[redacted]

Prof. Dr. Rüdiger Weis

| Aufgabe | 1 | 2 | 3 | 4 | 5 | 6 | Σ | Note |
|---|---|---|---|---|---|---|---|---|
| Punkte | 4 | 10 | 3 | 5 | 6 | 7 | 35 | 1,3 |

# Betriebssysteme, Wintersemester 2016/2017

Name, Vorname, **Matrikel-Nummer:**

[redacted]

## Aufgabe 1: Dateisysteme (4 Punkte)

Wieviel MB beträgt die maximal unterstützte Partitionsgröße eines FAT16 Filesystemes bei einer Blockgröße von 4 KB. Bitte begründen Sie Ihre Antwort.

256 MB ✓ 4

## Aufgabe 2 Reguläre Ausdrücke (12P)

Welche Ausgabe liefert folgendes Python-Skript?

```
import re
d = "-rwxr-xr-x 1 root root 92712 2011-02-23 04:34 /usr/bin/man"
m = ("[xrw]+", "-[wxr]*?", "r+o+[ot]*?", "1+1-0*2?", "1+1*1??", "oo??o?")
for item in m:
    print item, ":", re.search(item, d).group()
```

| | |
|---|---|
| [xrw]+ | : 'rwxr' ✓ |
| -[wxr]*? | : '-' ✓ |
| r+o+[ot]*? | : 'roo' ✓ |
| 1+1-0*2? | : '11-02' ✓ |
| 1+1*1?? | : '11' ✓ |
| oo??o? | : |

10

[redacted]

1)

Fat 16 $2^{16}$ Blöcke

4 KB $2^{12}$ Byte

$2^{12} \cdot 2^{16} = 2^{28}$ KB

$= 2^{18}$ MB

$= 256$ MB ✓

| | |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 |
| 7 | 128 |
| 8 | 256 |

3)

~~666.0 3~~
~~Division ausgeführt~~

666.0 3
222,0 ✓ 3
Division ausgeführt
69 '0'
Division ausgeführt ✓
Allgemeiner Fehler
Nicht durch Null teilen...
Aufraeumen.
Ende.

[redacted]

4)

Bei Semaphoren handelt es sich um Sperren, die den Zugang zu einem kritischen Bereich regeln. 5
Diese Schranke wird häufig durch einen Zähler realisiert.
Will ein Prozess in die kritische Phase eintreten, so muss er schauen, ob der Zähler noch einen Platz anzeigt.
↳ Funktion Probieren. (Schauen, ob der Zähler 0 oder größer ist)
Wenn ein Prozess den kritischen Bereich verlässt, muss er die Schranke für nachfolgende Prozesse wieder öffnen
Freigeben (Den Zähler auf den Ausgangszustand setzen)

5)

```
#! /bin/sh

for i in $@
do
    if [ $i -lt 42 -a $(expr $i % 2) -eq 0 -o
         $i -ge 23 ]
    then
        echo $i
    fi
done
```

6

6)

| | |
|---|---|
| 35 | 75 |
| 75 | 35 |
| 40 | 35 |
| 15 | 35 |
| 15 | 20 |
| 5 | 5 |
| 10 | 5 |
| 5 | 5 |

14/1

```
#! /bin/sh

z1=$1
z2=$2

while [ $z1 -ne $z2 ]
do
    if [ $z1 -gt $z2 ]
    then
        z1=$(expr $z1 - $z2)
    else
        z2=$(expr $z2 - $z1)
    fi
done
ggt=$z1
(*)
#weiter auf Seite 4!
(*)
if [ $ggt -eq 1 ]
then exit 1
fi
```

(3)

Blatt 4!

```
whilec=$ggt
while [ $whilec -ne 1 ]
do
    if [ $(expr $ggt % $whilec) -eq 0 ]
    then
    forc=1
        while [ $forc -ne $(expr $whilec - 1) ]
        do
            [illegible]
            [illegible]  (*)
            if [ $(expr $whilec / $forc) -eq 0 ]

            then
                break
            fi
            echo $whilec
            exit 0
        done
        whilec=$(expr $whilec - 1)

done
```

(*) forc=$(expr $forc + 1)

[redacted]

④