



Registers		
Coproc 1		
Coproc 0		
Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000001
\$v0	2	0x00000037
\$v1	3	0x00000000
\$a0	4	0x0000000a
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00400008
pc		0x00400064
hi		0x00000000
lo		0xffffffff

程式運作流程：

Main function:

1. 將測資放入\$a0

Fib function:

1. 備份\$a0 資料到\$sp 之中
2. 將\$a0 值-1，儲存在\$t1 之中
3. 比較\$t1 與\$zero 的值，若\$t1>0(表示 n>1)，則跳到 L1 function
4. 若\$t1<=0 (表示 n==0 || n==1)，則把\$a0 值存到\$t0 裡面，再存到 \$v0(result register) 裡面
5. 把 return address lw 到\$ra 裡，再把\$sp 加回來，然後 return

L1 function:

1. 把\$a0 值-1，丟回 fib，得 fib(n-1)
2. 因為此時\$a0 是已將原\$a0-1 的值，所以再減 1 丟回 fib 計算即得 fib(n-2)的結果
3. 把備份值 lw 到\$a0 裡
4. 將\$sp 加回來

心得：

撰寫 MIPS 由於不像之前寫 C 或 C++那麼直觀，需要更多的邏輯思考和反覆確認每一行指令，因為在 assemble 時基本上很多時候都會過，但結果卻會差了很多。因此寫組語需要更細心地去檢查指令內容，充分了解，才能夠使答案正確