



















	10	n-L	-ea	f Proc	edure Example								
•	MIPS code:												
	fact:												
		addi sw sw	\$sp, \$ra, \$a0	\$sp, -8 4(\$sp) 0(\$sp)	<pre># adjust stack for 2 items # save return address # save argument</pre>								
		slti beq	\$t0, \$t0,	\$a0, 1 \$zero, L1	# test for n < 1								
		addi addi j r	\$v0, \$sp, \$ra	\$zero, 1 \$sp, 8	<pre># if so, result is 1 # pop 2 items from stack # and return</pre>								
	L1:	addi j al	\$a0, fact	\$a0, -1	<pre># else decrement n # recursive call</pre>								
		lw Iw	\$a0, \$ra,	0(\$sp) 4(\$sp)	<pre># restore original n # and return address</pre>								
		addi	\$sp,	\$sp, 8	<pre># pop 2 items from stack</pre>								
		jr	\$v0, \$ra	\$a0, \$v0	# multiply to get result # and return								
M				Chapte	er 2 — Instructions: Language of the Computer — 6								









S	Str	ing	J C	ору Ех	a	mple	
	MI	⊃S c	ode:				
	stro	cpy:					1
		addi sw	\$sp, \$s0,	\$sp, -4 O(\$sp)	# #	adjust stack for 1 item save \$s0	
		add	\$s0,	\$zero, \$zero	#	i = 0	
	L1:	add	\$t1,	\$s0, \$a1	#	addr of y[i] in \$t1	
		l bu	\$t2,	0(\$t1)	#	\$t2 = y[i]	
		add	\$t3,	\$s0, \$a0	#	addr of x[i] in \$t3	
		sb	\$t2,	0(\$t3)	#	x[i] = y[i]	
		beq	\$t2,	\$zero, L2	#	<pre>exit loop if y[i] == 0</pre>	
		addi	\$s0,	\$s0, 1	#	i = i + 1	
		j	L1		#	next iteration of loop	
	L2:	l w	\$s0,	0(\$sp)	#	restore saved \$s0	
		addi	\$sp,	\$sp, 4	#	pop 1 item from stack	
		jr	\$ra		#	and return	
	<u>.</u>						
	<b>C</b>			Chapter 2	— I	nstructions: Language of the Computer -	_















