

Benutzerhandbuch für die Random-Access-Maschine (RAM)

Die Random-Access-Maschine (RAM) ist eine virtuelle Maschine, die es Ihnen ermöglicht, Programme auszuführen, die auf einem bestimmten Speicher arbeiten. Sie können vordefinierte Programme auswählen oder eigene Programme erstellen und ausführen.

Vordefinierte Programme

Das RAM bietet folgende vordefinierte Programme:

1. Addition: Dieses Programm addiert zwei Zahlen.
2. Subtraktion: Dieses Programm subtrahiert zwei Zahlen.
3. Multiplikation: Dieses Programm multipliziert zwei Zahlen. (mit marvin)
4. Division: Dieses Programm dividiert zwei Zahlen.
5. Fibonacci: Dieses Programm berechnet die Fibonacci-Zahlen.

Verwendung der RAM

1. Starten Sie das RAM-Programm in User.
2. Definieren Sie die Grösse des Speichers. Beachten Sie, dass ein zu kleiner Speicher zu Fehlermeldung führen kann.
3. Wählen Sie, ob Sie einen eigenen Speicher haben möchten.
 - Wenn Sie "Ja" wählen, werden Sie aufgefordert, die Anzahl der Speicherwerte einzugeben und diese einzeln einzugeben. Zum Beispiel: "5" gefolgt von "6 3 0 1 0".
 - Wenn Sie "Nein" wählen, wird automatisch ein vordefinierter Speicher verwendet und ein zufälliges Programm ausgeführt. Anschliessend wird das Programm beendet.
4. Wählen Sie, ob Sie ein eigenes Programm haben möchten.
 - Wenn Sie "Ja" wählen, werden Sie aufgefordert, den absoluten oder relativen Pfad zu Ihrem Programm einzugeben. Beispiel: "programm/EigenesProgramm.txt".<- Hier kann man eigene Programm interpretieren.
 - Wenn Sie "Nein" wählen, können Sie eine Zahl von 1 bis 5 eingeben, um ein vordefiniertes Programm auszuführen. Beachten Sie, dass das Ergebnis möglicherweise nicht korrekt ist, wenn das Programm nicht mit dem Speicher übereinstimmt. Sie können jedoch im Programmordner "Memoryrules" nachschauen, wo alle Regeln für die Programme und deren Speicheranforderungen aufgeführt sind.
5. Das ausgewählte Programm wird geladen und ausgeführt.
6. Das Ergebnis wird angezeigt, und das Programm endet.

Befehle

Die RAM unterstützt die folgenden Befehle in den Programmen:

- **HLT**: Beendet das Programm.
- **LIDA X**: Lädt den Wert aus dem Speicher an der Position X in den Akkumulator.
- **LDI X**: Lädt den Wert aus dem Speicher an der Position X in den Akkumulator und interpretiert diesen Wert als Speicheradresse, um den endgültigen Wert zu laden.
- **STA X**: Speichert den Wert im Akkumulator an der Position X im Speicher.
- **STI X**: Interpretiert den Wert im Akkumulator als Speicheradresse und speichert den Wert an dieser Adresse im Speicher.
- **ADD X**: Addiert den Wert aus dem Speicher an der Position X zum Akkumulator.
- **SUB X**: Subtrahiert den Wert aus dem Speicher an der Position X vom Akkumulator.
- **JMP X**: Springt zur Position X im Programm.
- **JMZ X**: Springt zur Position X im Programm, wenn der Akkumulator den Wert 0 enthält.
- **ADDLDI X**: Addiert den Wert X zum Akkumulator und interpretiert diesen Wert als Speicheradresse, um den endgültigen Wert zu laden.
- Hinweis: Der Speicher wird als Array von ganzen Zahlen dargestellt, und die Positionen beginnen bei 0. **Die Befehle im Programm beginnen jedoch mit Index 1. Es wird empfohlen, den Befehl "HLT" an Index 0 zu setzen.**
-

Fehlerbehandlung

Falls während der Programmausführung ein Fehler auftritt, wird eine entsprechende Fehlermeldung angezeigt. Stellen Sie sicher, dass die eingegebenen Speicherwerte und Programmdateien den Anforderungen entsprechen.

Das Benutzerhandbuch bietet eine grundlegende Anleitung zur Verwendung der RAM. Es wird empfohlen, sich mit den verschiedenen Befehlen und Programmen vertraut zu machen, um das volle Potenzial der RAM auszuschöpfen.