**汇编第二次大作业**

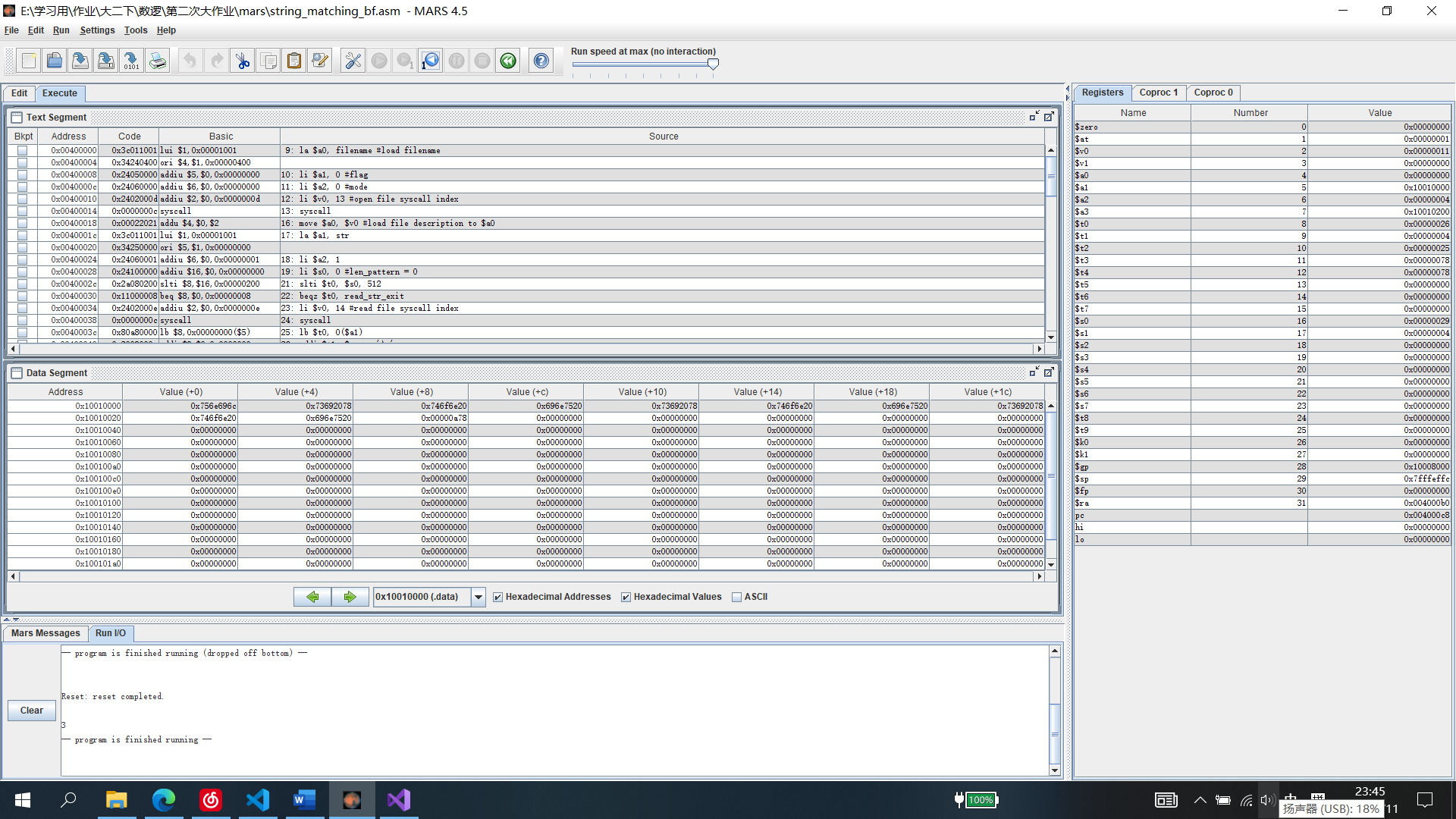
**1 . brute-force**

**$t0对应变量i**

**$t1对应变量j**

**$v0对应变量cnt**

**过程调用时先把$t0,$t1,$v0即i,j,cnt置0，计算得到len\_str-len\_pattern并存在$t2中，进入关于i的循环，把$t1(变量j)置0。进入关于j的循环，分别计算str[i+j]和pattern[j]并储存在$t3和$t4中，如果两者不相等跳出关于j的循环，判断j和len\_pattern是否相等，相等cnt+=1，不等则i+=1，判断i和len\_str-len\_pattern大小决定是进入下一次关于i的循环还是直接返回**



**代码运行结果是3，与c代码运行结果一致**

**2 . horspool**

**$a0对应变量i**

**$t4对应变量j**

**$t8对应变量cnt**

**过程调用时先把$a0储存的len\_str暂存到$t5,$a0载入1024，用syscall申请512字的空间，返回table首地址在$v0中，本次过程调用中用$t0保存table首地址，$t1也存入table首地址，$t2载入-1，向$t1指向地址写入$t2的值(-1),每次循环$t1加4，指向下一个table[i],$a0每次减4，由于$a0初值1024，实现计数256次，判断$a0大于零进入下一次循环。**

**$t2中存入pattern的首地址，$a0(变量i)置0，每次循环先取出pattern[i]存入$t3,左移2位得到地址偏移量，$t2加1得到下一个pattern[i]地址，$t1中储存从table首地址($t0)和偏移量($t3)加和得到的table[pattern[i]]的地址，把i($a0)存入这个地址，i($a0)自加1，判断i($a0)<** **len\_pattern ($a2)则进入下次循环。**

**i($a0)置为len\_pattern ($a2)-1,判断若len\_str<=i则跳出关于i的循环，若没跳出置j=0，**

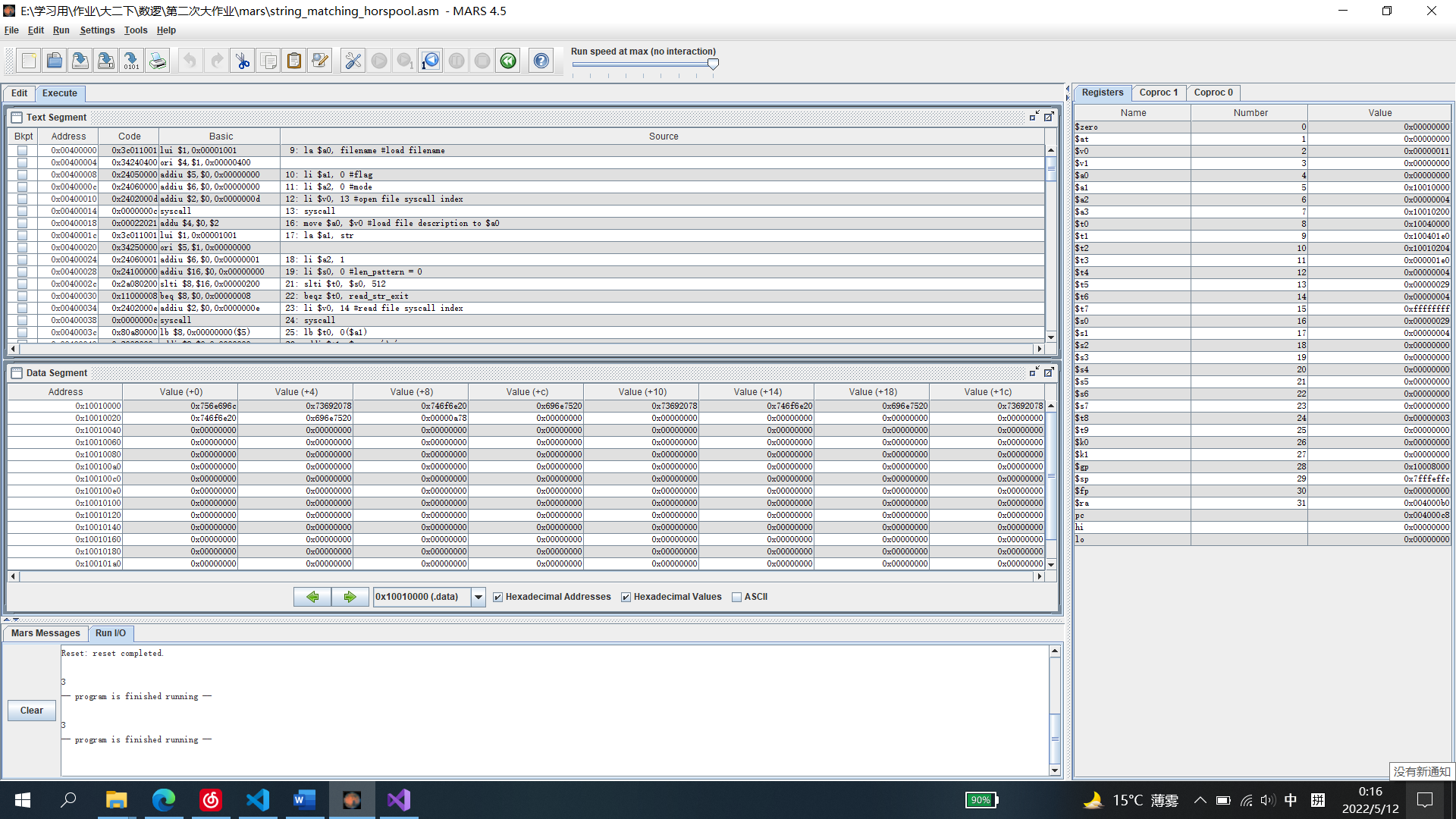
**进入关于j的循环，若len\_pattern<=j则跳出关于j的循环，计算pattern[len\_pattern-1-j]和str[i-j]分别存入$t7和$t6，如果两者不等则跳出关于j的循环，若未跳出j+=1，跳回关于j的循环。**

**跳出j的循环部分：判断j和len\_pattern是否相等，如果相等先执行cnt+=1,否则直接进入if分支**

**If分支：计算table[str[i]]存入$t6中，$t6自加1得到table[str[i]]+1，$t7存入len\_pattern-1-j，判断如果len\_pattern-1-j<table[str[i]]+1则进入else分支，若未进入else分支，更新i($a0)的值为i+len\_pattern-(table[str[i]]+1)，并跳回关于i的循环开始。**

**else分支：i+=1，跳回于i的循环开始。**

**跳出关于i的循环：恢复$a0的值len\_str，$v0存入(cnt)$t8并返回**



**代码运行结果是3，与c代码运行结果一致**

**3 . kmp**

**$t0对应变量i**

**$t1对应变量j**

**$t3对应变量cnt**

**过程调用时先把i,j置0，在$t2中保存$a0的值(len\_str),用syscall申请len\_pattern\*4的空间，$v0返回next的首地址，恢复$a0的值后用$t2储存next的首地址，调用next\_gen进程。**

**next\_gen进程：$t4对应变量i，$t5对应变量j，判断len\_pattern是否等于零，若等于零跳到return1分支，置$v0为1返回，否则继续执行，置next[0]=0,进入关于i的循环，取pattern[i]和pattern[j]判断是否相等，不等跳到if\_gen分支，相等继续执行，**

**更新next[i]=j+1，i+=1，j+=1，跳回关于i的循环开始**

**if\_gen分支：判断j和0大小，j<0则进入else\_gen分支，否则继续执行，更新j = next[j - 1]并跳回关于i的循环开始**

**else\_gen分支：更新next[i]=0,i+=1并跳回关于i的循环开始**

**return0分支：$v0置1，返回**

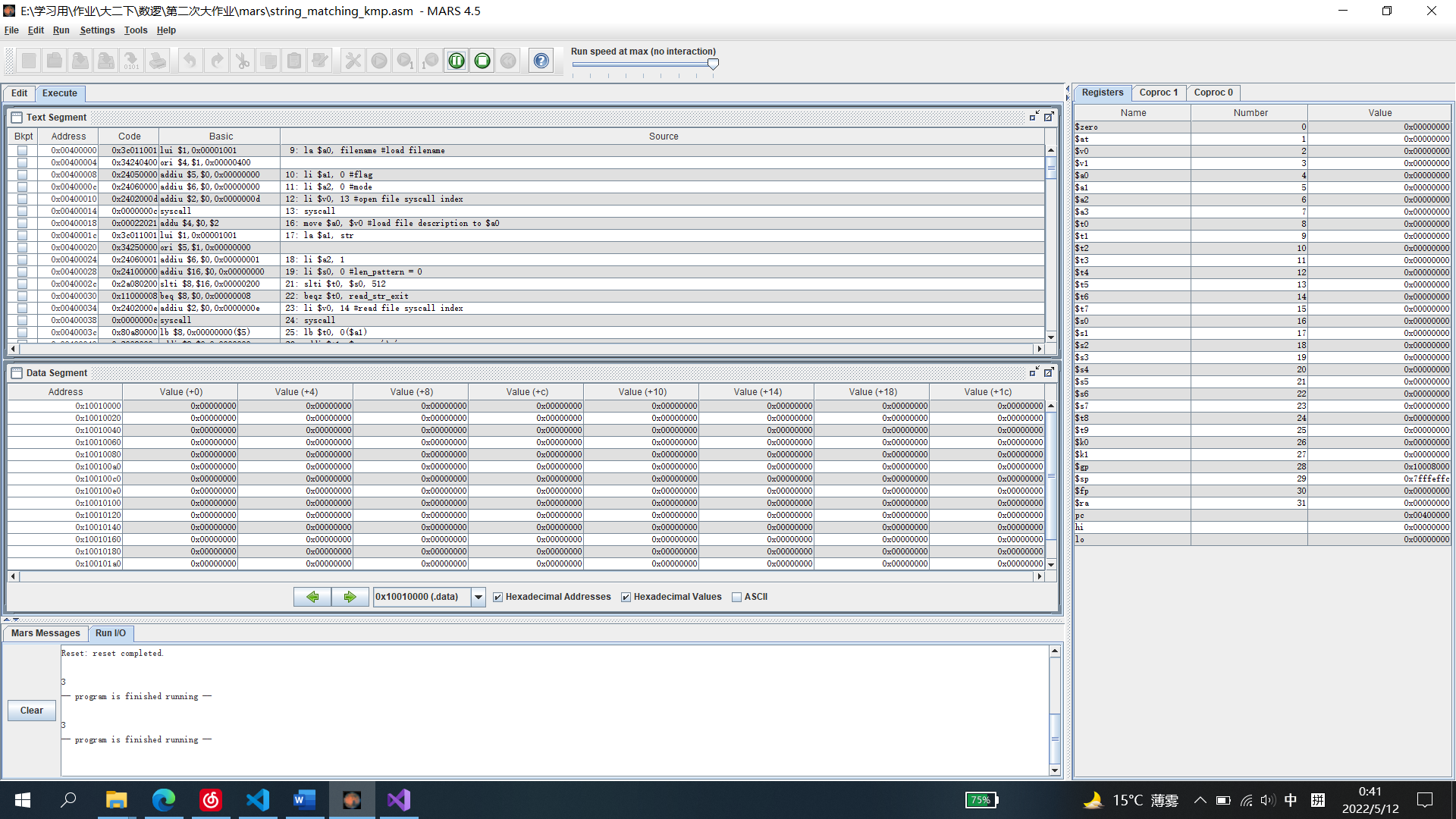
**回到kmp进程，进入关于i的循环，判断len\_str($a0)<=i($t0)则跳出关于i的循环,分别存str[i]和pattern[j]在$t4和$t5中，如果两者不等跳至else\_0分支，否则继续执行，存len\_pattern-1在$t4中，如果j($t1)和len\_pattern-1不等跳转到else\_2分支，否则继续执行，cnt($t3)+=1，更新j=next[len\_pattern-1]，i+=1，跳回关于i的循环开始**

**else\_2分支：更新i+=1，j+=1，跳回关于i的循环开始**

**else\_0分支：判断j<=0则跳转至else\_1分支，否则继续执行，更新j=next[j-1], 跳回关于i的循环开始**

**else\_1分支：更新i++，跳回关于i的循环开始**

**exit\_i分支：更新$v0的值为$t3(即cnt)，返回**



**代码运行结果是3，与c代码运行结果一致**