|  |
| --- |
| *Министерство образование и науки*  *Российской федерации*  Федеральное агентство по образованию  Курганский государственный университет |
| Кафедра «Безопасность информационных и автоматизированных систем» |
| Лабораторная работа № 2:  **«Измерение времени выполнения команд»** |
| Выполнил \_\_\_\_ Бутенко А.Ю.  Принял \_\_\_\_\_\_ Стукало В.А. |
|  |
| Курган 2016 |

Цель работы:

- углубление знаний программно-аппаратной организации IBM PC и

ее службы времени;

- измерение времени выполнения команды;

- закрепление навыков программирования на языках Assembler.

**Ход работы:**

Основная задача состоит в том, чтобы измерить время исполнения операции. Для этого необходимо исключить издержки на поддержание цикла и повысить точно измерения (увеличить количество итераций).

Для измерения времени воспользуемся прерыванием системного таймера. Перезапишем его так, чтобы иметь возможность каждый тик таймера инкрементировать счетчик. Для этого воспользуемся макросом.

1. macro set\_timer handler,first\_vector,second\_vector *; макрос реализующий вызов моей функции каждый тик системного таймера*
2. **{**
3. **push** **eax**,**es**
4. **cli**
5. clr **eax**
6. **mov** **es**,**ax**
7. **mov** **eax**,**[es**:8\*4**]**
8. **mov** **dword[**first\_vector**]**,**eax**
9. **mov** **eax**,**[es**:255\*4**]**
10. **mov** **dword[**second\_vector**]**,**eax**
11. **mov** **eax**,**dword[**first\_vector**]**
12. **mov** **[es**:255\*4**]**,**eax**
13. **mov** **ax**,handler
14. **mov** **[es**:8\*4**]**,**ax**
15. **mov** **[es**:8\*4+2**]**,**cs**
16. **sti**
17. **pop** **es**,**eax**
18. **}**

Этот макрос перезаписывает прерывание системного таймера моим обработчиком, код которого приведен ниже.

1. time\_counter: *; последовательность действий внутри таймера (инкрементируется счетчик) и посылается команда конца прерывания, для корректной работы компьютера*
2. **push** **eax**,**ebx**,**edx**,**ds**,**es**,**esi**
3. set\_seg **ds**,data\_seg
4. set\_seg **es**,data\_seg
5. **mov** **byte[**flag**]**,1
6. **inc** **dword[**counter**]**
7. **int** 0FFh
8. **mov** **al**,20h
9. **out** 20h,**al**
10. **pop** **esi**,**es**,**ds**,**edx**,**ebx**,**eax**
11. **Iret**

Здесь каждый тик таймера в переменную counter инкрементируется единица для того, чтобы я потом мог вычислить количество тиков, прошедшее с момента очистки счетчика ну и вычислить время работы этого промежутка программы.

Для вычисления времени работы возьмем 2 отрезка кода: first\_count и second\_count. Один из них замерит время работы цикла из 100 миллионов итераций с командой сложения внутри.

1. floop:
2. **inc** **ecx**
3. **add** **ebx**, **ebx**
4. **cmp** **ecx**, 0x5f5e100 *; 0x5f5e100 = 100 000 000*
5. **jb** floop *; получаем 100 млн итераций*

Весь код подпрограммы выглядит так:

1. first\_count:
2. **mov** **ecx**, 0
4. set\_timer time\_counter,**bp**-8,**bp**-12 *; интегрируем функцию счета в системный таймер*
6. **mov** **[**counter**]**, 0 *; очищаем счетчик*
8. floop:
9. **inc** **ecx**
10. **add** **ebx**, **ebx**
11. **cmp** **ecx**, 0x5f5e100 *; 0x5f5e100 = 100 000 000*
12. **jb** floop *; получаем 100 млн итераций*
14. *; mov eax, [counter]*
15. *; print\_eax*
17. reset\_timer **bp**-8,**bp**-12 *; возвращаем счетчик в исходное состояние*
19. **mov** **eax**, **[**counter**]**
20. **mov** **dword[bp**-204**]**, **eax** *; заносим в стек значение счетчика*
21. **mov** **dword[bp**-208**]**, 55  *; заносим в стек значение времени для одного тика системного таймера (55 ms)*
22. **mov** **dword[bp**-212**]**, 100
24. finit
25. fild **dword[bp**-212**]**
26. fild **dword[bp**-208**]**
27. fild **dword[bp**-204**]**
28. fmul **st0**, **st1**
29. fdiv **st0**, **st2**
30. fist **dword[bp**-300**]**
31. fwait
32. *; умножаем количество тиков на время одного тика - получаем время, затраченное на все итерации*
33. *; однако, в процессе вычисления получилось число в миллисекундах, его нужно умножить на миллион*
34. *; и разделить на количество итераций (100 миллионов), значит мы можем просто разделить число на 100*
35. *; ответ, полученный после вычислений будет равен времени затраченному на исполнение одной итерации цикла в наносекундах*
37. **mov** **eax**,**dword[bp**-300**]**
38. set\_seg **ds**, data\_seg
39. print\_eax
40. **ret**

Здесь мы замеряем время работы этого цикла с помощью метода, описанного выше. Дальше умножаем количество тиков на время одного тика (55 ms) и делим на 100. Связано это с тем, что время мы получаем в ms, а делить будем на 100 миллионов. Поэтому для большей точности переводим число в ns, умножая на миллион. Умножаем на миллион и делим на 100 миллионов, получается достаточно разделить число на 100. Вывод сохраняем в стеке.

Вторая подпрограмма замеряет тот же цикл из 100 миллионов итераций, только без команды внутри. Необходимо это для того, чтобы учесть все издержки, необходимые для работы цикла.

1. second\_count: *; аналогично для второго измерения, только без команды внутри цикла*
2. **mov** **ecx**, 0
4. set\_timer time\_counter,**bp**-8,**bp**-12 *; интегрируем функцию счетчика*
6. **mov** **[**counter**]**, 0 *; очищаем счетчик*
8. sloop:
9. **inc** **ecx**
10. *; add ebx, ebx*
11. **cmp** **ecx**, 0x5f5e100
12. **jb** sloop *; цикл из 100 миллионов итераций*
14. *; mov eax, [counter]*
15. *; print\_eax*
17. reset\_timer **bp**-8,**bp**-12 *; сброс счетчика*
19. **mov** **eax**, **[**counter**]**
20. **mov** **dword[bp**-204**]**, **eax**
21. **mov** **dword[bp**-208**]**, 55
22. **mov** **dword[bp**-212**]**, 100
24. finit
25. fild **dword[bp**-212**]**
26. fild **dword[bp**-208**]**
27. fild **dword[bp**-204**]**
28. fmul **st0**, **st1**
29. fdiv **st0**, **st2**
30. fist **dword[bp**-304**]**
31. fwait
32. *; вычисления по той же логике, умножаем количество тиков на их время и делим на 100*
34. **mov** **eax**,**dword[bp**-304**]**
35. set\_seg **ds**, data\_seg
36. print\_eax
37. **ret**

Для чистоты эксперимента запустим программу, реализующую вышеописанный функционал 5 раз.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| W cmd (ns) | 52 | 52 | 52 | 52 | 52 |
| W/O cmd (ns) | 39 | 39 | 39 | 39 | 39 |
| Cmd (ns) | 13 | 13 | 13 | 13 | 13 |

Из измерения видно, что время работы одной команды сложения занимает 13 наносекунд.

**Вывод:** мной было измерено время работы одной команды. Результат – 13 наносекунд.