School of Computer Science and Information Technology
Computer Systems Architecture (G51CSA) Autumn 2008
Thorsten Altenkirch

Coursework 7 (cw id 123)
Monday, 24 November 2008
Deadline: 1 December 08, 12:00

Collaborating in small groups of up to three students is permitted, but you must implement your own programs (absolutely *do not* copy and paste from others) and provide your own answers where appropriate.

The solution has to be submitted using the departmental coursework submission system, see

http://support.cs.nott.ac.uk/coursework/cwstud/.

Create a directory `ex07` and put all the files to be submitted (but nothing else) into this directory before submitting the directory.

Multiple submission before the deadline are allowed, only the last one will be taken into account.

1. Implement a procedure `strchr` in MIPS assembler which finds the first occurence of a character in a string and returns a pointer to the position. `strchr` has two parameters: the string (i.e. a pointer) and the character (represented as a byte). If the character doesn't occur in the string, `strchr` should return a pointer to the end of the string (i.e. an empty string). Below is a C implementation of `strchr`:

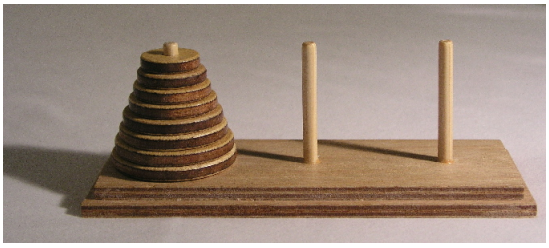
   ```
   char *strchr(char *s,char c) {
     while(*s!='\0' && *s!=c)
       s++;
     return s;
   }
   ```

   Call your procedure from main with some test cases, including the following:

   ```
   int main() {
     printf("%s\n",strchr("hallo",'l'));
     printf("%s\n",strchr("hallo",'h'));
     printf("%s\n",strchr("hallo",'o'));
     printf("%s\n",strchr("hallo",'x'));
   }
   ```

   Call your program `strchr.asm` Make sure that you follow the MIPS procedure calling conventions.

2. The Tower of Hanoi or Towers of Hanoi (also known as The Towers of Brahma) is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks neatly stacked in order of size on one rod, the smallest at the top, thus making a conical shape.



The objective of the puzzle is to move the entire stack to another rod, obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the pegs and sliding it onto another rod, on top of the other disks that may already be present on that rod.
- No disk may be placed on top of a smaller disk.

Our goal is to implement a recursive solution to the problem which prints a sequence of moves and also calculates the number of moves needed. The following C program does this and prints the solution for the case with 5 disks:

```
#include <stdio.h>

int hanoi(int n,char *from,char *via,char *to) {
  int i,j;
  if(n>0) {
    i=hanoi(n-1,from,to,via);
    printf("Move one disc from %s to %s\n",from,to);
    j=hanoi(n-1,via,from,to);
    return(i+j+1);
  } else
    return 0;
}


int main() {
  int n;
  n = hanoi(5,"A","B","C");
  printf("%d moves.\n",n);
}
```

Implement this procedure in MIPS following the procedure calling conventions. Call your program `hanoi.asm`.