Worksheet 1

1/15/2025

10 Points Possible

Attempt 1

1/13/2025

NEXT UP: Review Feedback

Attempt 1 Score: N/A

Add Comment

Unlimited Attempts Allowed

Details

Question 1

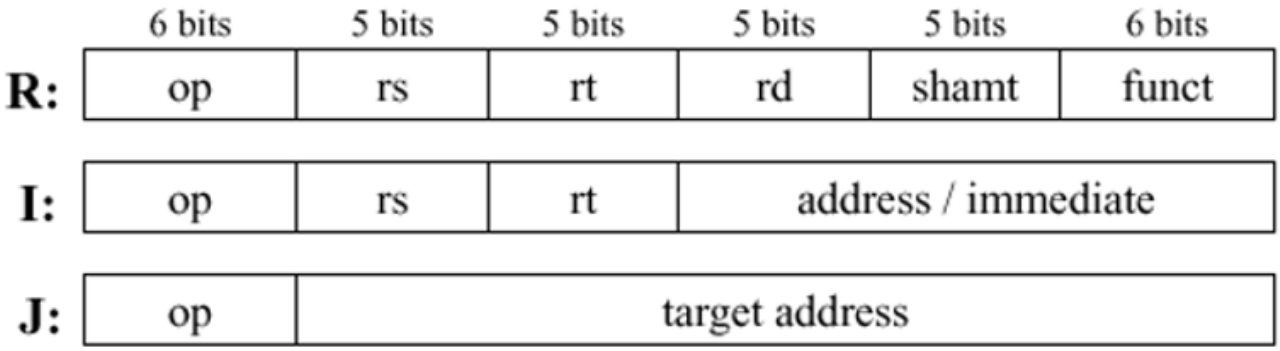
What sort of stuff could go wrong when running multiple programs? List 4 issues.

Question 2

What are the contents of the registers after executing the following instruction?

001110 01010 01100 01010 00000 000000

- 001110 is OP code for add, a type 'R' (Register) operation.
- Instruction format:



Initial Contents

Register	Content
R9	125
R10	56
R11	196

R12	1323

After Execution

Register	Content
R9	?
R10	?
R11	?
R12	?

Answer 1:

Here are the four issues:
a. memory management issue (not enough memory allocated) for the jobs
b. difficulty in resource management (jobs taking too much or too little share in CPU)
c. multiple jobs sharing/competing for the same resource
d. security issue of cross-interference between jobs (such as, read/write operation from one program by one user to another program by another user)

Answer 2:
(this is a naive first attempt with a baseline understanding)

According to the instructions, here is the probable mapping:

(initial) sequential mapping of register contents

register name	bits	instruction	meaning/value
op	6 bits	**001110**	add operation
rs	5 bits	**01010** (decimal number 10)	(initial) R10 = 56
rt	5 bits	**01100** (decimal number 12)	(initial) R12 = 1323
rd	5 bits	**01010** (decimal number 10)	(initial) R10 = 56
shamt	5 bits	**00000**	no operation

func	6 bits	**000000**	no operation

(after execution = after add operation)
MIPS operation: **add $rd, $rs, $rt**
which means, $rd (R10) <- $rs (R10) + $rt (R12)
I am assuming, we will add the contents of R10 and R12 and save the result in R10.

Initial values:
R10 = 56 , R12 = 1323

Execution:
R10 + R12 = 56 + 1323 = 1379

After execution values:
R10 = 1379 (as the result gets stored into R10)
R12 = 1323 (remains unchanged)

The answer after execution,
R9 = 125
R10 = 1379
R11 = 196
R12 = 1323

New Attempt