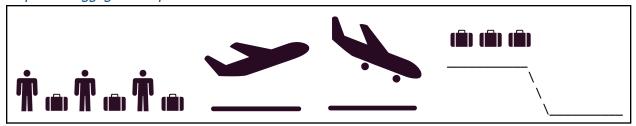
Stacks and Subroutines

Airplane Luggage Example



d ']',
:
•
:
:

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Input: "()"	
Character : "("	
Character : ")"	
Input: "{[]}"	ļ ļ
Character : "{"	
Character : "["	_
Character : "]"	
Character : "}"	
	<u> </u>
	i <u></u> i
	, ,
Input: "[)"	
Character : "["	
Character : ")"	
	'

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Input: "([)]" Character : "["	
Character : ")"	
Input: "([)]"	ļ ļ
Character : "("	
Character : "["	
Character : ")"	
Character : "]"	
Subroutine	
aka:	
We can use the stack to	
so we can retrieve them	
In MIPS, the stack grows	

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

egister Review
StO - \$t9
Ss0 - \$s7
SaO - \$a3
Sv0 - \$v1
Gra
• .
Ssp

Subroutine Diagram

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Stack Usage

Push One Register

How to push a register value to the stack:

Example

Push \$s0 to stack ex/ \$s0 = 0xAAAAAAAA

\$sp →

ADDRESS	CONTENTS
0x7FFF FE00	0xDEADBEEF
0x7FFF FDFC	
0x7FFF FDF8	
0x7FFF FDF4	

Pop One Register

How to pop a register value from the stack:

Example

Pop one word from stack

\$sp

ADDRESS CONTENTS

0x7FFF FE00 0xDEADBEEF

0x7FFF FDFC

0x7FFF FDF8

0x7FFF FDF4

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Push Multiple Registers			
How to push multiple registers to th	e stack:		
xample			
Push \$s0, \$s1, \$s2 to stack ex/ \$s0 = 0xAAAAAAAA		ADDRESS	CONTENTS
\$s1 = 0xBBBBBBBB \$s2 = 0xCCCCCCC	\$sp →	0x7FFF FE00	0xC0FFEEEE
		0x7FFF FDFC	
		0x7FFF FDF8	
		0x7FFF FDF4	
One at a time:			
Decrement \$sp once:			

© 2019, Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

How to pop multiple registers from the stack: ADDRESS CONTENTS	How to pop multiple registers from the stack: ADDRESS CONTENTS					
0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB 0x7FFF FDF4 0xCCCCCCCC One at a time:	How to pop multiple registers from the stack: ADDRESS CONTENTS					
How to pop multiple registers from the stack: ADDRESS CONTENTS	How to pop multiple registers from the stack: ADDRESS CONTENTS					
ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp 0x7FFF FDF4 0xCCCCCCCC One at a time:	ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBB \$sp 3 0x7FFF FDF4 0xCCCCCCCC	op Multiple Registers				
Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp → 0x7FFF FDF4 0xCCCCCCCC One at a time:	Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp 0x7FFF FDF4 0xCCCCCCCC One at a time:	How to pop multiple registers from	the stack	<:		
Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp → 0x7FFF FDF4 0xCCCCCCCC One at a time:	Pop 3 words from stack ADDRESS CONTENTS 0×7FFF FE00 0×C0FFEEEE 0×7FFF FDFC 0×AAAAAAAA 0×7FFF FDF8 0×BBBBBBBB \$\$p\$ 0×7FFF FDF4 0×CCCCCCCC One at a time:					
Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp → 0x7FFF FDF4 0xCCCCCCCC One at a time:	Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp 0x7FFF FDF4 0xCCCCCCCC One at a time:					
Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp	Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp 0x7FFF FDF4 0xCCCCCCCC One at a time:					
Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp	Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp 0x7FFF FDF4 0xCCCCCCCC One at a time:					
Pop 3 words from stack ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp → 0x7FFF FDF4 0xCCCCCCCC One at a time:	Pop 3 words from stack ADDRESS CONTENTS 0×7FFF FE00 0×C0FFEEEE 0×7FFF FDFC 0×AAAAAAAA 0×7FFF FDF8 0×BBBBBBBB \$\$p\$ 0×7FFF FDF4 0×CCCCCCCC One at a time:	vamnle				
ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp → 0x7FFF FDF4 0xCCCCCCCC One at a time:	ADDRESS CONTENTS 0x7FFF FE00 0xC0FFEEEE 0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBBB \$sp → 0x7FFF FDF4 0xCCCCCCCC One at a time:					
0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBB 0x7FFF FDF4 0xCCCCCCCC One at a time:	0x7FFF FDFC 0xAAAAAAAA 0x7FFF FDF8 0xBBBBBBB 0x7FFF FDF4 0xCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC			ADDRESS	CONTENTS	
\$sp → 0x7FFF FDF4 0xCCCCCCC One at a time:	\$sp → 0x7FFF FDF8 0xBBBBBBB 0x7FFF FDF4 0xCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC			0x7FFF FE00	0xC0FFEEEE	
\$sp	\$sp			0x7FFF FDFC	0×AAAAAAAA	
One at a time:	One at a time:			0x7FFF FDF8	0xBBBBBBBB	
One at a time:	One at a time:			0x7FFF FDF4	0xcccccc	
			7			
Increment \$sp once:	<pre>Increment \$sp once:</pre>	One at a time:				
<pre>Increment \$sp once:</pre>	Increment \$sp once:					
<pre>Increment \$sp once:</pre>	Increment \$sp once:					
Increment \$sp once:	Increment \$sp once:					
<pre>Increment \$sp once:</pre>	Increment \$sp once:					
<pre>Increment \$sp once:</pre>	Increment \$sp once:					
<pre>Increment \$sp once:</pre>	Increment \$sp once:					
Increment \$sp once:	Increment \$sp once:					
Increment \$sp once:	Increment \$sp once:					

^{© 2019,} Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

Jump Instructions
J:
Give an example of how to use this instruction
J
What does this instruction do?
How does this instruction affect the registers?
now does this instruction affect the registers:
JAL:
Give an example of how to use this instruction
JAL
What does this instruction do?
How does this instruction affect the registers?
JALR:
Give an example of how to use this instruction
JALR

© 2019, Rebecca Rashkin - This document may be copied, redistributed, transformed, or built upon in any format for **educational**, non-commercial purposes. Please give me appropriate credit should you choose to modify this resource. Thank you:)

What does this instruction do? How does this instruction affect the registers? JR: _____ Give an example of how to use this instruction

_	_	-	_	1-	_	-	-	_	_	-	_	_	_	_	_	_	-	_	-

What does this instruction do?

How does this instruction affect the registers?

Examples

Basic Example

address of instruction	instruc	instruction					
0x100	Main:	NOP					
0x104		JAL	Sub1				
0x108		NOP					
0x10C		JAL	Sub2				
0x110		NOP					
0x200	Sub1:	NOP					
0x204		JR	\$ra				
•••	• • •						
0x300	Sub2:	NOP					
0x304		JR	\$ra				

time	current pc	ra (after instr)	pc (after instr)
1			
2			
3			
4			
5			
6			
7			
8			
9			

Nested Subroutine

address of instruction	instruction
0x100	Main: NOP
0x104	JAL Sub2
0x108	NOP
•••	
0x200	Sub1: NOP
0x204	JR \$ra
•••	
0x300	Sub2: NOP
0x304	JAL Sub1
0x308	NOP
0x30C	NOP
0x310	JR \$ra

time	current pc	ra (after instr)	pc (after instr)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Which subroutines are the callers?							
Which are the callees?							
Is there a problem with this code?							

Callee Save Registers

Caller Save Registers

Callee Save Example

Use the stack to save \$ra

address of instruction	instruction		
0x100	Main:	NOP	
0x104		JAL	Sub2
0x108		NOP	
• • •			
0x200	Sub1:	NOP	
0x204		JR	\$ra
0x300	Sub2:	NOP	
0x304			
0x308			
0x30C		JAL	Sub1
0x310		NOP	
0x314		NOP	
0x318			
0x31C			
0x310		JR	\$ra
•••			

time	pc (crnt instr)	ra (after instr)	pc (after instr)	memory address (in stack)	contents
1				0xFE00	
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					