Prof. Dr. Rüdiger Weis , Klausur2 Betriebssysteme, WS 17/18

Name (Druckbuchstaben) und Matrikelnummer : [redacted]

Aufgabe	1 (4)	2 (12)	3 (6)	4 (4)	5 (6)	6 (10)	Summe (42)	Note
Punkte	4	10	6	4	6	4	34	

Aufgabe 1: Dateisysteme (4 Punkte)

Wieviele MB beträgt die maximal unterstützte Partitionsgröße eines FAT16 Filesystemes bei einer Blockgröße von 8 KB. Bitte begründen Sie Ihre Antwort.

Partionsgröße in MB:

$\frac{2^{16} \cdot 8}{1024}$ 512

Aufgabe 2: Reguläre Ausdrücke (12 Punkte)

Welche Ausgabe liefert folgendes Python-Skript?

```
import re
zeichenk="2342 pts/0 00:00:03 emacs"
muster=("pts+?","[x0]+","s.+s+","[emacs]+","emacs*?","(00:)+")
for item in muster:
    print item,":",re.search(item,zeichenk).group()
```

Muster	Match
pts+?	pt
[x0]+	0
s.+s+	s/0 00:00:03 emacs
[emacs]+	s
emacs*?	emac
(00:)+	00:00:

Aufgaben 3: Aufgaben Betriebssyteme (6 Punkte)

Bitte nennen Sie die Hauptaufgaben eines Betriebssystemes.

Aufgabe 4: Ausnahmebehandlung (4 Punkte)

Welche Ausgabe liefert folgendes Python-Skript?

```
dividiere = lambda x, y : x / y
tests = ((99, 3.0), (23, 0), (255.0, 0.0))
try:
    try:
        for a, b in tests:
            try:
                print (a, b)
                print dividiere(a, b)
            finally:
                print("Division ausgefuehrt")
    except:
        print("Alle Fehler")
        raise ZeroDivisionError
except ZeroDivisionError:
     print("Nicht durch Null teilen...")
finally:
     print("Aufraeumen.")
print("Ende.")
```

Aufgabe 5: Parameter Parsen (6 Punkte)

Schreiben Sie ein Shell-Skript, welches von den beim Aufruf übergebenen Zahlen ausgibt, die gerade und höchstens 25 oder mindestens 77 sind.

Aufgabe 6: Shell-Programmierung (10 Punkte)

Schreiben Sie ein möglichst effektives Shell-Skript, welches den größten gemeinsamen Primteiler von drei als Parameter übergebenen Zahlen ausgibt. Bitte verwenden Sie für die Lösung der Aufgabe nur Befehle der Standard-Shell.

- Bearbeitungszeit: 90 Minuten, Viel Erfolg!

[redacted]

3. Aufgaben eines Betriebssystems:

- virtuelle Maschiene
- abstraktion komplexer Aufgaben
- User - Interface
- Ressourcen Management

4.

(99, 3.0)
33,0
"Division ausgeführt"
(23, 0)
"Division ausgeführt"
"Alle Fehler"
"Nicht durch Null teilen"
"Aufraumen."
"Ende"

5.

```
#!/bin/bash

for par in $@
do
    if [ $(expr $par % 2) -eq 0 -a $par -le 25 -o $par -ge 77 ]
        echo $par
    fi
done
exit
```

6.

```
#!/bin/bash
  fl=0    # success Nacy
  if [ $1 -le $2 ]
  then
      if [ $1 -le $3 ]
      then
          p=$1
      else
          p=$3
      fi
  else
      if [ $2 -le $3 ]
      then
          p=$2
      else
          p=$3
      fi
  fi
```

```
while [ $p -ge 2 ]
do
  ~~if [ $(expr $1 % $p) -eq 0~~
  if [ $(expr $1 % $p) -eq 0 -a $(expr $2 % $p) -eq 0
       -a $(expr $3 % $p) -eq 0 ]
  then
     n=1
     ~~while [ $p -ge $n ]~~
     ~~do~~
     ~~if [ $(expr $p % $n) -eq 0 ]~~
     ~~break~~
     ~~else~~
     ~~n=$(expr $n + 1)~~
     ~~if [ $p -eq $n ]~~
     ~~then~~
     ~~n=$(expr $n + 1)~~
     ~~if [ $~~
     while [ $p -ge $n ]
     do
        n=$(expr $n + 1)
        ~~if [ $(expr $p -ge $n) ]~~
        ~~then~~
        if [ $(expr $p % $n) -eq 0 ]
        then
           break
        else
           if [ $p -eq $n ]
           then
              fl=1
           fi
        fi
     done
     if [ fl -eq 1 ]
        echo $p ist groesster Primteiler
        exit
     fi
  else
     p=$(expr $p - 1)
  fi
done
echo kein Primteiler gefunden
exit
```

2/2