
Übung 2

Abgabe bis **Donnerstag, 4. April 10:15** via EPIIC: <http://ep.iic.jku.at>.

MIPS Programme müssen in *QtSpim*¹ ausführbar sein, dürfen keine Pseudobefehle enthalten (es sei denn sie sind explizit erlaubt) und müssen ausreichend kommentiert werden.

1. Funktionsaufrufe in MIPS (8 Punkte)

Das C-Funktion `titlecase` konvertiert einen String (Character Array) in Title-Case-Schreibweise (der erste Buchstabe jedes Wortes ist groß geschrieben) und gibt die Anzahl der Worte zurück. Übersetze die Funktion sowie `main` in MIPS Assembler und beachte dabei jeweils die Konventionen für Funktionsaufrufe. Der Pseudobefehl `la` ist erlaubt.

```
char str[] = "this is a string!";

int titlecase(char* c) {
    int words = 0;
    int last_was_whitespace = 1;
    while(*c) {
        if(last_was_whitespace && *c >= 'a' && *c <= 'z') {
            *c ^= 32;
            last_was_whitespace = 0;
        } else {
            if(*c == ' ') {
                words += 1 - last_was_whitespace;
                last_was_whitespace = 1;
            } else {
                last_was_whitespace = 0;
            }
        }
        c++;
    }
    return words + 1 - last_was_whitespace;
}

int main() {
    int words = titlecase(str);
    puts(str);
}
```

Hinweis: Zahlen und Strings können mit dem Befehl `syscall` auf der Konsole ausgegeben werden. Weitere Informationen findest du unter <http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>.

¹Download für alle Plattformen unter <https://sourceforge.net/projects/spimsimulator/files/>.

2. Rekursive Funktionen in MIPS (10 Punkte)

Die Catalan-Zahlen sind durch folgende Rekursion definiert:

$$C(n) = \begin{cases} 1 & \text{wenn } n = 0, \\ \sum_{i=0}^{n-1} C(i) * C(n-i-1) & \text{sonst.} \end{cases}$$

Die folgende rekursive C-Funktion erlaubt die Berechnung der n-ten Catalan Zahl. Übersetze die Funktion in MIPS Assembler und speichere dabei die Variablen **res** und **i** in **\$s0** bzw. **\$t0**. Beachte dabei die Konventionen für Funktionsaufrufe und teste deine MIPS Funktion indem du **C(5)** berechnest.

```
int catalan(int n) {
    if(n < 2) {
        return 1;
    }
    int res = 0;
    for(int i=0; i<n; i++) {
        res += catalan(i)*catalan(n-i-1);
    }
    return res;
}
```

3. Maschinensprache (6 Punkte)

Übersetze die folgende binär kodierte MIPS Funktion zurück in MIPS Assembler und danach in C. Beschreibe die Funktionalität der Funktion in Worten unter der Annahme, dass sie eine positive Zahl als Argument erhält. Eine Übersicht der MIPS Befehle und deren Kodierung findest du in EPIIC.

Adresse	Befehl
0x00400024	0x23BDFFFC
0x00400028	0xAFBE0000
0x0040002C	0x23BE0004
0x00400030	0x20820000
0x00400034	0x20080000
0x00400038	0x20090001
0x0040003C	0x208AFFFF
0x00400040	0x19400005
0x00400044	0x01091020
0x00400048	0x35280000
0x0040004C	0x38490000
0x00400050	0x214AFFFF
0x00400054	0x08100010
0x00400058	0x8FBE0000
0x0040005C	0x23BD0004
0x00400060	0x03E00008