

Praktikum Rechnerarchitektur Sommersemester 2024

Praktikum 2

Ausgabedatum: 22.04.2024

Übung 1:

Schreiben Sie ein MIPS-Programm, das den Text

„Das sollte nun auf dem Stack zu sehen sein“

auf dem Stack ablegt und diesen dann auf dem Bildschirm ausgibt. Ihre MIPS-Implementierung **muss** dabei folgendermaßen aufgebaut sein:

- 1) Implementieren Sie eine `main`: Funktion, die den obigen Text auf den Stack legt. Sie muss dafür natürlich genügend Platz auf dem Stack reservieren. Sie müssen dafür dynamisch die Stringlänge ermitteln, implementieren Sie dafür eine `strlen(char* s)` Funktion die die Länge (ohne das `'\0'` Endezeichen) ermittelt. Ihre `strlen` Funktion darf annehmen, dass `s` ein gültiger Zeiger auf einen Standard C-String ist.
- 2) Implementieren Sie eine `printit`: Funktion, der Sie (aus `main` aufgerufen) den Zeiger auf den String übergeben und die dann die `printf` Funktion zur Ausgabe des Strings aufrufen soll. Beachten Sie die Calling Convention¹, insbesondere z.B. was für ein korrektes Funktionieren von Schachtelfunktionen notwendig ist.

Wichtig: der String muss so abgelegt werden, dass er durch Aufruf der `print String syscall` Funktion (`syscall 4`) ALS GANZES ausgegeben werden kann. Lösungen, die den String Zeichen-für-Zeichen ausgeben, werden nicht akzeptiert.

Eine korrekte Lösung wird also insgesamt drei Funktionen beinhalten.

Übung 2:

Implementieren Sie die eine Funktion mit dem Namen „`magic`“ die folgendermaßen rekursiv definiert ist:

`Magic(n) = n - 10` für `n > 100`

`Magic(n) = Magic(Magic(n+11))` für `n <= 100`

Implementieren Sie die Funktion rekursiv und beachten Sie dabei insbesondere die Application Binary Interface Konventionen. Da die Funktion selbst auch eine Funktion

¹ Die Calling Convention können Sie im Patterson/Hennessy Lehrbuch 5. Auflage auf Seiten 102-108 nachlesen.

aufruft, müssen Sie insbesondere sicherstellen, dass das `$ra` Register vor dem Rücksprung wiederhergestellt wird.
Schreiben Sie zusätzlich eine `main` Funktion, die einen Integer einliest (die Dokumentation -- HELP Menü -- von Mars lesen um den passenden `syscall` zu finden) und damit dann die `magic` Funktion aufruft und das Ergebnis anschließend auf dem Schirm ausgibt. Denken Sie daran, dass für Funktionsaufrufe ausschließlich `jal` verwendet werden darf.

Optionale Erweiterung: anstatt nur eine Zahl n einzulesen für die der Magic Wert berechnet wird, verwenden Sie die eingelesene Zahl als Schleifenobergrenze n und geben Sie alle $Magic(i)$ Werte für $i = 0 \dots n$ aus.

Lösungen, die die ABI nicht korrekt umsetzen sind falsch und werden nicht akzeptiert.

Das gilt insbesondere für das Verwenden von falschen Registern zur Parameterübergabe.

Übung 3:

Implementieren Sie die Hofstädtersche „male“ und „female“ Funktionen. Rufen Sie diese aus `main` für Argumente 0..10 jeweils auf. (via Schleife) und geben Sie das Ergebnis in auf dem Bildschirm aus.

Die Funktionen sind folgendermaßen definiert:

$Male(0) = 0$

$Female(0) = 1$

$Female(n) = n - Male(Female(n-1))$ für $n > 0$

$Male(n) = n - Female(Male(n-1))$ für $n > 0$

Viel Spaß
