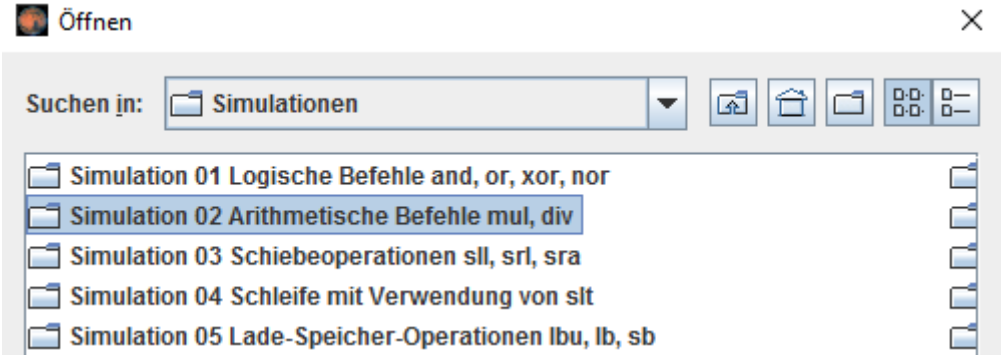
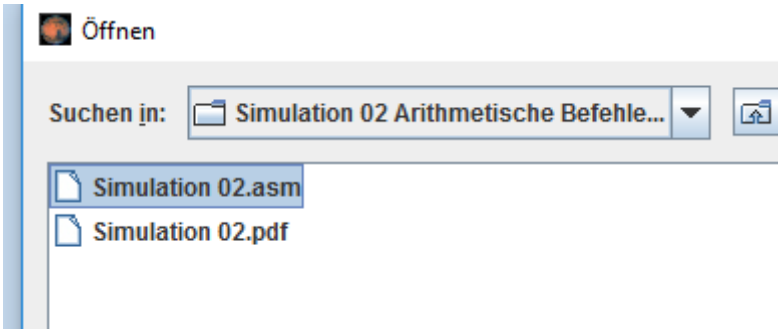
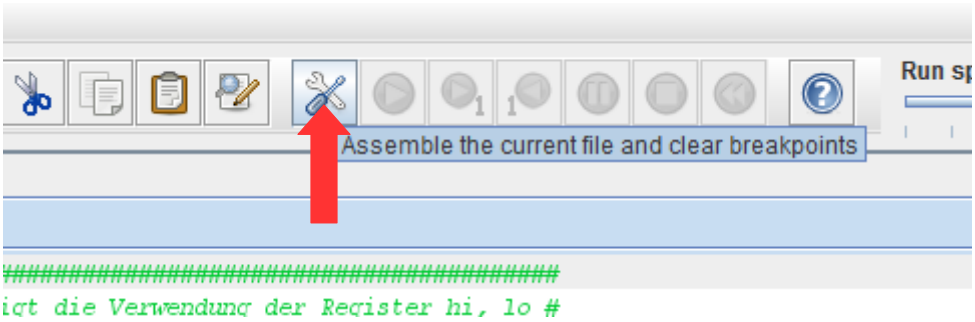
- MARS = **M**IPS **A**ssembler and **R**untime **S**imulator
  - MARS ist eine schlanke, javabasierte Entwicklungsumgebung für die Programmierung in MIPS Assembler. Es ist also zwingend erforderlich, dass ein (aktuelles) JDK/JRE (oder OpenJDK) auf Ihrem System vorhanden ist.
  - Im Ordner *Tutorials* finden Sie ein sehr ausführliches (englischsprachiges) *MARS Tutorial* der Entwickler und ebenso ein Dokument *Mars features*, das einige nützliche Tipps enthält.
  - Auf der Homepage des Simulators wird eine Vielzahl von Hilfestellungen gegeben, zu finden unter: http://courses.missouristate.edu/KenVollmar/mars/Help/MarsHelpIntro.html
  - Dieses *HowTo* hat nicht den Anspruch, Ihnen Assembler-Programmierung beizubringen, sondern soll dazu dienen, Ihr Interesse am MARS zu wecken und eventuell den Einstieg zu erleichtern.
- Erste Schritte → vorhandene Simulationen
  - über die Konsole: wechseln Sie in das Verzeichnis, in dem die .jar Datei liegt und öffnen Sie sie mit: java -jar „MARS Simulator.jar“, abhängig von Ihren Systemeinstellungen reicht eventuell ein Doppelklick.
  - Über das Menü File/Open... können Sie im Ordner MARS/Simulationen die 19 verfügbaren Simulationen finden und öffnen, in jedem der Ordner liegt sowohl die .asm Datei mit der Simulation als auch eine .pdf-Kurzbeschreibung. Es ist sinnvoll, diese pdf ebenfalls zu öffnen:

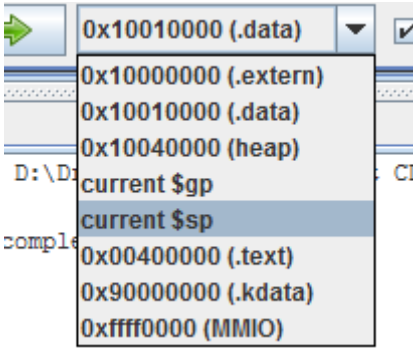




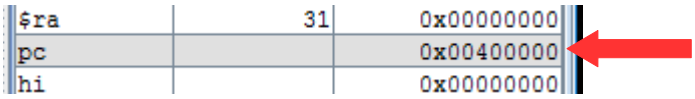
- Erste Schritte → ein Programm durchlaufen lassen
  - Programm laden, wie oben beschrieben
  - den Code assemblieren per Klick auf den *Assemble* Button



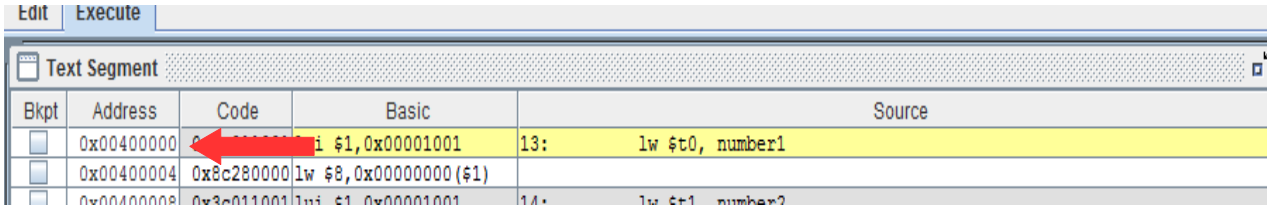
  - daraufhin wird das *Execute* Fenster angezeigt, wo im *Text Segment* die Adressen der einzelnen Befehle, sowie der Code selbst zeilenweise zu sehen sind. Das *Data Segment* zeigt die Verwendung des Speichers und ermöglicht über das Dropdown-Menü die Auswahl verschiedener Ansichten, z.B. lässt sich so die Ansicht des Stacks auswählen:



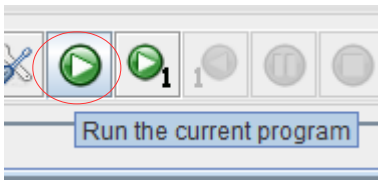
- Auf der rechten Seite findet sich die Register-Übersicht. Dort lässt sich gut verfolgen, was während des Programmablaufs passiert, wichtig ist beispielsweise der PC, der immer die Adresse des als nächstes auszuführenden Befehls enthält:

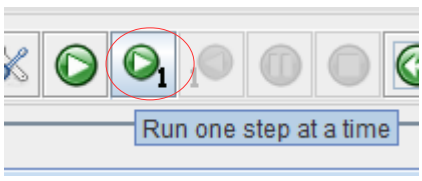


- Diese Adresse findet sich links im *Text Segment* wieder, es ist die Adresse des ersten Befehls im .text Teil unseres Beispielcodes:



- Empfehlenswert ist es, die Programme nicht über den Button *Run the current program*, sondern über *Run one step at the time* laufen zu lassen. So wird schrittweise vorgegangen, man kann die Auswertung von Bedingungen, Sprünge, Registermanipulationen usw. sehr gut nachverfolgen und damit verstehen:

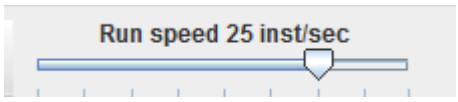




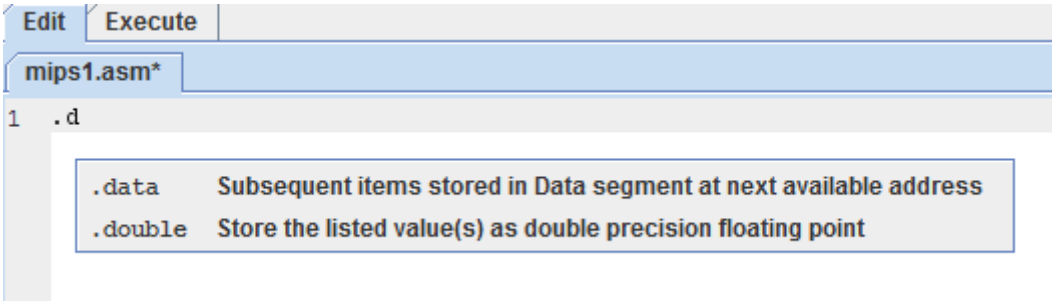
- Eine eventuell generierte Ausgabe findet sich ganz unten, im Tab *RUN I/O*, der andere Tab dient der Ausgabe der Mars messages.

• Tipps

- Unter *Run speed x inst/sec* lässt sich die Geschwindigkeit einstellen, wenn Sie beispielsweise ein Programm einfach durchlaufen lassen, aber dennoch verfolgen wollen, was passiert, senken Sie den Wert dort und können den Ablauf beobachten:



- Nutzen Sie die Vorschläge des Mars beim Eintippen von Code, sobald angefangen wurde zu tippen, schlägt Mars sinnvolle, mögliche Vervollständigungen vor:



- Ebenso gibt es zu jedem Befehl per Mouse-Over-Effekt nützliche Erläuterungen, die bei der Auswahl des richtigen Befehls helfen:

- Nutzen Sie die Kurzbeschreibungen, die jeder vorgefertigten Simulation beiliegen.

Ebenso wie beim Hades-Simulator, der für die Simulation von Schaltnetzen benutzt wird (die Reihe *Simulationen mit dem Hades-Simulator*) kann der Funktionsumfang des MARS zuerst abschrecken. Allerdings ist das hier ebenso unnötig, denn sobald man sich damit ein wenig vertraut gemacht hat, gelingt auch das selbstständige Verfassen von eigenen Programmen zügig.

Viel Spaß & Erfolg mit MARS!