CMPE 12 Homework #4

John Allard

November 20th, 2014

1. Write the following in MIPS assembly (with each line commented!) HINT: Use your lab knowledge to do it easily. **Answer :**

```
!  int myarray[5]={1,3,5,7,9};
ADDIU V0, ZERO, 1
SW V0, 4(S8)
ADDIU V0, ZERO, 3
SW V0, 8(S8)
ADDIU V0, ZERO, 5
SW V0, 12(S8)
ADDIU V0, ZERO, 7
SW V0, 16(S8)
ADDIU V0, ZERO, 9
SW V0, 20(S8)
SW ZERO, 0(S8)
for(int i = 0;i<5;i++)
J 0x9D000150
NOP
LW V0, 0(S8)
ADDIU V0, V0, 1
SW V0, 0(S8)
LW V0, 0(S8)
SLTI V0, V0, 5
BNE V0, ZERO, 0x9D00011C
NOP
! myarray[i] = (myarray[i] & 0xf) << 4;
LW V0, 0(S8)
SLL V0, V0, 2
ADDU V0, S8, V0
LW V0, 4(V0)
SLL V0, V0, 4
ANDI V1, V0, 255
LW V0, 0(S8)
SLL V0, V0, 2
ADDU V0, S8, V0
SW V1, 4(V0)
!    }
ADDU SP, S8, ZERO
LW S8, 28(SP)
ADDIU SP, SP, 32
JR RA
NOP
```

2. Write the following in MIPS assembly, with each line commented **Answer :**

```
!int Fibonacci(int n)
{
ADDIU SP, SP, -32
SW RA, 28(SP)
SW S8, 24(SP)
SW S0, 20(SP)
```

```
ADDU S8, SP, ZERO
SW A0, 32(S8)
!if ( n == 0 )
LW V0, 32(S8)
BNE V0, ZERO, 0x9D00010C
NOP
!return 0;
ADDU V0, ZERO, ZERO
J 0x9D000158
NOP
!else if ( n == 1 )
LW V1, 32(S8)
ADDIU V0, ZERO, 1
BNE V1, V0, 0x9D000128
NOP
!return 1;
ADDIU V0, ZERO, 1
J 0x9D000158
NOP
!else
!return ( Fibonacci(n-1) + Fibonacci(n-2) );
LW V0, 32(S8)
ADDIU V0, V0, -1
ADDU A0, V0, ZERO
JAL Fibonacci
NOP
ADDU S0, V0, ZERO
LW V0, 32(S8)
ADDIU V0, V0, -2
ADDU A0, V0, ZERO
JAL Fibonacci
NOP
ADDU V0, S0, V0
}
ADDU SP, S8, ZERO
LW RA, 28(SP)
LW S8, 24(SP)
LW S0, 20(SP)
ADDIU SP, SP, 32
JR RA
NOP
```

3. Given that `a` and `b` are both integers where a and b have been assigned the values 6 and 9 respectively, what is the value of each of the following (12) expressions? If `a` or `b` changes then give their new value.

   (a) `a | b`
   Answer : `000110 | 001001 = 001111` = 15

   (b) `a || b`
   Answer : `6 || 9` = 1 (or any other non-zero value)

   (c) `a & b`
   Answer : `000110 & 001001 = 000000 = 0`

   (d) `a && b`
   Answer : `6 && 9` = 1 (or any other non-zero value)

(e) `!(a+b)`
Answer : False, because `(a+b)` is true

(f) `a % b`
Answer : 6, because $\frac{6}{9} = 0$ r 6

(g) `b / a`
Answer : `1`

(h) `a = b`
Answer : `a = 9`, value of expression returned is 9

(i) `a = b = 5`
Answer : `b = 5, a = 5`, the value returned is 5

(j) `++a + b--`
Answer : `a = 7, b = 8` Expression returns the value of `7+9 = 16`

(k) `a = (++b < 3)? a : b)`
Answer : `a = 10, b = 10`, Expression returns a value of `10`

(l) `a<<=b`

4. Supposed a program contains the two integer variables `x` and `y`, which have values of 3 and 4 respectively. Write C statements that will exchange the values in `x` and `y` so that after the statements are executed, `x` is equal to 4 and `y` is equal to 3.

(a) First do this using a temporary variable for storage
```
int temp = x;
x = y;
y = temp;
```

(b) Now rewrite this routine without using a temporary variable.
```
x = x+y
y = x-y
x = x-y
```

5. (a) Convert the following `while` loop into a `for` loop.
```
while (condition)
loopBody
```
Answer :
```
for(;condition;)
loopbody
```

(b) Convert the follow `for` loop into a while loop
```
for(init;confition;reinit)
loopbody
```
Answer :
```
init;
while(condition) {
loopBody;
reinit;
}
```

6. Provide the output for each of the following code statements.

(a) **Answer :** `******`

(b) **Answer :** `**********`

(c) **Answer :** `*****` (one for each of the odd numbers, 1-10)

(d) **Answer :** `*********`

7. For each of the following items, identify whether the caller fucntion or the callee function performs the actions.

   (a) Writing the parameters into the activation record.
   Answer : Caller

   (b) Writing the return value.
   Answer : Callee

   (c) Writing the dynamic link.
   Answer : Callee

   (d) Modifying the value in R5 to point within the callee functions activation record.
   Answer : Caller

8. TODO

9. Write a C program that computes the pig-latin translation of an english word.

```
int translate(char str[]) {
if(str == NULL) // make sure the argument isn't null
return 0;
int i = 0;
while(str[i+1] != '\0') // find the end of the c string
i++;
char temp = str[i]; // swap the first and last letters;
str[i] = str[0];  str[0] = temp;

str[++i] = 'a'; // add on the last two letters
str[++i] = 'y';
str[++i] = '\0'; // set the null character to mark the new end of the string
}
```