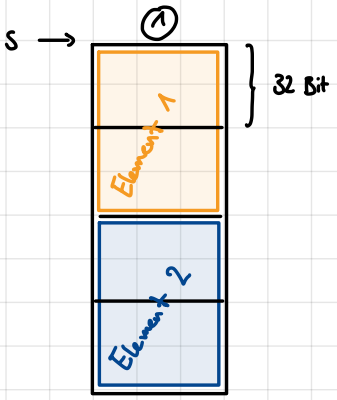
ERA - Übungsblatt 03

1 a) Startadresse + $n \cdot \frac{64}{8}$

b) Für x Byte: Startadresse + $n\,x$

c)



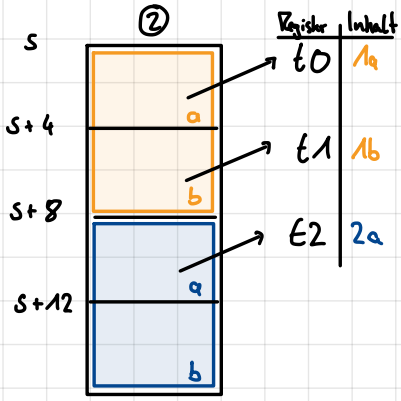
Unsere Elemente sind 64 Bit (= 8 Byte groß), aber in die Register nur 32 Bit: Wir brauchen also für jedes Element 2 Register. Wir starten, indem wir die Adresse s des ersten Elements berechnen. Die Formel aus a) ist hierbei hilfreich:

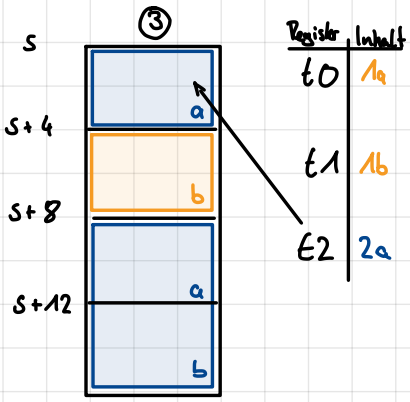
$s = a0 + a1 \cdot 8$

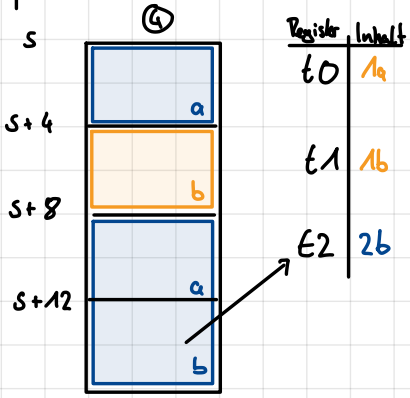
Der Assembly-Code sieht folgendermaßen aus:

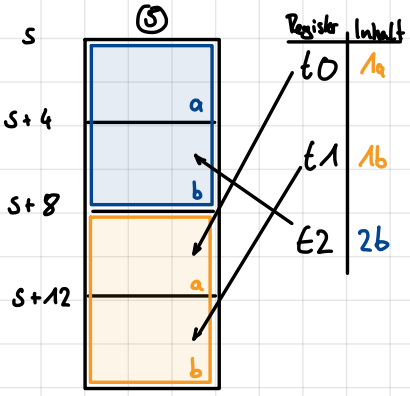
swap:

slli a1, a1, 3	entspricht $a1 \cdot 2^3 = a1 \cdot 8$
add a1, a1, a0	entspricht $a0 := a0 + a1 \equiv a0 := s$
lw t0, 0(a1)	} wir laden die ersten drei 4-Byte-Blöcke
lw t1, 4(a1)	} in temporäre Register. Schritt ②
lw t2, 8(a1)	
sw t2, 0(a1)	} um Register zu sparen, können wir da ersten 4-Byte-Block
lw t2, 12(a1)	} des ersten Elements, also 1a, überschreiben, da wir den Wert
sw t0, 8(a1)	} ja in t0 gesichert haben. Schritt ③
sw t1, 12(a1)	Wir haben also t2 frei, um 2b zu laden. Schritt ④
sw t2, 4(a1)	Abschließend speichern wir noch alle verbliebenden Teile. Schritt ⑤
jalr zero, 0(ra)	Rücksprung nicht vergessen!









Bonus: Versucht das Ganze mal mit nur 2 Registern!

2. a) ASCII: 1 Byte ≙ 1 Zeichen / Character

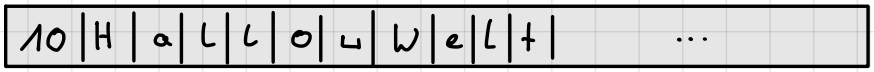
in Hexadezimal: 48 61 6c 6c 6f 20 57 65 6c 74

H a l l o ␣ W e l t

b) NULL-terminierte Strings:

H	a	l	l	o	␣	W	e	l	t	100	...

Pascal - Strings:



Code siehe Webseite / Musterlösung.

c) ⊕ Länge des Strings kann in konstanter Zeit, d.h. $O(1)$, abgerufen werden → sehr wichtig für effiziente String-Operationen
⊖ Mit 1 Byte Längenpräfix auf Längen 0–255 beschränkt.
⊖ Änderungen der Stringlänge (bspw. Konkatenation) erzwingen eine Anpassung des Längenpräfix

3. Code siehe Webseite / Musterlösung

Allgemeines zu Artemis-Hausaufgaben

Hilfe, mein Code funktioniert nicht!

- „Lokal bekomme ich die richtigen Ergebnisse, aber auf Artemis nicht. Der Tester muss kaputt sein!!!"
  → Überprüfe, ob du alle Register, die du verwendest, richtig initialisierst & die calling convention einhälst

- „Mein Code erzeugt eine Endlosschleife und springt nicht dahin wo ich möchte"
  → Überprüfe, ob du die Rücksprungadresse in ra überschreibst, bspw. implizit durch jal label
  Entweder du sicherst ra in einem Register / auf dem Stack oder verwendest Konstrukte wie "j label", "beq zero, zero, label" etc.

- „Mein Code kompiliert auf Artemis nicht, aaaaahhh...."
  → Überprüfe, ob du ein Label mehrmals verwendest, ein Leerzeichen zwischen Labelname und Doppelpunkt steht, ein Komma bei einer Instruktion fehlt, du nicht die Registernamen zero, tp, s1, ... sondern x0, x1, x2, ... verwendest, etc.

- „Absoluter Super-GAU, ich hab seit Stunden debuggt, ich stürme gleich den Artemis-Serverraum"
  → auf Zulip schauen, ob jemand das selbe Problem hat/hatte. Falls nicht ein neues Topic im spezifischen Channel „ERA - Übungsblatt xx" stellen.