



**h\_da**

HOCHSCHULE DARMSTADT  
UNIVERSITY OF APPLIED SCIENCES

**fbi**  
FACHBEREICH INFORMATIK

RECHNERARCHITEKTUR

WS2020/21

Termin1

Laborordnung; Zahlensysteme; Einführung in Konzepte der Informatik; Einführung in eine  
Entwicklungsumgebung

| Name, Vorname | Matrikelnummer     | Anmerkungen  |
|---------------|--------------------|--------------|
|               |                    |              |
|               |                    |              |
| Datum         | Raster (z.B. Mi3x) | Testat/Datum |
|               |                    |              |

Legende: V:Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

### *Hinweise zur Laborordnung für D10/0.32*

Sinn der Laborordnung ist die Festlegung von Regeln für die Benutzung der Labore in D10.

Jeder ordentliche Student des Fachbereich ist in den jeweils ersten Veranstaltungen auf die Laborordnung hinzuweisen.

Die Laborordnung und weitere wichtige Informationen hängen beim Laboreingang.

1. Der Notausschalter des Labors D10/0.32 befindet sich über dem Lichtschalter. Der Notausschalter ist im Notfall zu betätigen, um sämtliche elektrische Geräte stromlos zu schalten. Achten Sie bei der Einführung auf die entsprechenden Hinweise.
2. Es ist nicht gestattet sich alleine im Labor aufzuhalten.
3. Im Labor ist kein Personal vorhanden. Bei Unfällen oder Unwohlsein wenden Sie sich bitte an den Betreuer.
4. Die Ersthelfer für das Gebäude sind Herr Rüdiger Raab (Raum 0.36 / Tel.: 38426), Herr Gerdion Neumann (Raum 0.37 / Tel.: 38491), Frau Bettina Kurz (Raum 0.36 / Tel.: 38453) und Herr Manfred Pester (Raum 0.33 / Tel.: 38428). Wenden Sie sich bitte im Falle einer Verletzung an einen der Ersthelfer.
5. Es ist nicht gestattet Kabel zu entfernen, Gehäuseschrauben und andere Hardware zu installieren oder sonstige Änderungen an der Laborinfrastruktur vorzunehmen. Sollte etwas nicht funktionieren, oder es wird etwas benötigt, welches die vorhandene Infrastruktur nicht abdeckt, so wenden Sie sich an den Betreuer des Labors oder direkt an den zuständigen Laboringenieur Manfred Pester (D10/0.33 / Tel.: 38428).
6. Fahren Sie die von Ihnen benutzten Geräte am Ende Ihres Praktikum/Ihrer Übung herunter und schalten diese aus, es sei denn Sie bekommen vom zuständigen Betreuer andere Anweisungen.
7. Sie sind für den Inhalt Ihrer Arbeitsplätze verantwortlich. Achten Sie auf einen ordentlichen Arbeitsplatz.
8. Bei der Benutzung des Labors ist Sorgfalt im Umgang mit der Hardware zu walten.
9. Evtl. ausgestellte Dokumentationen dienen der Laborarbeit und müssen im Raum verbleiben.
10. Die Benutzung von Mobiltelefonen ist untersagt. Schalten Sie vor dem Betreten des Raumes die Geräte (nicht nur lautlos) ab. In besonderen Fällen können Sie sich für den Notfall an den Notruf (061 1 38428) wenden.
11. Hängen Sie Ihre Kleidung (Mäntel, Jacken, ..) an die dafür vorgesehenen Kleiderständer und nicht über die Stühle.
12. Deponieren Sie Taschen, Laptops u.s.w. nicht in den Gängen, sondern möglichst an den Seiten des Labors oder unter den Tischen.
13. Verlassen Sie Ihren Arbeitsplatz aufgeräumt! Müll gehört in die vorhandenen Mülleimer, Altpapier in die dafür vorgesehene blaue Altpapierwanne.
14. Die Fluchtwege sind frei zu halten.

Bei Verstößen gegen die Laborordnung kann die Benutzungsberechtigung versagt werden.

### **Ziel der folgenden Aufgaben:**

Die ersten Aufgaben sollen Ihnen eine Einführung in einige Konzepte der Informatik mit historischen Hilfsmitteln geben. Sie bekommen hierzu ein Aufgabenblatt ausgeteilt. Mit den zur Verfügung gestellten historischen Rechenwerkzeugen, sollten Sie, mit **der richtigen Vorbereitung**, in der Lage sein alle Aufgaben zu lösen.

Auch ohne Hilfsmittel sollten wir in der Lage sein, Werte zwischen verschiedenen Zahlensystemen zu wandeln. Hierzu einige Übungen.

## **1. Wandlung zwischen Zahlensystemen**

**Beispiel: Divisionsverfahren** zur Umwandlung von Dezimalzahlen in ein anderes Zahlensystem ( b = Basis des Zahlensystems)

| <b>Division</b> | <b>Rest</b> | <b>binäre Stelle</b> |
|-----------------|-------------|----------------------|
| 71 : b = 35     | 1           | 1 * b <sup>0</sup>   |
| 35 : b = 17     | 1           | 1 * b <sup>1</sup>   |
| 17 : b = 8      | 1           | 1 * b <sup>2</sup>   |
| 8 : b = 4       | 0           | 0 * b <sup>3</sup>   |
| 4 : b = b       | 0           | 0 * b <sup>4</sup>   |
| b : b = 1       | 0           | 0 * b <sup>5</sup>   |
| 1 : b = 0       | 1           | 1 * b <sup>6</sup>   |

am Beispiel 71 von dezimal → b = 2  
Das Ergebnis ist  $1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0$  also 10001111

Übung: Umwandlung von Zahlen in andere Zahlensysteme ohne Taschenrechner

1. Dezimal → binär  
Wandeln sie die Zahl 71<sub>10</sub> in eine binäre Zahl mit Hilfe des Divisionsverfahrens
2. Dezimal → hexadezimal  
Wandeln sie die Zahl 712<sub>10</sub> in eine hexadezimale Zahl mit Hilfe des Divisionsverfahrens
3. Hexadezimal → dezimal  
Wandeln sie die Zahl FFF<sub>16</sub> in eine dezimale Zahl mit Hilfe der Exponentialdarstellung

4. Hexadezimal → binär  
Wandeln sie die Zahl  $FAB_{16}$  in eine Binäre Zahl
  
5. Binär → Hexadezimal  
Wandeln sie die Binärzahl  $1010\ 1111\ 1100\ 0011_2$  in eine Hexadezimalzahl
  
6. Binär nach dezimal  
Wandeln sie die Binärzahl  $1010\ 1111\ 1100\ 0011_2$  in eine Dezimalzahl

Für die folgenden Aufgaben gibt es eine Aufgabenblatt aus der Sammlung „Termin1\_Brief.pdf“.

## 2. Rechnen mit einer Walther Rechenmaschine

Mithilfe einer Walther Rechenmaschine sollen sie eine Addition, eine Subtraktion und eine Multiplikation durchführen. Sie lernen dabei das Konzept des Akkumulators kennen und lernen wie man eine Multiplikation auf mehrere Additionen zurückführen kann.

Sehen sie sich als Vorbereitung auch an, wie eine schriftliche Multiplikation funktioniert.

Link:

[http://www.history.didaktik.mathematik.uni-wuerzburg.de/ausstell/sammlung/sprossenrad/sprossenrad\\_interaktiv.html](http://www.history.didaktik.mathematik.uni-wuerzburg.de/ausstell/sammlung/sprossenrad/sprossenrad_interaktiv.html)

## 3. Rechnen mit dem Abakus

Mithilfe eines Abakus werden sie eine Addition und eine Subtraktion durchführen. Sie studieren dabei das Konzept des Übertrags, das für alle arithmetischen Rechnungen von hoher Bedeutung ist.

Sehen sie sich als Vorbereitung die schriftliche Addition und die schriftliche Subtraktion an und überlegen sie sich, wie diese im binären Zahlensystem aussehen könnten. Suchen sie sich eine Anleitung für das Rechnen mit einem Abakus

Link:

<https://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/code/abacus/sanpan.html>

## 4. Der Rechenschieber

Mithilfe eines Rechenschiebers führen sie eine Multiplikation und eine Division aus. Diese Aufgabe soll sie in das Rechnen mit Zahlen in Exponentialdarstellung, d. h. Zahlen die eine Mantisse und einen Exponenten besitzen, einführen. Dies ist wichtig zum Verständnis von Gleitpunktzahlen in der Informatik.

Sehen sie sich als Vorbereitung an, wie man mit einem Rechenschieber rechnet. Anleitungen findet man im Internet. Verstehen sie, wie man eine Zahl in Mantisse und Exponent aufspaltet. Verstehen sie die Funktion des Logarithmus und warum damit eine Multiplikation auf eine Addition zurückgeführt werden kann.

Im Unterordner Rechenschieber finden sich verschiedene Rechenschieber-Simulatoren. Durch Öffnen der zugehörigen html-Dateien werden diese im Web-Browser geöffnet.

Oder versuchen Sie den Link:

<https://www.arndt-bruenner.de/mathe/scripts/aristoscholar0903.htm>

## 5. Der Taschenrechner mit UPN

Die umgekehrte polnische Notation (UPN) stellt ein wichtiges Konzept in der Informatik dar. Jeder Compiler zerlegt eine komplexe Rechnung in der Form, wie Sie es in der UPN anwenden. Sie sollen in dieser Aufgabe einen ersten Kontakt mit dem Konzept eines Stack und seinen Operationen push und pop bekommen.

Vorbereitung: Die Darstellung eines komplexen algebraischen Ausdrucks mit der UPN. Das Konzept eines Stacks und die Operationen push und pop.

Im Unterordner HP-35C findet sich eine UPN-Taschenrechner-Simulation. Durch Öffnen der html-Datei kann diese mit einem Web-Browser genutzt werden. Oder versuchen Sie den Link: <https://www.hpmuseum.org/simulate/hp35sim/hp35sim.htm>

## 6. Einführung in eine Entwicklungsumgebung

Ab dem Termin 3 sollen Aufgaben gelöst werden bei denen auch programmiert werden soll. Es sollen Programme in Assembler entwickelt, geschrieben, übersetzt, gebunden und getestet werden.

Nachdem wir eine Lösung auf Papier haben, muss diese eingegeben und getestet werden. Hierzu gehen wir im Labor D10/0.32 Mikroprozessor/Mikrocontrollerlabor wie folgt vor:

In der Pandemie-Zeit nicht möglich

Schalten Sie einen der Laborrechner MI-2014-2 bis MI-2014-9 ein.

Wählen sie das Linux-Betriebssystem (wird - wenn Sie warten auch automatisch gestartet) aus.

Der gestartete Rechner bietet Ihnen einen Anmeldebildschirm an.

Melden Sie sich mit Ihrem vom der h\_da erhaltenen Namen (stxyabcd) an.

„st“ steht für Student,

„xy“ steht für die ersten Buchstaben ihres Vornamens und

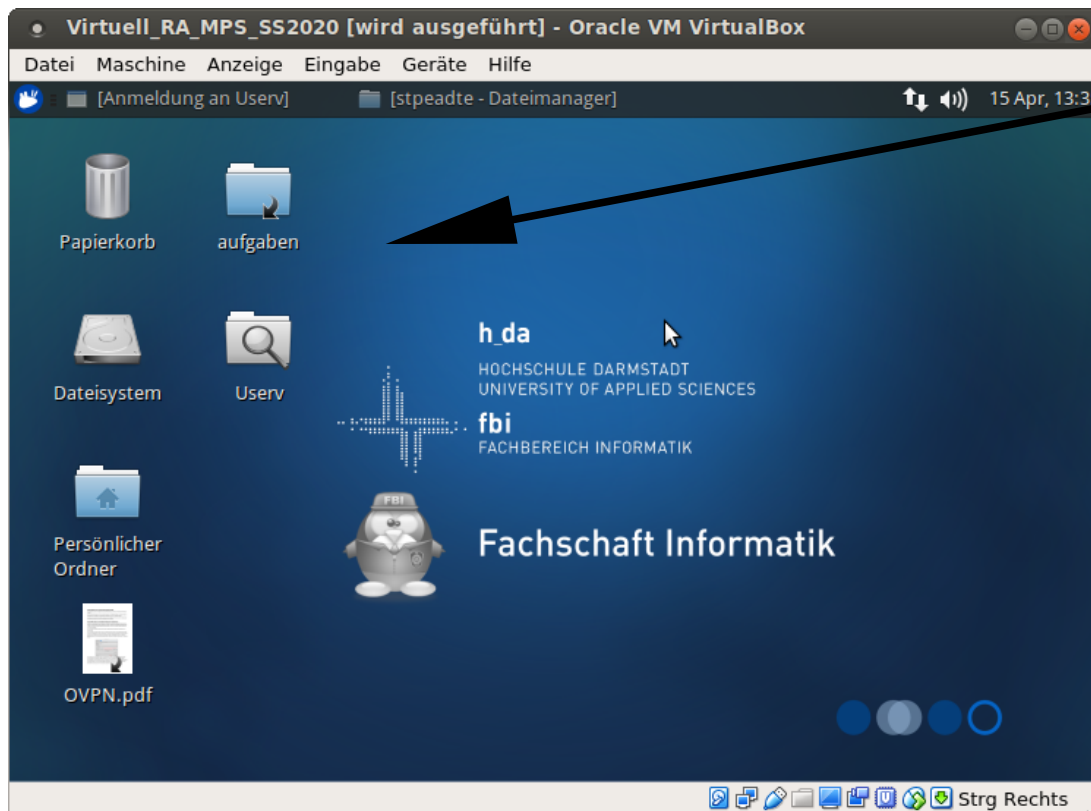
„abcd“ steht für die ersten Buchstaben Ihres Nachnamens.

Installieren und starten Sie VirtualBox mit dem Image „Virtuell\_RA\_MPS\_SS2020.ova“. Das Image gibt es auf der Seite des Mikroprozessorklabor (<https://fbi.h-da.de/studium/labore/mikroprozessor/>).

Das Passwort für den Nutzer Rechnerarchitektur ist fs.

Auf dem Desktop (Schreibtisch) im Ordner „aufgaben“ finden Sie für MPS und RA Informationen und Aufgaben. **Achtung** Inhalte der Ordner können nicht zu jeder Zeit aktuell sein.

Holen Sie sich die aktuellen Informationen und Aufgabenstellungen. Beachten Sie die Informationen in Ihrer zugehörigen Vorlesung.



Möglichkeiten der Aktualisierung der Aufgaben:

- Schauen Sie in Moodle
- Schauen Sie im Internet auf den Seiten des verantwortlichen Professors/Lehrenden.
- Melden Sie sich an Userv an (siehe Symbol auf Schreibtisch)
- ..

Mit dem Dateimanager (Thunar, Dolphin, Caja, ..), welcher beim Wechsel in Ihrem Persönlichen Ordner starten sollte, können Sie sich die für das Labor notwendigen Dateien holen um dann damit zu arbeiten. Hierzu gehen Sie z. B. wie folgt vor:

1. Wechseln Sie aus Ihrem Heimatverzeichnis in den Ordner „/home/groups/LabDisk/MI/“
2. Kopieren Sie sich den Ordner „raWS2020/21“ in Ihr Heimatverzeichnis „~“ oder besser in den Ordner /home/fs/aufgaben/. Die dort schon vorhandenen Dateien werden aktualisiert.

*Info:*

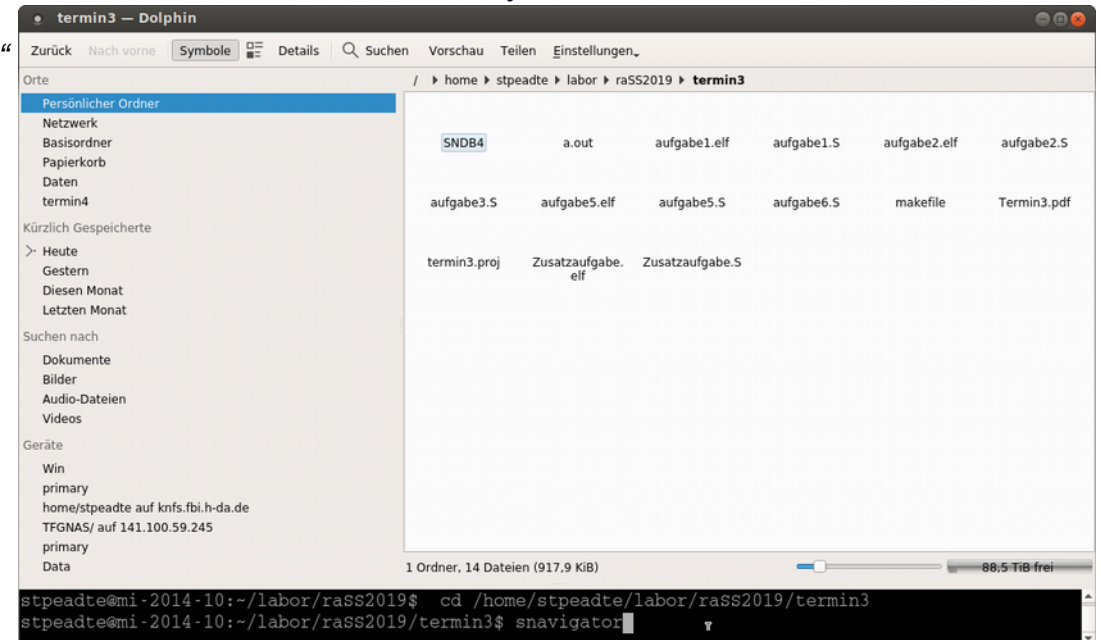
*Das Ihnen auf `userv-shell.fbi.h-da.de` zur Verfügung gestellte Heimatverzeichnis „/home/stxyabcd“ ist auch aus dem Internet wie folgt zu erreichen:*

*„[sftp://stxyabcd@userv-shell.fbi.h-da.de/home/stxyabcd](ftp://stxyabcd@userv-shell.fbi.h-da.de/home/stxyabcd)“*

*Oder Sie verbinden sich z.B. mit:*

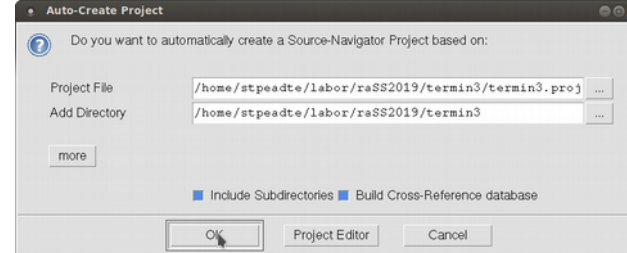
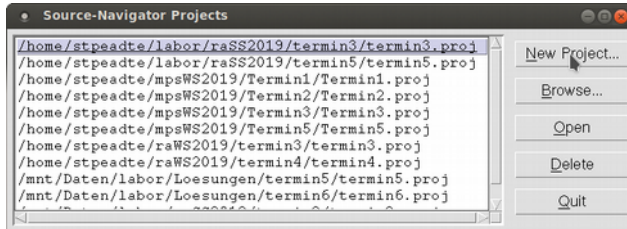
*„~\$ ssh -X [stxyabcd@userv-shell.fbi.h-da.de](mailto:stxyabcd@userv-shell.fbi.h-da.de)“*

3. Wechseln Sie nun in Ihrem Heimatverzeichnis in den Ordner des zu erledigenden Termins z. B. „aufgaben/raWS2020/21/termin3“
4. Öffnen Sie durch Drücken der rechten Maustaste (beim Dateimanager Thunar) im Fenster eine Konsole (Terminal).
5. Die Eingabeaufforderung (Prompt) in der Konsole sollte den Pfad des Projektordners, in dem gearbeitet werden soll, zeigen. Starten Sie nun die zur Verfügung gestellte Entwicklungsumgebung mit „snavigator“.

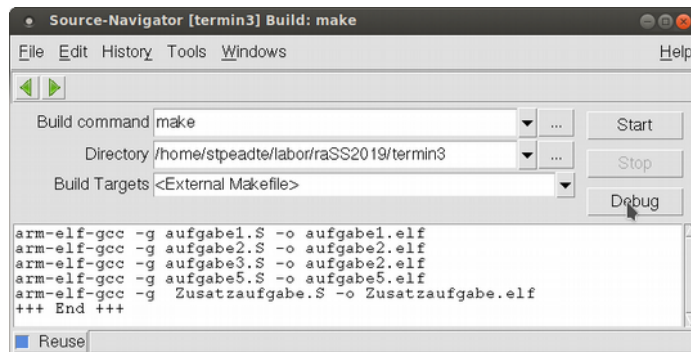




6. Legen Sie möglichst immer ein neues Projekt an und übernehmen die gegebenen Pfade.

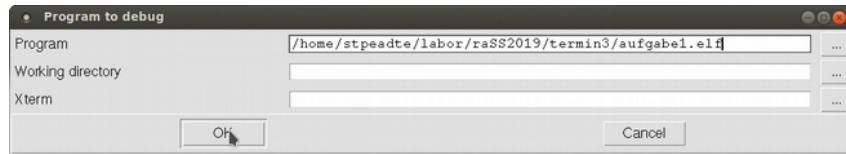


7. Im sich öffnenden Fenster Symbols können Sie durch doppeltes Anklicken einer Datei veranlassen, dass ein Editierfenster geöffnet wird.
8. Machen Sie sich auch mit dem zur Verfügung gestellten makefile vertraut.
9. Über „Tools“ „Build“ können Sie ein Build-Fenster öffnen. In diesem Fenster wird „make“ als Build command angeboten. Starten Sie make und das im Projektverzeichnis vorhandene makefile wird ausgeführt.



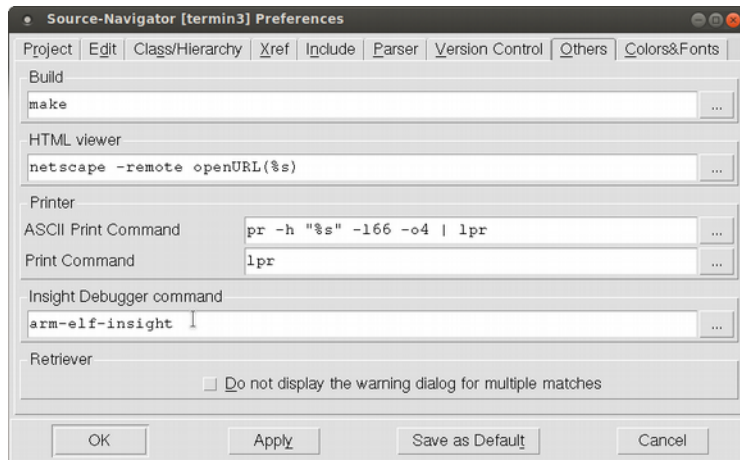
10. Im Ausgabefenster können Sie nachlesen, ob das makefile fehlerfrei abgearbeitet wurde.

11. Zum Testen eines fehlerfreien übersetzten Programms starten Sie „Debug“ und wählen die zu testende Datei.elf.



12. Sollte der Debugger (Insight) nicht starten, stellen Sie im Fenster Symbols unter File → Project Preferences → Others das richtige Insight Debugger command (arm-elf-insight oder arm-eb63-elf-insight) ein. Danach sollte Insight funktionieren.

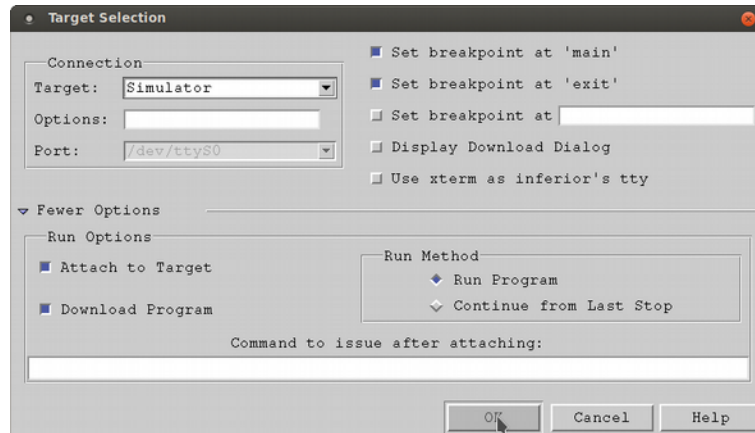
**ACHTUNG: Die Toolchain könnte auch arm-eb63-elf- sein.**



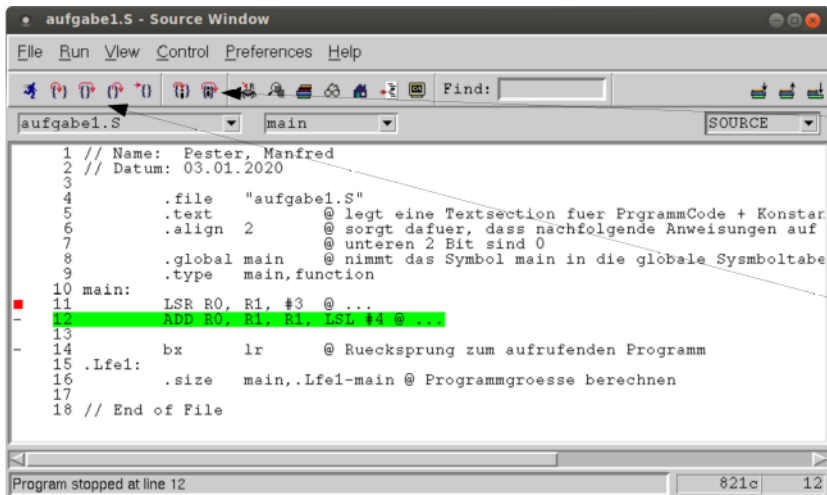
13. Drücken Sie auf den Läufer.

```
1 // Name:      Matrikelnummer:
2 // Name:      Matrikelnummer:
3 // Datum:
4
5     .file     "aufgabe1.S"
6     .text     @ legt eine Textsection fuer ProgrammCode + Konstanten an
7     .align 2   @ sorgt dafuer, dass nachfolgende Anweisungen auf einer du
8               @ unteren 2 Bit sind 0
9     .global  main @ nimmt das Symbol main in die globale Sysmboltabelle auf
10    .type    main, function
11 main:
12    LSR R0, R1, #3 @ ...
13    ADD R0, R1, R1, LSL #4 @ ...
14
15 @ Testen ob Thumb-Befehle übersetzt werden
16    push    {r1, r2}
17 @
18    bx     lr @ Ruecksprung zum aufrufenden Programm
19    mov    pc, lr @ Ruecksprung zum aufrufenden Programm
20
21 .Lfel:
22    .size   main, .Lfel-main @ Programmgroesse berechnen
23
24 // End of File
```

14. Im sich evtl. öffnenden Fenster Target Selection wählen Sie Target: Simulator. Sollte nur das erste mal auftreten.



15. Sie können nun Ihr Programm ausführen oder in Einzelschritten testen.



Assemblerbefehle werden durch Bedienen der Icon



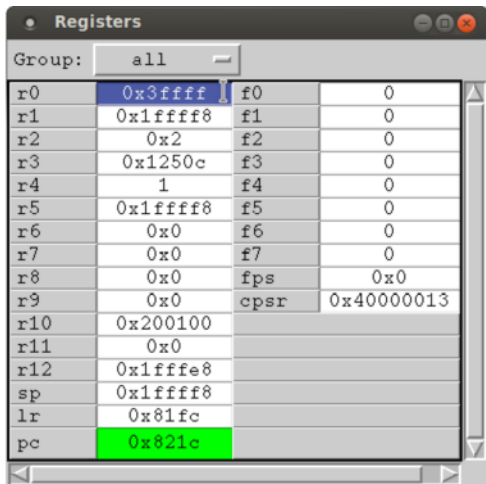
ausgelöst.

Die Icon



nur für Hochsprachendebugging verwenden.

16. Öffnen Sie ein Fenster mit den Registern um die Auswirkung der Befehle zu sehen.



*Folgen Sie den Anweisungen im Labor, machen Sie sich Notizen um ein Protokoll zu diesem Termin erstellen zu können.*

*Haben Sie den Mut und stellen Sie Fragen!*