# 汇编第二次大作业

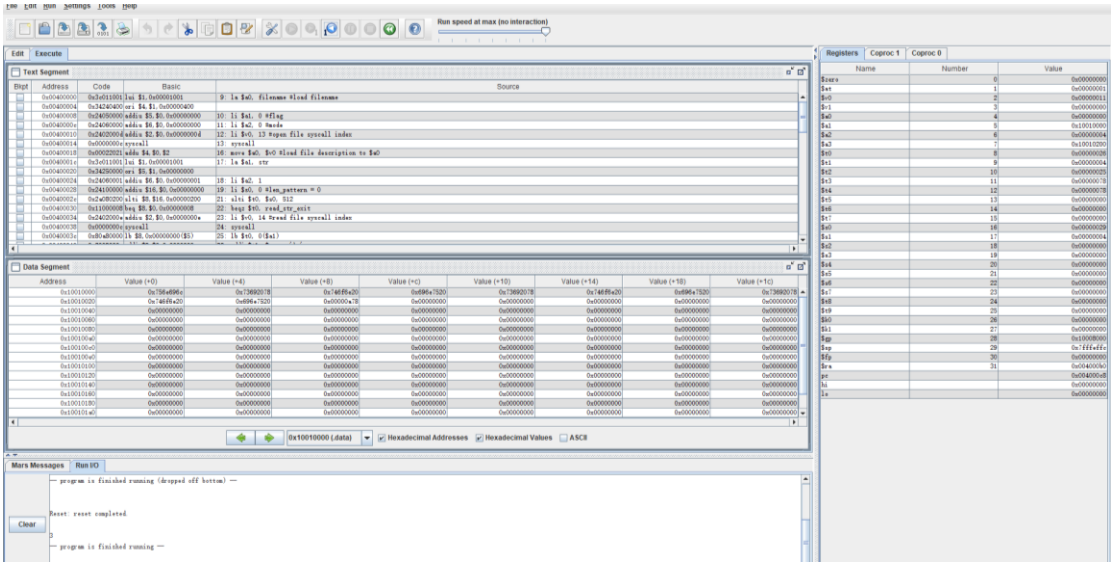
1 . brute-force

$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。

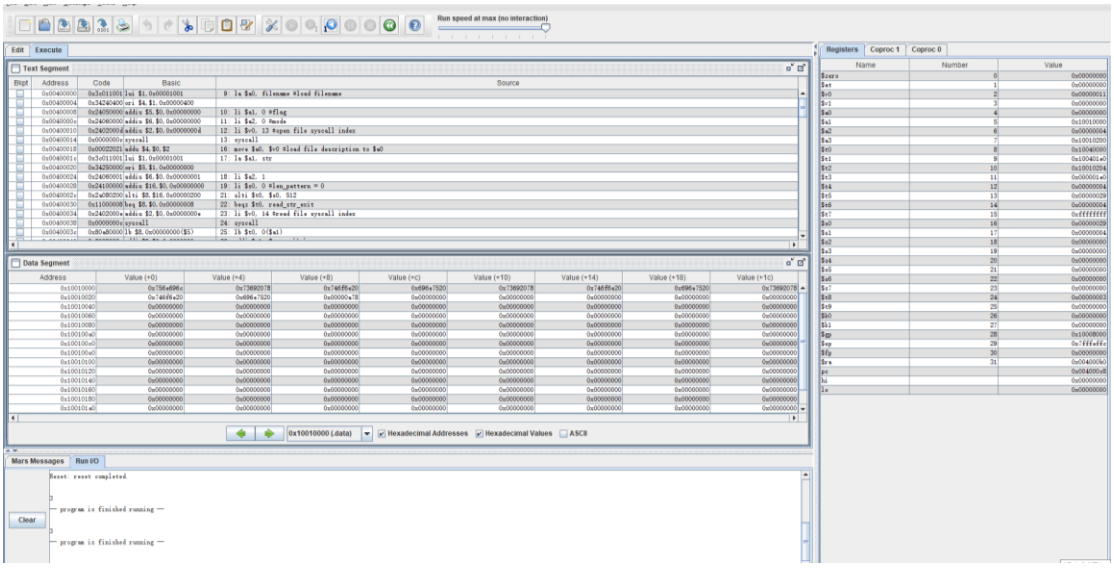
跳出 j 的循环部分：判断 j 和 len_pattern 是否相等，如果相等先执行 cnt+=1,否则直接

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

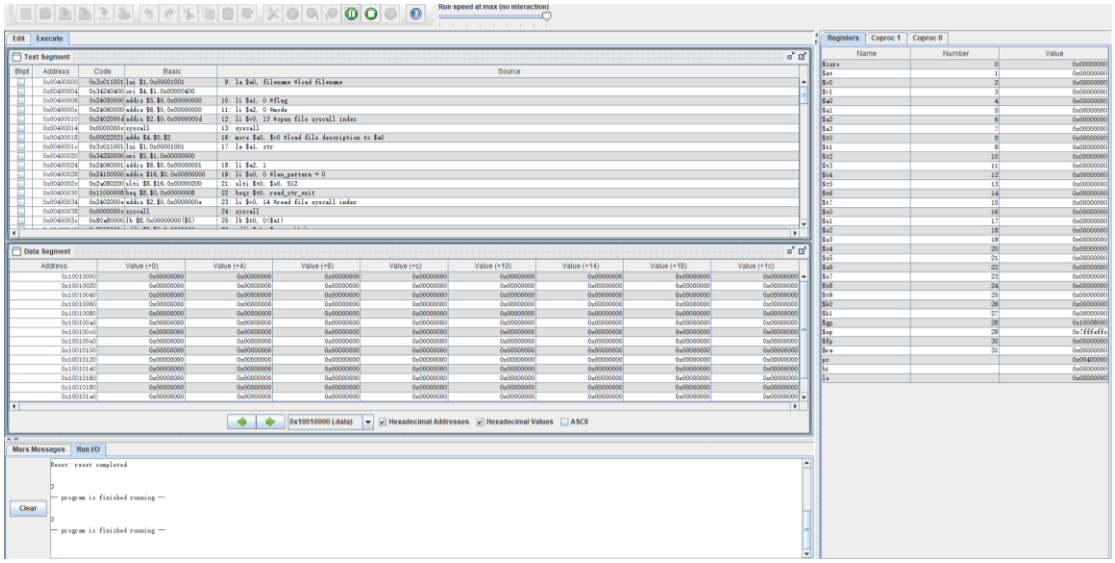
回到 kmp 进程，进入关于 i 的循环，判断 len_str($a0)<=i($t0)则跳出关于 i 的循环,分别存 str[i]和 pattern[j]在$t4 和$t5 中，如果两者不等跳至 else_0 分支，否则继续执行，存 len_pattern-1 在$t4 中，如果 j($t1)和 len_pattern-1 不等跳转到 else_2 分支，否则继续执行，cnt($t3)+=1，更新 j=next[len_pattern-1]，i+=1，跳回关于 i 的循环开始

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