

Klausur „Systemprogrammierung“ 01. Februar 2017

| | | |
|---------------|------------|-------|
| Name: | [REDACTED] | |
| Vorname: | [REDACTED] | |
| Matrikel-Nr.: | [REDACTED] | |
| Studienfach: | [REDACTED] | |

Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit (ein Deckblatt und Aufgabenblätter mit den Seitennummern 2–10)
2. Die Klausur dauert **90 Minuten**.
3. Alle Aufgaben sind auf dem jeweils zugehörigen Aufgabenblatt zu bearbeiten; Rückseiten können bei Bedarf verwendet werden.
4. Die Klausur ist komplett (mit Deckblatt und allen Aufgabenblättern) abzugeben.
5. Zugelassene **Hilfsmittel**: Keine.
6. **Unterschreiben Sie die letzte Seite der Klausur!**

| | Punkte | erreicht |
|-----------|--------|----------|
| Aufgabe 1 | 20 | 19 |
| Aufgabe 2 | 10 | 10 |
| Aufgabe 3 | 20 | 20 |
| Aufgabe 4 | 25 | 22 |
| Aufgabe 5 | 15 | 14 |
| Gesamt | 90 | 85 |

110

Name: [redacted] Matrikelnummer: [redacted]

Aufgabe 1 (Bash Programmieraufgabe)

20 Punkte

Schreiben Sie ein Bash-Skript `sum.sh` welches den Summe der Kommandozeilenparameter berechnet und ausgibt.

Beispiel:

```
$ ./sum.sh 5 3 2  
10
```

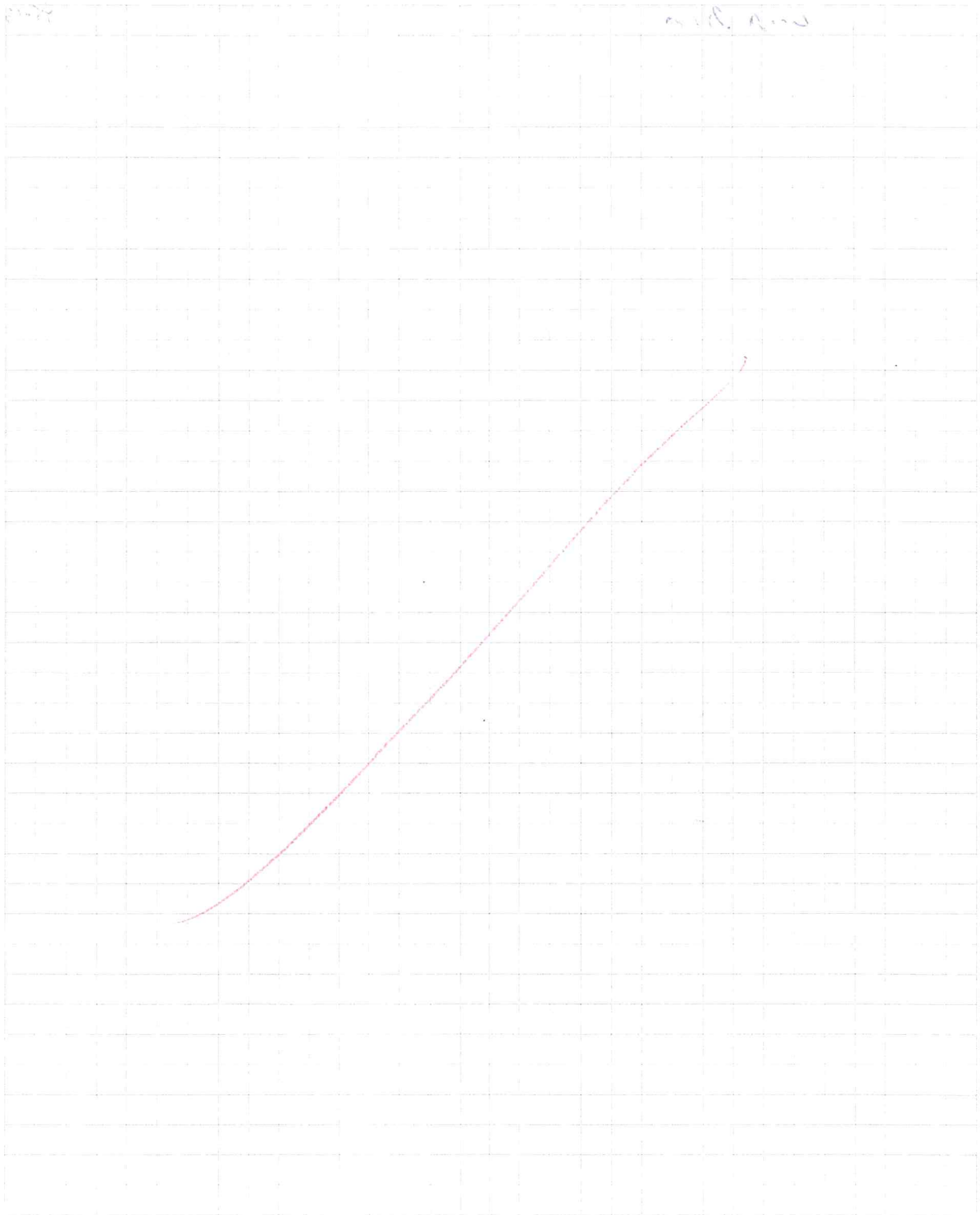
```
#!/bin/bash  
#!/bin/bash  
IF [ $# 0 ]  
THEN  
    return -1  
FI  
sum=0  
for i in $@  
do  
    sum=$((sum + $i))  
done  
echo "$sum"  
return 0
```

19/20

Zusätzlicher Platz zur Bearbeitung der Aufgabe

01.10.2022

mit dem



Name: [redacted] Matrikelnummer: [redacted]

Aufgabe 2 (Dateideskriptor Puzzle)

10 Punkte

```
#include <sys/stat.h>

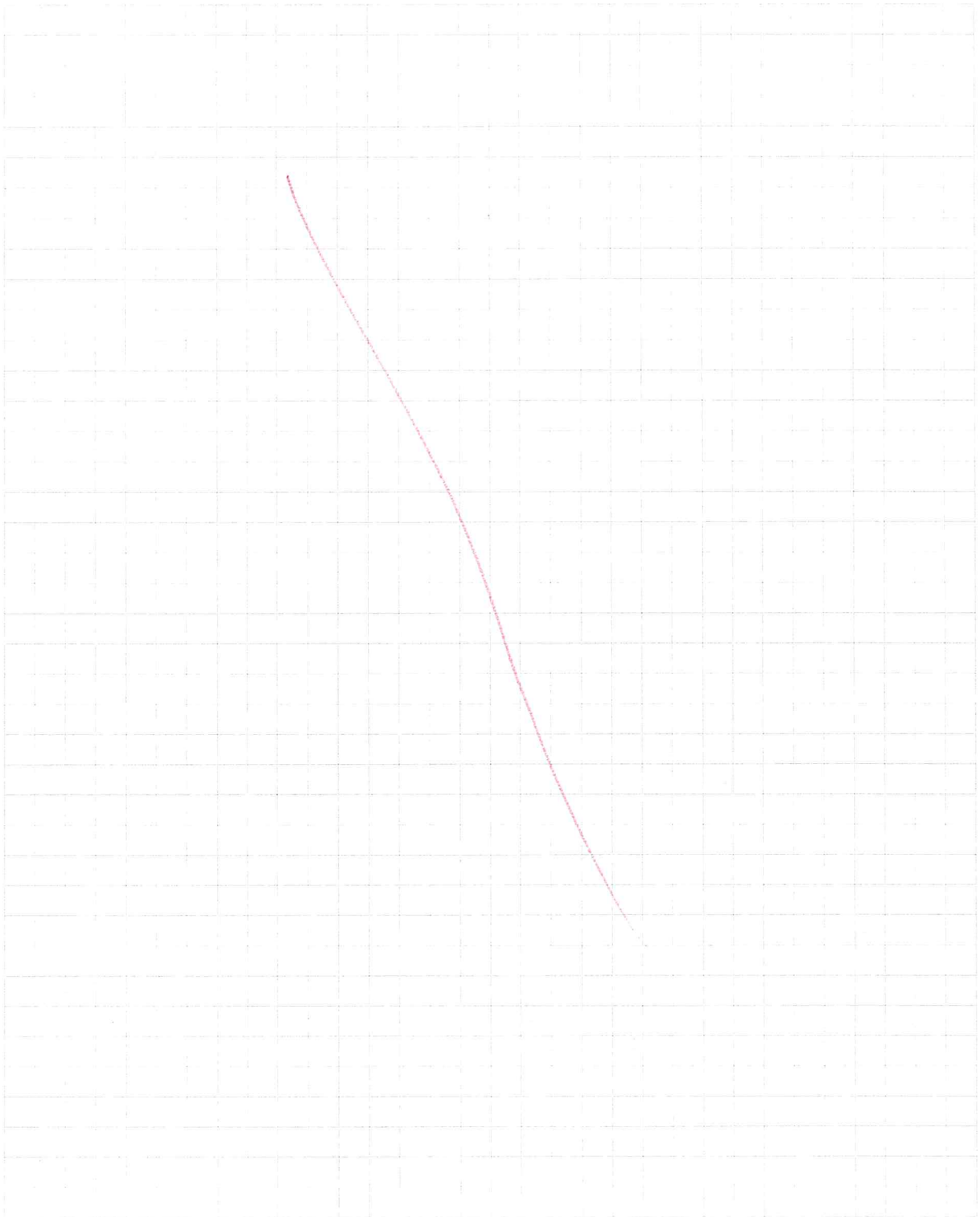
int main() {
    int fd = open("a", O_RDONLY);
    close(0);
    fd = open("b", O_RDONLY);
    fd = open("c", O_RDONLY);
    close(3);
    fd = open("d", O_RDONLY);
    close(fd);
    fd = open("d", O_RDONLY);
}
```

0
1 out
2 in
3
4

Geben Sie die jeweiligen Wert der Variable fd nach deren Zuweisung an. Sie können davon ausgehen, dass sämtliche Funktionsaufrufe erfolgreich sind.

- Zeile 4: 3 ✓
- Zeile 6: 0 ✓
- Zeile 7: 4 ✓
- Zeile 9: 3 ✓
- Zeile 11: 3 ✓

Zusätzlicher Platz zur Bearbeitung der Aufgabe



Name: [REDACTED] Matrikelnummer: [REDACTED]

Aufgabe 5 (Makefile)

15 Punkte

Schreiben Sie ein Makefile mit dem sich das Programm `sysprog_klausur.c` bauen und wieder löschen lässt.

Gehen Sie dabei von den folgenden Abhängigkeiten aus.

1. Die Datei `sysprog_klausur.c` benutzt Funktionen welche in den Dateien `beuth.c`, `klausur.c` und `sysprog.c` implementiert sind.
2. Die Quelldatei `beuth.c` verwendet Funktionen, welche in den Dateien `wedding.c` und `fb6.c` implementiert sind.

Ihr Makefile sollte mind. die 3 Targets `sysprog_klausur`, `beuth` und `clean` enthalten.

```

SKFILES = sysprog_klausur.c beuth.c klausur.c
BEFILES = wedding.c fb6.c
CC = gcc beuth.c
sysprog_klausur: $(SKFILES) $(BEFILES) -7
|<| $(CC) -o $@ $^
beuth: $(BEFILES)
|<| $(CC) -o $@ $^
clean:
|<| rm *.o ; rm beuth ; rm sysprog_klausur
    
```

15/15

Zusätzlicher Platz zur Bearbeitung der Aufgabe



Name: [redacted] Matrikelnummer: [redacted]

Aufgabe 4 (Shell-Kommando)

25 Punkte

Geben Sie ein (Verbund-)Kommando an welches ...

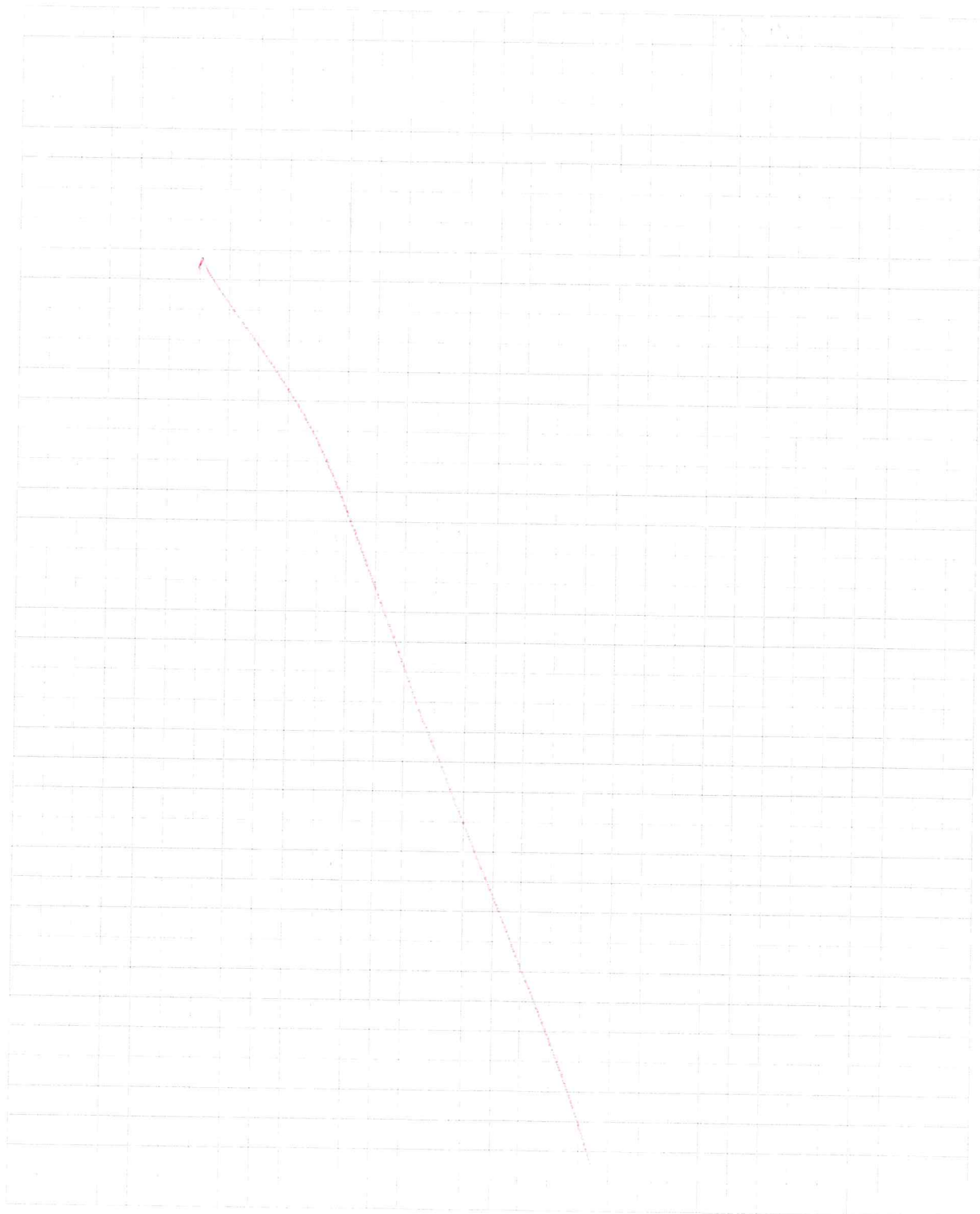
- a) ... die letzten 7 Zeilen der Datei foo.txt ausgibt.
- b) ... die Anzahl der laufenden Prozesse ausgibt.
- c) ... alle Dateien die mit g oder k enden nach /opt/backup kopiert.
- d) ... das Verzeichnis abc anlegt, falls dieses noch nicht vorhanden ist.
- e) ... bei der CSV-Datei foo.csv nur die 2 und 4 Spalte ausgibt.

(Hinweis: Die Zeile einer CSV-Datei besteht aus mehrere Werte die jeweils durch ein Komma voneinander separiert werden.)

- a) `tail foo.txt -n 7` ✓
- b) `ps -A | wc -l` ✓
CP
- c) `ls /* *[gk] | mv $? /opt/backup/` ✓
- d) `ls abc || mkdir abc` ✓
- e) `cat foo.csv | cut -d"," -f 2,4` ✓

22/25

Zusätzlicher Platz zur Bearbeitung der Aufgabe



Name: [redacted] Matrikelnummer: [redacted]

Aufgabe 3 (Linecount)

20 Punkte

Schreiben Sie ein Bash-Skript linecount.sh welches für jede Datei im aktuellen Verzeichnis die Anzahl der Zeilen bestimmt und diese zusammen mit dem Dateinamen ausgibt.

```
#!/bin/bash
let z=0
for i in $(ls)
do
  if [ ! -f "$i" ]
  then
    z=$((z - 1))
  else
    if [ $? -eq -1 ]
    then
      echo "$i : $z"
    fi
  fi
done
return 0
```

20/20