

## Laborübung 1

### Bearbeitungshinweise

Die Laborübung ist in Intel x86-64 Assembler (64-bit Programmiermodell) für den NASM Assembler unter dem Betriebssystem Linux (z.B. Ubuntu 16.04) zu erstellen. Verwenden Sie für die folgenden Aufgaben das folgende git-Repository als Grundlage: <https://github.com/rdrcode/snp>

Im Verzeichnis `asmtime` sind Beispielprogramme für das 32-Bit und 64-Bit Programmiermodell sowie ein `Makefile` enthalten.

### Aufgaben

1. Erweitern Sie das Assemblerprogramm `asmtime64`, so dass dieses die aktuelle Systemuhrzeit in Stunden, Minuten und Sekunden berechnet und auf der Standardausgabe ausgibt.

Verwenden Sie für das Ermitteln der Systemuhrzeit den Linux System Call `gettimeofday` (siehe `man 2 gettimeofday`). Der System Call liefert die Zeit in Sekunden seit 1. Jan. 1970, 0:00:00 GMT (sog. Unix Epoche) zurück. Das Ergebnis wird in folgender Struktur abgespeichert (C Syntax):

```
struct timeval {
    time_t tv_sec;    /* seconds since Jan. 1, 1970 */
    suseconds_t tv_usec; /* and microseconds */
};
```

Überlegen Sie zunächst, wie sie aus diesen Werten die Anzahl der am aktuellen Tag vergangenen Sekunden ermitteln können. Berechnen Sie daraus die einzelnen Werte für Stunden, Minuten und Sekunden der aktuellen Uhrzeit und speichern Sie diese im `bss` Segment entsprechend Tabelle 1. Die verwendeten Register bei einer Division mit einem 32-Bit Divisor sind in Abbildung 1 dargestellt.

| Variable                | Bytes | Beschreibung                                    |
|-------------------------|-------|---|
| <code>tv_sec</code>     | 8     | Sekunden seit der Unix Epoche                   |
| <code>tv_usec</code>    | 8     | Mikrosekunden in aktueller Sekunde              |
| <code>secs_today</code> | 4     | Sekunden des aktuellen Tags, d.h. seit 00:00    |
| <code>days_epoch</code> | 4     | Tage seit der Unix Epoche                       |
| <code>wday</code>       | 1     | Wochentag (0 = Sonntag, 1 = Montag, usw.)       |
| <code>hours</code>      | 1     | Stunden der aktuellen Uhrzeit <b>HH</b> :MM:SS  |
| <code>minutes</code>    | 1     | Minuten der aktuellen Uhrzeit HH: <b>MM</b> :SS |
| <code>seconds</code>    | 1     | Sekunden der aktuellen Uhrzeit HH:MM: <b>SS</b> |

Tabelle 1: Variablen im `bss` Segment (`asmtime64`)

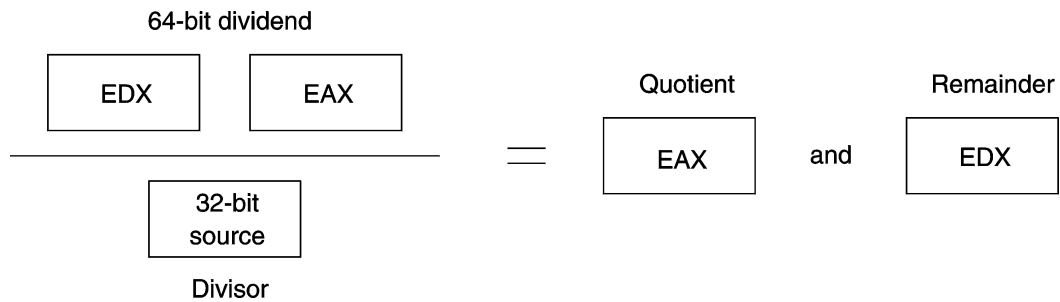


Abbildung 1: Registerbelegung bei einer Division mit 32-Bit Divisor

Für die Ausgabe der einzelnen Integer-Werte wandeln Sie diese in ihre entsprechende BCD Darstellung um und speichern das Ergebnis in einer Zeichenkette im `data` Segment. Geben Sie zusätzlich zur Uhrzeit auch den aktuellen Wochentag auf zwei (deutsch) oder drei Buchstaben (englisch) abgekürzt aus und speichern Sie den numerischen Wert in der Variablen `wday`, wobei „Sonntag“ mit 0, „Montag“ mit 1 usw. kodiert ist.

Die Ausgabe der Zeichenkette erfolgt mit einem *einzelnen* Aufruf des Linux System Calls `write` (siehe `man 2 write`) auf der Standardausgabe (`stdout`).

Formatieren Sie die Werte der Uhrzeit bei der Ausgabe jeweils als zweistellige Dezimalzahl mit führender Null (siehe folgendes Beispiel).

**Beispielausgabe (englisch):**

```
$ ./asmtime
Mon 20:28:31 GMT
```

Die korrekte Ausgabe Ihres Programms können Sie mit dem folgenden Befehl überprüfen (siehe `man 1 date`):

```
$ date --utc +"%a %H:%M:%S GMT"
```

Beachten Sie, dass das Betriebssystem die aktuelle Systemzeit in der Zeitzone GMT zurückgibt.