DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Computer Organization and Architecture (CS2600)

Total Time : 50 minutes **Tutorial 1**
Max Marks : 20 (10 + 10) **09-02-2024**

---

1. Complete this program that returns the sum of the first N elements of the Fibonacci sequence.

```
.section .text
.global _start

_start:
    . . . # fill in here to pass parameter
    jal ra, fibonacci_sum
    ret

.global fibonacci_sum
fibonacci_sum:
    addi sp, sp, -8
    sw   ra, 4(sp)
    sw   fp, 0(sp)
    add  fp, sp, 8
    # Initialize variables
    li   t0, 0      # Counter for Fibonacci sequence
    li   t1, 1      # First Fibonacci number
    li   t2, 1      # Second Fibonacci number
    li   t3, 0      # Sum of Fibonacci numbers
fibonacci_loop:
     . . . # complete the remaining here
     ret
```

```
    # pass parameter as follows
    li  a0, 10  # N = 10 in this program

fibonacci_loop:
    # Update the sum
    add  t3, t3, t1

    add  t4, t1, t2  # t4 = current + next
    mv   t1, t2      # Move next to current
    mv   t2, t4      # Move t4 to next

    # Increment counter
    addi t0, t0, 1

    # Check if N Fibonacci numbers have been summed
    blt  t0, a0, fibonacci_loop

    # Function epilogue (restore ra and fp)
    lw   ra, -4(fp)
    lw   fp, -8(fp)
    addi sp, sp, 8

    # Return the sum in a0
    mv   a0, t3
    ret
```

Grading: • 5 marks to get the logic correct • 1 mark for return in a0 • 2 marks for function epilogue correctly; • 1 mark for using the right registers i.e. caller / callee saved. • 1 mark to pass parameter correctly

2. Complete the program that performs the `strcat` function.

```
.section .data
source_string:  .ascii "Hello, "
append_string:  .ascii "world!"
.section .bss
destination_string: .space 50  # Allocate space for the destination string
.section .text
.global _start
_start:
    . . . # fill to pass parameters in the following format.
    . . . # a0 source_string, a1 append_string, a2 destination_string
    jal ra, strcat
    ret
# strcat function
.global strcat
strcat:
    # Function prologue (save ra)
    addi sp, sp, -4
    sw   ra, 0(sp)
    # Copy source string to destination
copy_source:
    . . . # to be filled
end_strcat:
    # Copy append string to destination
copy_append:
    . . . # to be filled
    # Function epilogue (restore ra)
    . . . # to be filled
    ret
```

```
.section .data
source_string:  .ascii "Hello, "
append_string:  .ascii "world!"
.section .bss
destination_string: .space 50
.section .text
.global _start
_start:
    la   a0, source_string
    la   a1, append_string
    la   a2, destination_string
    jal ra, strcat
    ret

# strcat function
.global strcat
strcat:
    # Function prologue (save ra)
    addi sp, sp, -4
    sw   ra, 0(sp)
    # Copy source string to destination
copy_source:
    lb   t0, 0(a0)   # Load byte from source
    beq  t0, zero, end_strcat  # Same as copy_append

    sb   t0, 0(a2)   # Store byte to destination
    addi a0, a0, 1   # Move to the next byte in source
    addi a2, a2, 1   # Move to the next byte in destination
    j    copy_source  # Repeat the loop

end_strcat:
    # Copy append string to destination
```

```
copy_append:
    lb   t0, 0(a1)   # Load byte from append
    beq  t0, zero, epilogue  # If end of append string, exit loop

    sb   t0, 0(a2)   # Store byte to destination
    addi a1, a1, 1   # Move to the next byte in append
    addi a2, a2, 1   # Move to the next byte in destination
    j    copy_append  # Repeat the loop

epilogue:
    # Function epilogue (restore ra)
    lw   ra, 0(sp)
    addi sp, sp, 4

    ret
```

Grading: • 6 marks to get the logic correct • 1 mark for return in a0 • 1 marks for function epilogue correctly; • 1 mark for using the right registers i.e. caller / callee saved. • 1 mark to pass parameter correctly