

# Recursividad

*Trazando factorial(3) usando la pila  
con QtSpim*



# STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
	7ffffe1c
	7ffffe20
	7ffffe24
	7ffffe28
	7ffffe2c
	7ffffe30
\$sp →	7ffffe34

# BANCO REGISTROS

\$a0	3	
\$v0		
\$ra	0040002C	00400028
User data segment [10000000]..[10040000] [10000000]..[1003ffff] 00000000		
		0040002C

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
           li $v0,1
           jr $ra
masquecero:
           addi $sp, $sp, -8 # reservo 8 bytes en la pila
           sw $ra, 4($sp) # guardo $ra
           sw $a0, 0($sp) # guardo $a0

           addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial      # llamo a fact(n-1) devuelve en $v0
00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
           lw $ra,4($sp)      # recupero $ra en el mismo orden
           addi $sp, $sp, 8    # dejo el puntero de pila como estaba

           mult $a0,$v0        # multiplico n*fact(n-1) que es $a0*$v0
           mflo $v0            # dejo en $v0 el valor del factorial
           jr $ra
# FIN FUNCION factorial
```



# STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
	7ffffe1c
	7ffffe20
	7ffffe24
	7ffffe28
\$sp →	7ffffe2c
	7ffffe30
	7ffffe34

# BANCO REGISTROS

\$a0	3
\$v0	
\$ra	0040002C

00400028

0040002C

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra

masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial # llamo a fact(n-1) devuelve en $v0

00400060 → lw $a0,0($sp) # recupero $a0 en el mismo orden
            lw $ra,4($sp) # recupero $ra en el mismo orden
            addi $sp, $sp, 8 # dejo el puntero de pila como estaba

            mult $a0,$v0 # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0 # dejo en $v0 el valor del factorial
            jr $ra
IN factorial
```

User Stack [7ffffe2c]..[80000000]			
[7ffffe2c]	00000003	00000001	7ffffe6d
[7ffffe30]	0040002c	00000000	7ffffe6e
[7ffffe40]	7fffffff	7fffffd1	7fffffc2
[7ffffe50]	7fffff6d	7fffff5e	7fffff38
[7ffffe60]	7ffffed8	7ffffeca	00000000
[7ffffe70]	2f737265	6c726163	442f736f
[7ffffe80]	642f786f	6e65636f	2f616963
[7ffffe90]	69706963	6f43736f	7475706d
[7ffffea0]	632f7365	6f737275	322d3931
[7ffffeb0]	6f697365	7475546e	6169726f



## STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
	7ffffe1c
	7ffffe20
	7ffffe24
	7ffffe28
\$sp →	3
	7ffffe2c
0040002C	7ffffe30
	7ffffe34

## BANCO REGISTROS

\$a0	2
\$v0	
\$ra	00400060

00400028 →

0040002C →

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
           li $v0,1
           jr $ra
masquecero:
           addi $sp, $sp, -8 # reservo 8 bytes en la pila
           sw $ra, 4($sp) # guardo $ra
           sw $a0, 0($sp) # guardo $a0
0040005C → addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
           jal factorial    # llamo a fact(n-1) devuelve en $v0
00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
           lw $ra,4($sp)     # recupero $ra en el mismo orden
           addi $sp, $sp, 8  # dejo el puntero de pila como estaba

           mult $a0,$v0      # multiplico n*fact(n-1) que es $a0*$v0
           mflo $v0          # dejo en $v0 el valor del factorial
           jr $ra
# FIN FUNCION factorial
```



# STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
	7ffffe1c
	7ffffe20
\$sp → 2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
0040002C	7ffffe30
	7ffffe34

# BANCO REGISTROS

\$a0	2
\$v0	
\$ra	00400060

00400028

0040002C

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C→    jal factorial      # llamo a fact(n-1) devuelve en $v0

00400060→    lw $a0,0($sp)        # recupero $a0 en el mismo orden
            lw $ra,4($sp)    # recupero $ra en el mismo orden
            addi $sp, $sp, 8 # dejo el puntero de pila como estaba

            mult $a0,$v0    # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0        # dejo en $v0 el valor del factorial
            jr $ra
ON factorial
```

User Stack [7ffffe24]..[80000000]			
[7ffffe24]	00000002	00400060	00000003
[7ffffe30]	0040002c	00000001	7ffffe6d
[7ffffe40]	7fffffff4	7fffffff1	7fffffc2
[7ffffe50]	7fffffff6d	7fffffff5e	7fffff38
[7ffffe60]	7ffffed8	7ffffeca	00000000
[7ffffe70]	2f737265	6c726163	442f736f
[7ffffe80]	642f786f	6e65636f	2f616963
[7ffffe90]	60706063	6f427265	74757063



## STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
	7ffffe1c
	7ffffe20
\$sp →	2
	00400060
	3
	0040002C
	7ffffe34

## BANCO REGISTROS

\$a0	1
\$v0	
\$ra	00400060

00400028

0040002C

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0
0040005C → addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
            jal factorial    # llamo a fact(n-1) devuelve en $v0
00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
            lw $ra,4($sp)    # recupero $ra en el mismo orden
            addi $sp, $sp, 8 # deajo el puntero de pila como estaba

            mult $a0,$v0     # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0         # deajo en $v0 el valor del factorial
            jr $ra
# FIN FUNCION factorial
```





# STACK

# BANCO REGISTROS

\$sp →

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
1	7ffffe1c
00400060	7ffffe20
2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
	7ffffe30
0040002C	7ffffe34

\$a0	1
\$v0	
\$ra	00400060

00400028

0040002C

User Stack [7ffffe1c]..[80000000]				
[7ffffe1c]	00000001			
[7ffffe20]	00400060	00000002	00400060	00000003
[7ffffe30]	0040002c	00000001	7ffffe6d	00000000
[7ffffe40]	7fffffff4	7fffffd1	7fffffc2	7fffff80
[7ffffe50]	7fffff6d	7fffff5e	7fffff38	7ffffeff
[7ffffe60]	7ffffed8	7ffffeca	00000000	73552f00
[7ffffe70]	2ffff7265	62726162	44267265	62706572

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C→    jal factorial      # llamo a fact(n-1) devuelve en $v0

00400060→    lw $a0,0($sp)      # recupero $a0 en el mismo orden
            lw $ra,4($sp)  # recupero $ra en el mismo orden
            addi $sp, $sp, 8 # dejo el puntero de pila como estaba

            mult $a0,$v0   # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0       # dejo en $v0 el valor del factorial
            jr $ra
N factorial
```



## STACK

\$sp →

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
1	7ffffe1c
00400060	7ffffe20
2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
0040002C	7ffffe30
	7ffffe34

## BANCO REGISTROS

\$a0	0
\$v0	
\$ra	00400060

00400028 →

0040002C →

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

0040005C →    addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
            jal factorial # llamo a fact(n-1) devuelve en $v0

00400060 →    lw $a0,0($sp) # recupero $a0 en el mismo orden
            lw $ra,4($sp) # recupero $ra en el mismo orden
            addi $sp, $sp, 8 # dejo el puntero de pila como estaba

            mult $a0,$v0 # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0 # dejo en $v0 el valor del factorial
            jr $ra
# FIN FUNCION factorial
```





## STACK

\$sp →

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
1	7ffffe1c
00400060	7ffffe20
2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
0040002C	7ffffe30
	7ffffe34

## BANCO REGISTROS

\$a0	0
\$v0	1
\$ra	00400060

00400028 →

0040002C →

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
           li $v0,1
           jr $ra

masquecero:
           addi $sp, $sp, -8 # reservo 8 bytes en la pila
           sw $ra, 4($sp) # guardo $ra
           sw $a0, 0($sp) # guardo $a0

           addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial      # llamo a fact(n-1) devuelve en $v0

00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
           lw $ra,4($sp)      # recupero $ra en el mismo orden
           addi $sp, $sp, 8    # dejo el puntero de pila como estaba

           mult $a0,$v0        # multiplico n*fact(n-1) que es $a0*$v0
           mflo $v0            # dejo en $v0 el valor del factorial
           jr $ra

# FIN FUNCION factorial
```



## STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
	7ffffe1c
	7ffffe20
	7ffffe24
	7ffffe28
	7ffffe2c
	7ffffe30
	7ffffe34

  

	1
	00400060
	2
	00400060
	3
	0040002C

\$sp →

## BANCO REGISTROS

\$a0	1
\$v0	1
\$ra	00400060

00400028

0040002C

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial      # llamo a fact(n-1) devuelve en $v0
00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
            lw $ra,4($sp)      # recupero $ra en el mismo orden
            addi $sp, $sp, 8    # deajo el puntero de pila como estaba

            mult $a0,$v0        # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0            # deajo en $v0 el valor del factorial
            jr $ra

# FIN FUNCION factorial
```



## STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
1	7ffffe1c
00400060	7ffffe20
2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
0040002C	7ffffe30
	7ffffe34

\$sp →

## BANCO REGISTROS

\$a0	2
\$v0	2
\$ra	00400060

00400028 →

0040002C →

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial      # llamo a fact(n-1) devuelve en $v0
00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
            lw $ra,4($sp)      # recupero $ra en el mismo orden
            addi $sp, $sp, 8    # deajo el puntero de pila como estaba

            mult $a0,$v0        # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0            # deajo en $v0 el valor del factorial
            jr $ra

# FIN FUNCION factorial
```



## STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
1	7ffffe1c
00400060	7ffffe20
2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
0040002C	7ffffe30
\$sp →	7ffffe34

## BANCO REGISTROS

\$a0	3
\$v0	6
\$ra	0040002C

00400028

0040002C

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial      # llamo a fact(n-1) devuelve en $v0
00400060 → lw $a0,0($sp)      # recupero $a0 en el mismo orden
            lw $ra,4($sp)      # recupero $ra en el mismo orden
            addi $sp, $sp, 8    # deajo el puntero de pila como estaba

            mult $a0,$v0        # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0            # deajo en $v0 el valor del factorial
            jr $ra

# FIN FUNCION factorial
```



## STACK

	7ffffe08
	7ffffe0c
	7ffffe10
	7ffffe14
	7ffffe18
1	7ffffe1c
00400060	7ffffe20
2	7ffffe24
00400060	7ffffe28
3	7ffffe2c
0040002C	7ffffe30
\$sp →	7ffffe34

## BANCO REGISTROS

\$a0	3
\$v0	10
\$ra	0040002C

00400028

0040002C

```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff] 00000000
[10010000] 00000006 00000000 00000000 00000000
[10010010]..[1003ffff] 00000000
```

```
.data
resultado: .word 0
.text
# Ejemplo calcular factorial de 3
main:
    li $a0,3
    jal factorial
    # en $v0 queda el resultado
    sw $v0,resultado
    li $v0, 10
    syscall

# SUBROUTINA factorial $v0 = ($a0)!
factorial: bgt $a0,$zero,masquecero
            li $v0,1
            jr $ra
masquecero:
            addi $sp, $sp, -8 # reservo 8 bytes en la pila
            sw $ra, 4($sp) # guardo $ra
            sw $a0, 0($sp) # guardo $a0

            addi $a0,$a0,-1 # fact(n) = n * fact(n-1)
0040005C → jal factorial # llamo a fact(n-1) devuelve en $v0

00400060 → lw $a0,0($sp) # recupero $a0 en el mismo orden
            lw $ra,4($sp) # recupero $ra en el mismo orden
            addi $sp, $sp, 8 # dejo el puntero de pila como estaba

            mult $a0,$v0 # multiplico n*fact(n-1) que es $a0*$v0
            mflo $v0 # dejo en $v0 el valor del factorial
            jr $ra
# FIN FUNCION factorial
```