

GEORG-SIMON-OHM-Hochschule Nürnberg

Name:

Schriftliche Prüfung im Fach

Vorname:

"Betriebssystem Linux/UNIX"

Matrikel-Nr.:

Prüfungstermin: 13.10.2017

Semester:

Prüfungsdauer: 90 Minuten

Note:

Punktzahl:

Anzahl der Aufgabenblätter: 6

Maximal erreichbare Punktzahl: 100 Punkte + 10 Zusatzpunkte

Zugelassene Hilfsmittel: Skript, Mitschrift, Literatur

Aufgabensteller: Stockmayer, Lehrbeauftragter an der Fakultät efi

Prüfer: Stockmayer

Aufgabe 1: Erklären Sie folgende Begriffe (20 Punkte):

Pipe (was ist das?, wo kommt es vor?,
Beispiel?):
Verbindung zwischen 2 Kommandos: stdout -> stdin
via Pipe-Puffer (ca. 4-64k), Prozesssynchronisiert
Koppeln von Tools
|
find ... | less

Umlenken (was ist das?, wie geht es?, Beispiel?):
Ausgabe/Eingabe eines Kommandos in/von eine Datei/Gerät
nicht Bildschirm/Tatatur
find ... 2> /dev/null > erg
stdout: > >>
stdin: <
stderr: 2>

Systemcall (was ist das?, wozu wird es benötigt?):
Kernaufruf, normiert, > 1000, C-Funktion, wird von Tools
benutzt, open/close

i-Knoten, i-node (was ist das?, wozu wird er benötigt?):
Verwaltungseinheit in Dateisystem, jede Datei hat eine
i-Nr, im korresp. i-Knoten werden die Verwaltungsdaten
der Datei gespeichert (Typ, Rechte, ..., Adressen)
ls -i

Parent-Prozess (was ist das?, wozu muss man ihn kennen?,
wie findet man ihn?):
Eltern-Prozess, der diesen Prozess erzeugt hat, der die
Umgebung einstellt (env, export, akt. Verz.)
PPID, ps -l

Aufgabe 2: Schreiben Sie ein Shell-Skript, das alle 2 Minuten die Uhrzeit und die Prozesse des Benutzers 'otto' anzeigt. Geben Sie ALLE Schritte an, wie Sie zu dem Skript kommen und wie Sie es im Hintergrund zum Laufen bringen. (15 Punkte)

```
#!/bin/sh
# Uhrzeit, Prozesse
while sleep 120
do
    date +%T
    ps -u otto
done
# exit nicht notwendig, da endlos

chmod a+rx uhr.sh
PATH=$PATH:. # oder das Verz., wo uhr.sh liegt
uhr.sh &
./uhr.sh &
sh uhr.sh &
```

Aufgabe 3: Erklären Sie den folgenden Befehl detailliert (15 Punkte):

```
cd /usr/share/doc/HTML
gehe ins Verzeichnis /usr/share/doc/HTML

for dir in *                # alle Dateien in diesem
                           # Verzeichnis in Variable dir
                           # holen
do
    ls -lisa $dir          # und dann ausführlich mit i-Nr,
                           # Speicherverbrauch und versteckte
                           # Dateien listen
done |                     # dies alles weiterreichen via
                           # Pipe in nächstes Kommando
                           # (stdout -> stdin)

tee /tmp/listings |        # verzweigen und speichern in
                           # Datei listings im
                           # Verzeichnis /tmp

sort -r |                  # und rückwärts sortieren

tee -a /tmp/listings2 |   # verzweigen und anhängen an
                           # /tmp/listings2

lp &                      # via Defaultdrucker drucken,
                           # die gesamte Pipekette im
                           # Hintergrund
```

Aufgabe 4: Was ist ein Dateisystem: Aufgaben, Struktur aus Benutzer- und Festplattensicht, einige Befehle, Ansprechen von Dateien (alle Möglichkeiten)?
Speichern und Verwalten von Dateien
Benutzersicht: Baumstruktur (Bild)
Festplattensicht: Partition mit Boot/Super/i-List/Datenbereich, montierbar(mount)
ls, df, du, cp/mv/lm, cd, pwd, mkdir, touch
cat/less/more/vi

Was machen insbesondere folgende Befehle:

```
cp a* /tmp # kopiert alle Dateien, die mit a beginnen nach /tmp
```

```
mv a* /tmp # verschiebt alle Dateien, die mit a beginnen nach /tmp
```

```
ln a* /tmp # hardlinked alle Dateien, die mit a beginnen nach /tmp  
# muss auf gleichem Dateisystem liegen
```

(20 Punkte)

Aufgabe 5: Was ist falsch oder fehlt in folgendem Shell-Skript:

```
#!/bin/sh #3P
# sucht nach Files ab einem Startverzeichnis #1P

echo -n 'bitte Suchfilename: ' #1P
read file #1P
echo -n 'bitte Startverzeichnis: ' #1P
read dir #1P

find $dir -name $file 2> /dev/null > ~otto/erg #6P

exit $? #1P
```

(15 Punkte)

Aufgabe 6: Skizzieren Sie die Aufgaben, Kommandos, Mechanismen der UNIX-Prozessverwaltung und machen Sie jeweils ein Beispiel.
(15 Punkte)

Starten, killen, kontrollieren von Prozessen

Vorder/Hintergrund

Anhalten/Weiter

Kommandos:

ps Optionen -e -l -u ...

pstree

top

|

;

&

kill Option -9 (sonst -15)

killall

jobs

bg fg

^c

^z

PID, PPID

Prozess-Status: R,Z,W,S,...

Signale: 1-64

ps -el | less

find ... &

nice