

Programmierung 2 - SS19

Projekt 1 - 2048

Autoren: Stefan Oswald, Lauritz Timm

24. April 2019

Universität des Saarlandes

1. Technische Hinweise
2. How-To: Register
3. Zum Projekt

Technische Hinweise

Konfiguration

`$ git config` dient der Konfiguration von Git Repositories.

- `--global` richtet die globale Konfiguration ein
 - `user.name "Vorname Nachname"`
 - `user.email "...stud.uni-saarland.de"`

Konfiguration

`$ git config` dient der Konfiguration von Git Repositories.

- `--global` richtet die globale Konfiguration ein
 - `user.name "Vorname Nachname"`
 - `user.email "...stud.uni-saarland.de"`

Beispiel

```
$ git config --global user.name "Konrad Klug"
```

```
$ git config --global user.email "konrad@klug.de"
```

Git Projekt-Repository

Wir können das Projekt mit `$ git clone` unter folgender URL beziehen:

```
https://prog2scm.cdl.uni-saarland.de/git/project1/<NAME>
```

<NAME> = Euer Benutzername auf der Prog2-Website

Achtung!

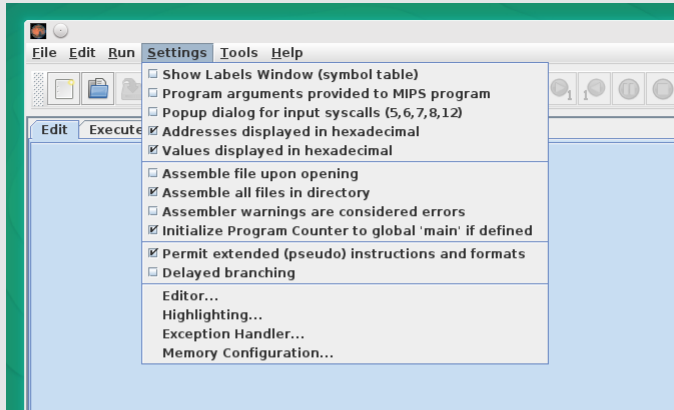
Die Repositories sind nur innerhalb des Uninetzes erreichbar. Von außerhalb kann man eine VPN-Verbindung zum Uninetz einrichten.

Eine Anleitung hierzu findet sich unter [Software](#).

MARS Einstellungen

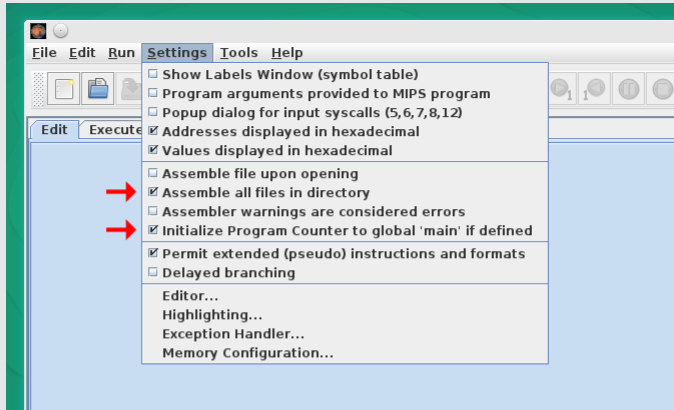
Achtung

Wir müssen zwei Einstellungen anpassen:



Achtung

Wir müssen zwei Einstellungen anpassen:



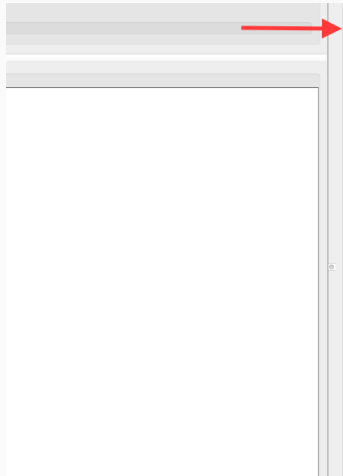
How-To: Register

Bedeutung der einzelnen Register

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffefffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention

Bedeutung der einzelnen Register

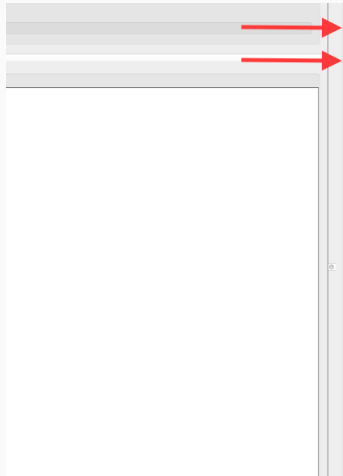


A diagram showing the MIPS register file. A red arrow points from the left towards the \$zero register in the table.

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention

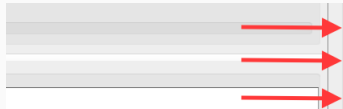
Bedeutung der einzelnen Register



Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention

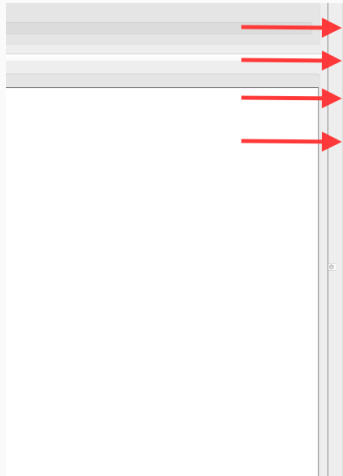
Bedeutung der einzelnen Register



Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention

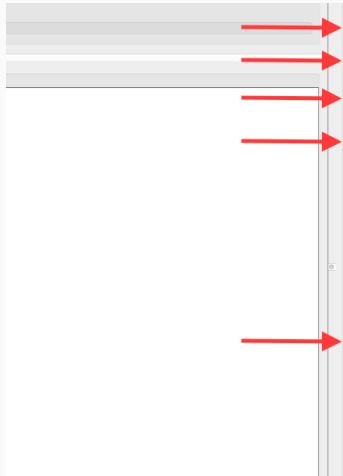
Bedeutung der einzelnen Register



Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention

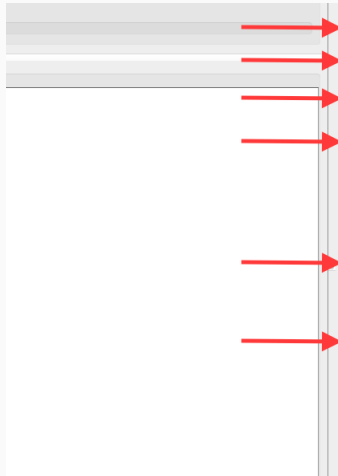
Bedeutung der einzelnen Register



Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention


Bedeutung der einzelnen Register



Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Aufrufkonvention

Bedeutung der einzelnen Register



Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffefffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000

Wir können in unserem Projektordner mit `$./run_tests.py` die Public Tests ausführen.

Wir können in unserem Projektordner mit `$./run_tests.py` die Public Tests ausführen.

Achtung!

Es müssen alle Public Tests zu einer Teilaufgabe bestanden werden, um Punkte für diese Teilaufgabe erhalten zu können.

Folgende Skripte sind gegeben:

- `run_tests.py` - führt die Public Tests aus, mit `-t <testpfad>` (ohne `.asm`) kann ein einzelner Test ausgeführt werden. Es können eigene Tests in einem `tests/student/` Ordner angelegt werden, diese werden mit `--student` ausgeführt.
- `run_gui.py` - ruft die Java-GUI auf
- `build_testbox <testpfad>` - erstellt eine Testbox für den gewählten Test.

Fragen?

Codebeispiel

Zum Projekt

Aufgabe 1

In der Datei `check.asm`:

Siegsbedingung überprüfen

- `$a0` - Adresse des Spielfelds
- `$a1` - Länge des Spielfeldes

Aufgabe 1 - Beispiel

\$a0
↓

2	4	2	4
2	4	2	4
0	0	8	16
32	64	0	8

Das Spielfeld ist eine Reihung von vorzeichenlosen Halbwörtern.

Es gilt herauszufinden ob eine 2048 auf dem Feld liegt.

Aufgabe 1 - Beispiel

\$a0

+2

2	4	2	4
2	4	2	4
0	0	8	16
32	64	0	8

Das Spielfeld ist eine Reihung von vorzeichenlosen Halbwörtern. Es gilt herauszufinden ob eine 2048 auf dem Feld liegt.

Aufgabe 1 - Beispiel

\$a0

+2

2	4	2	4
2	4	2	4
0	0	8	16
32	64	0	8

Das Spielfeld ist eine Reihung von vorzeichenlosen Halbwörtern. Es gilt herauszufinden ob eine 2048 auf dem Feld liegt.

Aufgabe 1 - Beispiel

\$a0 +2




2	4	2	4
2	4	2	4
0	0	8	16
32	64	0	8

Das Spielfeld ist eine Reihung von vorzeichenlosen Halbwörtern.

Es gilt herauszufinden ob eine 2048 auf dem Feld liegt.

Aufgabe 1 - Beispiel

\$a0
+2



2	4	2	4
2	4	2	4
0	0	8	16
32	64	0	8

Das Spielfeld ist eine Reihung von vorzeichenlosen Halbwörtern.

Es gilt herauszufinden ob eine 2048 auf dem Feld liegt.

Falls ja: gebe 1 zurück!

Spielfeld in linearer Darstellung



Die einzelnen Felder liegen direkt hintereinander im Datensegment.

Aufgabe 2

In der Datei `place.asm`:

Stein platzieren

- `$a0` - Startadresse des Spielfeldes
- `$a1` - Länge des Spielfeldes
- `$a2` - Feld Nummer, in das gesetzt werden soll
- `$a3` - Wert, der gesetzt werden soll
- `$v0` - Rückgabewert (1 oder 0)

Aufgabe 2 - Beispiel

\$a0
↓

2	4	2	4
2	4	2	4
0	0	8	16
32	64	0	8

Beispiel

- \$a1: 16
- \$a2: 8
- \$a3: 64

Aufgabe 2 - Beispiel

\$a0
↓

2	4	2	4
2	4	2	4
64	0	8	16
32	64	0	8

Beispiel

- \$a1: 16
- \$a2: 8
- \$a3: 64
- \$v0 \Rightarrow 0

Erinnerung

In allen¹ Aufgaben, die das Spielfeld zeilenweise betrachten, wird nicht die Adresse des Spielfeldes übergeben!

Der Puffer:

- `$a0` enthält die Adresse eines Puffers - `$a1` dessen Länge

¹nicht Bonus

Erinnerung

In allen Aufgaben, die das Spielfeld zeilenweise betrachten, wird nicht die Adresse des Spielfeldes übergeben!

Der Puffer:

- `$a0` enthält die Adresse eines Puffers - `$a1` dessen Länge
- Enthält *Adressen* von Spielfeldern

Erinnerung

In allen Aufgaben, die das Spielfeld zeilenweise betrachten, wird nicht die Adresse des Spielfeldes übergeben!

Der Puffer:

- `$a0` enthält die Adresse eines Puffers - `$a1` dessen Länge
- Enthält *Adressen* von Spielfeldern
- Adressen haben *word* Größe (4 Byte).

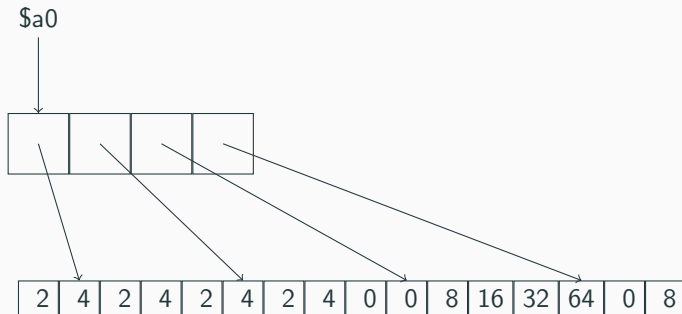
Erinnerung

In allen Aufgaben, die das Spielfeld zeilenweise betrachten, wird nicht die Adresse des Spielfeldes übergeben!

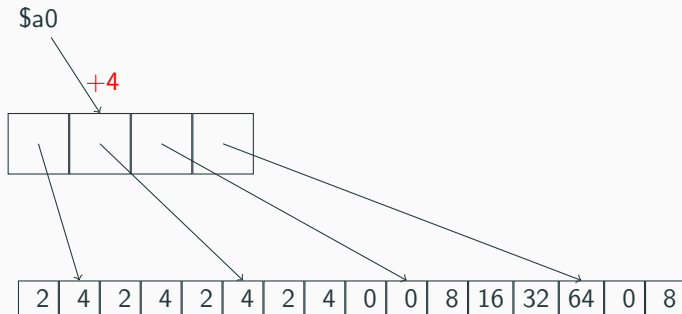
Der Puffer:

- `$a0` enthält die Adresse eines Puffers - `$a1` dessen Länge
- Enthält *Adressen* von Spielfeldern
- Adressen haben *word* Größe (4 Byte).
- Elemente sind so angeordnet, dass immer (ein move) "nach links" betrachtet werden muss.

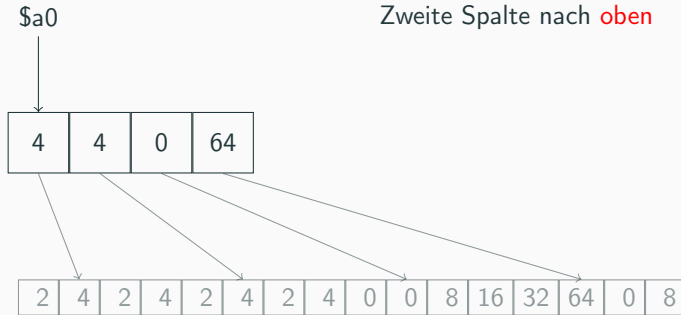
Puffer Beispiel: Zweite Spalte nach oben



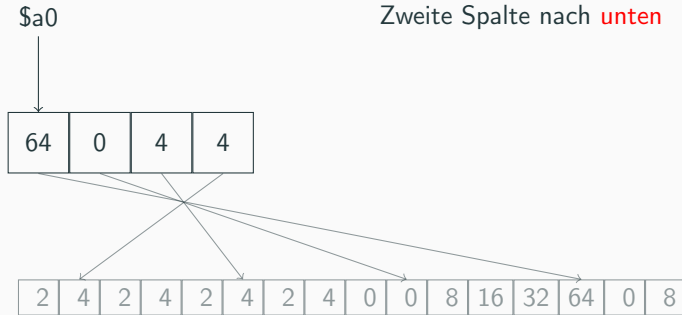
Puffer Beispiel: Zweite Spalte nach oben



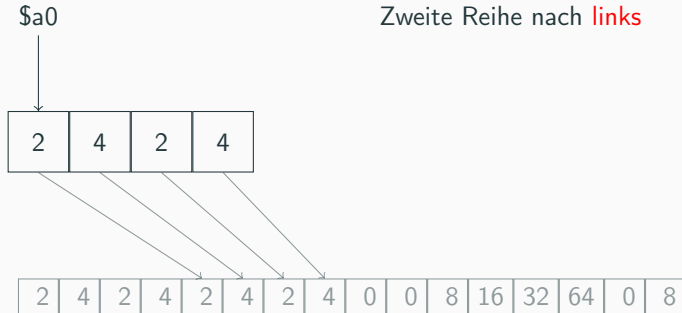
Puffer: vereinfachte Ansicht



Puffer: vereinfachte Ansicht



Puffer: vereinfachte Ansicht



Aufgabe 3

In der Datei `move_check.asm`:

Move prüfen

- `$a0` - Startadresse des Puffers
- `$a1` - Länge des Puffers
- `$v0` - Rückgabewert (1 oder 0)

Mögliche moves

0	4	16	4
---	---	----	---

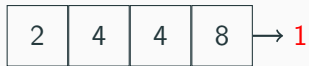
0	0	0	0
---	---	---	---

4	0	0	8
---	---	---	---

2	4	8	0
---	---	---	---

2	4	4	8
---	---	---	---

Mögliche moves



Aufgabe 4 - Spielzug ausführen

In den Dateien `move_one.asm`, `complete_move.asm`,
`merge.asm`, `move_left.asm`:

Steine um eins verschieben

- `$a0` - Startadresse des Puffers
- `$a1` - Länge des Puffers
- `$v0` - Rückgabewert (0 oder 1)

Aufgabe 4 - Spielzug ausführen

In den Dateien `move_one.asm`, `complete_move.asm`,
`merge.asm`, `move_left.asm`:

Steine um eins verschieben

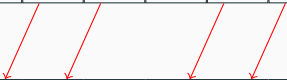
- `$a0` - Startadresse des Puffers
- `$a1` - Länge des Puffers
- `$v0` - Rückgabewert (0 oder 1)

Verschmelzen und schieben

- `$a0` - Startadresse des Puffers
- `$a1` - Länge des Puffers

Move one

0	8	16	0	4	4
---	---	----	---	---	---



8	16	0	4	4	0
---	----	---	---	---	---



8	16	4	4	0	0
---	----	---	---	---	---

8	16	4	4	0	0
---	----	---	---	---	---

Move one

0	8	16	0	4	4
---	---	----	---	---	---

8	16	0	4	4	0
---	----	---	---	---	---

8	16	4	4	0	0
---	----	---	---	---	---

8	16	4	4	0	0
---	----	---	---	---	---

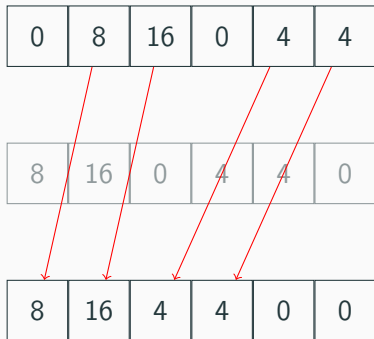
2	2	0	0	8
---	---	---	---	---

2	2	0	8	0
---	---	---	---	---

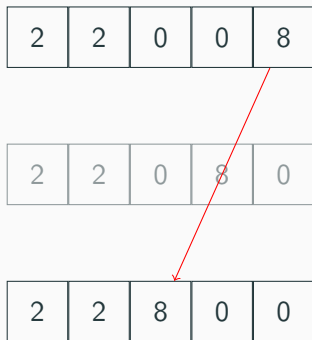
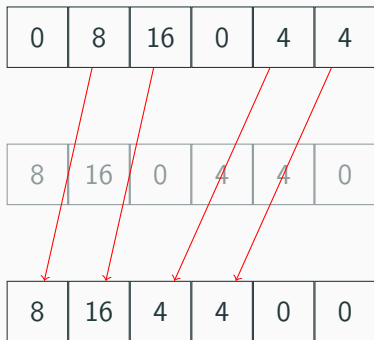
2	2	8	0	0
---	---	---	---	---

2	2	8	0	0
---	---	---	---	---

Move left



Move left



Merge

Einmal pro Zug werden zwei gleiche, nebeneinander liegende Steine verschmolzen.

0	2	2	0
---	---	---	---

0	4	0	0
---	---	---	---



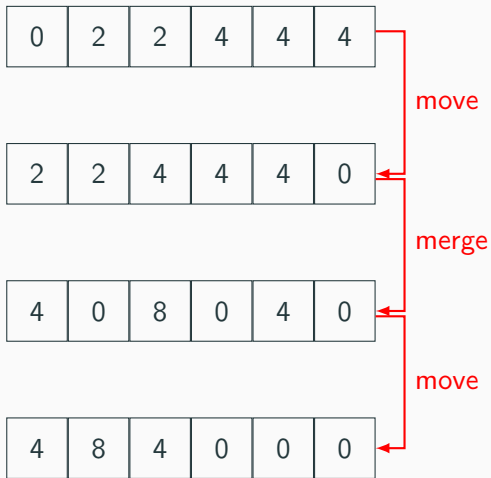
2	2	8	8	8
---	---	---	---	---

4	0	16	0	8
---	---	----	---	---

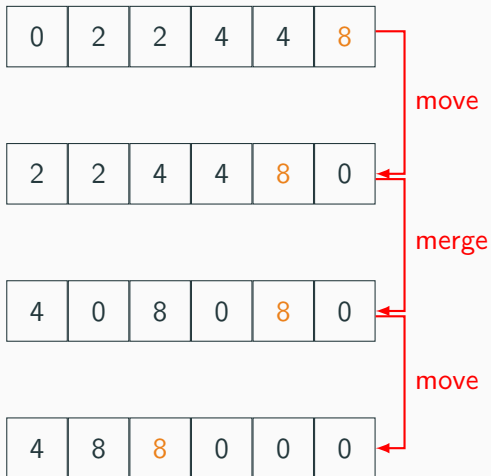


Es gilt zu beachten, dass der neue Stein auf dem linken der beiden Felder entsteht.

Complete Move



Complete Move



Aufgabe 5

In der Datei `printboard.asm`:

Spielfeld ausgeben

- `$a0` - Startadresse des Spielfeldes

Aufgabe 5

In der Datei `printboard.asm`:

Spielfeld ausgeben

- `$a0` - Startadresse des Spielfeldes

Hinweise!

- Nur für ein Feld der Größe 4×4 benötigt
- Leerzeichen beachten! Alle Zahlen sind rechtsbündig. Hat die Zahl zu wenig Stellen, müssen führende Leerzeichen (auf 4 Stellen füllend) eingefügt werden.
- Nach jeder ausgegeben Zeile muss ein `'\n'` ausgegeben werden.
- Nach der letzten Zeile muss eine zusätzliche Leerzeile ausgegeben werden.

Grafische Ausgabe - Beispiel

uuuuuu uuuuuu uuuuuu uuuuuu
uuuu2u uuuu4u uuuu0u uuuu0u
uuuuuu uuuuuu uuuuuu uuuuuu

uuuuuu uuuuuu uuuuuu uuuuuu
uuuu4u uuuu8u uuuu0u uuuu0u
uuuuuu uuuuuu uuuuuu uuuuuu

uuuuuu uuuuuu uuuuuu uuuuuu
uu16u uu32u uuuu4u uuuu2u
uuuuuu uuuuuu uuuuuu uuuuuu

uuuuuu uuuuuu uuuuuu uuuuuu
uuuu2u u2048u uu512u uuuu8u
uuuuuu uuuuuu uuuuuu uuuuuu

In der Datei `points.asm` gilt es, die Punkte für einen Spielzug zu berechnen.

Spielfeld ausgeben

- `$a0` - Startadresse des Spielfeldes (4×4)
- `$a1` - Die Richtung, symbolisiert durch 'w', 'a', 's', 'd'
- `$v0` - Punkte, Rückgabewert

In der Datei `points.asm` gilt es, die Punkte für einen Spielzug zu berechnen.

Spielfeld ausgeben

- `$a0` - Startadresse des Spielfeldes (4×4)
- `$a1` - Die Richtung, symbolisiert durch 'w', 'a', 's', 'd'
- `$v0` - Punkte, Rückgabewert

Regeln

Sei x die Summe der Werte durch "mergen" neu erzeugter Blöcke. Sei v die Anzahl der "merges".

Dann soll $x \times 2^{v-1}$ zurückgegeben werden.

Das Feld selbst darf nicht geändert werden.

Punkte - Beispiel

2	4	2	4
2	4	8	4
8	0	8	16
32	64	0	8

Richtung = 'w'

$$x = 4 + 8 + 16 + 8 = 36$$

$$v = 4$$

$$\$v0 \leftarrow 288 = 36 \times 2^{4-1}$$

Punkte - (Noch mehr) Beispiel(e)

2	0	2	4
2	4	4	4
8	0	8	16
32	64	0	8

Richtung = 'a'

$$x = 4 + 8 + 16 = 28$$

$$v = 3$$

$$\$v0 \leftarrow 112 = 28 \times 2^{3-1}$$

Fragen?

Bei Problemen nutzt das
Forum oder kommt in
die Office-Hours!