Bildschirm aus:

```
echo "\$TERM"
$TERM
```

```
ls /
Dateinamen im root-Verzeichnis:
bin boot dev etc home lib media mnt opt proc root sbin srv
sys tmp usr var
```

```
date '+Nun ist es: %T'
Nun ist es: 12:33:00
```

```
grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

```
find /etc -name '*.conf' -ls
...
1050674 4 -rw-r--r-- 1 root root 391 Apr 13 2010 /etc/ant.conf
...
find: "/etc/webmin/openslp": Keine Berechtigung
...
Dateien in /etc ausführlich gelistet, die mit
.conf enden;
UND Fehlermeldungen wegen Berechtigungen!
```

(10 Punkte)

Aufgabe 3: Erklären Sie folgende Begriffe (und machen Sie wenn irgend möglich ein Beispiel bzw. illustrieren Sie Ihre Lösung) (16 Punkte):

i-Liste: Bereich des Dateisystems in einer Partition, die für jede Datei unter der i_Nr. einen Eintrag enthält (i-Node) mit Typ, Rechten, Besitzer, Größe, Modifikationsdatum und Adressen (ext2: 13)
ls -li

PPID: ParentProzessID, Elternprozeßnummer, der Prozeß, der den Prozeß gestartet hat, der ihm seine Umgebung gesetzt hat, der den exit-Code bekommt, ps -l; env; echo \$?

alias: Alternativname für Kommando, in der Shell gespeichert, ansonsten in Startup-Datei neu setzen (~/.bashrc),
alias l='ls -l', alias, unalias l, Starten mit l

Prozeßstatus: Was macht der Prozeß gerade (ps -l, top), Stati: laufend(R)(in der CPU oder knapp davor), wartend(W)(auf andere Prozesse, Ressourcen), schlafen(S), Zombie(Z)(anderer noch nicht fertig), gestoppt(T)(^Z,bg)

Aufgabe 4: Erstellen Sie ein Shell-Skript 'suche', das folgende Aufgaben löst und bringen Sie es zum Laufen (20 Punkte):

- als Parameter sollen ein Verzeichnis und ein Dateiname übergeben werden: `suche /home gedicht` (prüfen, ob 2 Parameter)
- dann soll nach dem Dateinamen ab dem Verzeichnis gesucht werden (Fehlermeldungen beim Suchen wegen magelnder Rechte ausblenden)
- wenn nichts gefunden, Meldung
- ansonsten bei Problemen Meldung an `stderr`
- sonst Ausgabe der gefundenen Datei mit Pfadangabe

```
#!/bin/sh
# sucht nach Dateien ab Verzeichnis
# usage: suche verz file

err=0
tmp=/tmp/erg

if [ $# = 2 ]
then
    if [ -d "$1" ]
    then
        find "$1" -name "$2" 2> /dev/null > $tmp
        if [ $(wc -l < $tmp) -gt 0 ]
        then
            cat $tmp
        else
            echo "nichts gefunden" >&2
            err=3
        fi
    else
        echo "Verzeichnis $1 existiert nicht" >&2
        err=2
    fi
else
    echo "usage: $0 verz file" >&2
    err=1
fi

exit $err

zum Laufen bringen:
chmod a+rx suche
echo $PATH # überprüfen und ggf. erweitern (PATH=$PATH:...)
suche /home passwd
./suche ... # dann ohne PATH, aber im akt. Verzeichnis suchend
sh suche ...# auch ohne PATH und ohne #!/bin/sh
```

Aufgabe 5: Nennen Sie Aufgaben und Kommandos im Bereich der Shell

Machen Sie jeweils dazu ein Beispiel. (24 Punkte)

Interpreter: `ls -l /etc | more`

unterschiedliche: `sh, csh, ksh, bash`

Konfigurierbar: `/etc/profile, ~/.profile, ~/.bashrc`

Umlenken der STDEin- und Ausgaben: `sort < in >> out 2>> err`

Pipe: `cat f1 f2 | sort | pr | tee f3 | lpr`

Hintergrund: `cat f1 f2 | sort | pr | tee f3 | lpr &`

`kill PID; jobs; bg; fg`

Metazeichen: `echo /usr/bin/[a-z]?*[^0-9]`

Kommandosubstitution: `PFAD='pwd'=$(pwd)`

Variable: `var='inhalt'; echo $var; set; unset var;`

`export var; env`

Schutz: `echo "*****" '$$$xxx' \$`

Skript: `#!/bin/sh ... exit 0`

Alias: `alias l="ls -l"; unalias l`

Funktionen: `fkt() { }`

History `history`

TAB-Ergänzung

`$? $! ...`

Kontrollstrukturen (Schleifen, Verzweigungen):

`for while; case if`

Beispiel:

```
while true
do
    date
done
```

Aufgabe 6: Welche Hilfemöglichkeiten gibt es bei UNIX/Linux? (10 Punkte)?

```
man kommando
man -k begriff
apropos begriff
whatis kommando
kommando -? oder --help
info kommando
/usr/share/doc/packages/*
www.tldp.org (howtos)
Mailinglisten
www.linux.org oder andere Portale (www.opensuse.org, ...)
Konfigdateien anschauen und auswerten
Quellcode versuchen zu verstehen (opensource!)
mit Kollegen reden
```