АВТОМАТИЧЕСКОЕ ТЕСТИРОВАНИЕ ПРОЦЕДУР НА ЯЗЫКЕ АССЕМБЛЕРА

М.Д. Новиков, *канд. физ.-мат. наук, старший научный сотрудник*
Московский государственный университет им. М.В. Ломоносова
(Россия, г. Москва)

DOI:10.24412/2500-1000-2022-3-2-144-147

Аннотация. *В статье описывается система, предназначенная для автоматического тестирования программ на языке Ассемблера, выполняемых студентами первого курса факультета Вычислительной математики и кибернетики МГУ им. М.В. Ломоносова в рамках практикума на ЭВМ. Рассматриваются особенности тестирования ассемблерных процедур: передача параметров процедуре, выполнение процедуры и получение результата. Система разрабатывается на факультете ВМК с 2020 года.*

Ключевые слова*: язык Ассемблера, процедуры, функции, тестирующие системы, программирование, информатика.*

Язык программирования Ассемблер изучается студентами первого курса факультета ВМК МГУ во втором семестре. В качестве компьютерной версии языка взята 32-битная версия MASM 6.14. Задания по практикуму берутся, в основном из книг [2] и [3], в которых собрано много задач на темы, охватывающие все основные конструкции языка Ассемблер. Большое внимание уделяется написанию ассемблерных процедур. Возникает задача проверки корректного составления процедур, т.е. соответствия их заданным требованиям. Описываемая в статье система автоматического тестирования программ (далее – САТП) позволяет проверять правильность составления и выполнения ассемблерных процедур в автоматическом режиме.

Постановка задачи.

Каждая процедура должна удовлетворять стандартным соглашениям о связях [1]. Эти соглашения включают в себя следующие пункты.

1) Фактические параметры должны передаваться процедуре через стек.

2) Все используемые процедурой регистры процессора должны сохраняться: если какой-либо регистр используется в теле процедуры, то его значение должно быть занесено в стек в начале выполнения процедуры и извлечено из стека по окончании выполнения процедуры.

3) Внутри процедуры недопустимы никакие внешние объекты (переменные, константы или метки).

4) По окончании выполнения процедуры стек должен очищаться от фактических параметров.

5) Если процедура по условию задачи должна вырабатывать некоторое значение (т.е. если она является аналогом функции в языках высокого уровня), то результат должен возвращаться через регистры al (байт), ax (слово) и eax (двойное слово).

Примечание. В некоторых случаях допускается передача фактических параметров не через стек, а через регистры процессора. Это отдельно оговаривается для каждой конкретной задачи.

Задания в [1] и [2] формулируются следующим образом.

1. Требуется составить процедуру на языке Ассемблера, удовлетворяющую заданным свойствам.

2. Требуется написать полную программу на языке Ассемблера, внутри которой описать процедуру с заданными свойствами.

САТП позволяет тестировать как отдельную процедуру, так и полную программу с содержащейся внутри неё процедурой. В первом случае к процедуре добавляется специальный блок данных и команд, дополняющий ее до полной программы, которая затем тестируется. Этот блок содержит описания необходимых переменных, команды ввода исходных дан-

ных, команды передачи фактических параметров процедуре, ее вызова и вывода результатов ее работы. Во втором случае отдельно тестируется полная программа и входящая в нее процедура, которая выделяется из программы и дополняется до полной программы как в предыдущем случае.

Проверка требований, предъявляемых к процедуре.

Проверка правильности передачи фактических параметров процедуре и проверка результата затруднений не вызывает. Для этого достаточно выполнить команды Push <фактические параметры> перед вызовом процедуры и сравнить выдаваемый процедурой результат с правильным.

Проверку сохранения регистров процессора в теле процедуры САТП производит следующим образом. Перед командой вызова процедуры в тестируемую программу вставляются команды занесения в регистры процессора случайных значений. По окончании работы процедуры проверяются значения всех регистров процессора и, в случае несовпадения с заданными ранее значениями, выдается сообщение об ошибке.

Для проверки отсутствия внешних объектов (переменных, констант и меток) выполняются следующие действия. Ко всем именам в блоке, дополняющем процедуру до полной программы, добавляются случайные символы - разные для каждого тестирования. Тем самым, если какой-либо внешний объект присутствует в теле процедуры, то неизбежно возникает ошибка на этапе компиляции (неописанное имя).

Для проверки очистки стека от фактических параметров значение вершины стека (содержимое регистра esp) запоминается перед занесением в него фактических параметров процедуры и ее вызовом. По окончании выполнения процедуры проверяется, что значение esp не изменилось.

Актуальность автоматического тестирования процедур.

Если неверная передача фактических параметров процедуре и неправильное ее исполнение легко выявляются при тестировании вручную, то остальные требования выявить труднее. Ошибки, связанные с отсутствием сохранения регистров и очистки стека от фактических параметров не фиксируются ни на этапе компиляции, ни на этапе исполнения ассемблерной программы (в отличие, например, от синтаксических ошибок или ошибок периода выполнения). При ручном тестировании процедуры такие ошибки трудно заметить (особенно, если процедура содержит много команд). Однако в реальных программах подобные ошибки в процедурах часто приводят к трудно выявляемым ошибкам на этапе исполнения программ – случайным образом меняется содержимое некоторого регистра процессора, переполняется стек и проч. Поэтому важно приучить студента к корректному написанию программ, что как раз и достигается при автоматическом тестировании с помощью САТП.

Пример тестирования процедуры.

В качестве примера рассмотрим задачу 11.8 из [2]. Требуется описать процедуру Setabs, которой передается адрес некоторого знакового байта памяти, и которая заменяет его значение на абсолютную величину.

Возможный вид процедуры Setabs:

```
Setabs proc
push ebp
mov ebp,esp
push ebx
mov ebx, [ebp+8] ; требуемый адрес - в [ebp+8], так как по адресу [ebp] находится само значение регистра ebp после выполнения команды mov ebp,esp, а в [ebp+4] находится адрес возврата из процедуры.
cmp [ebx], byte ptr 0
jge m5
neg byte ptr[ebx]
m5: pop ebx
pop ebp
ret 4          ; очистка стека от фактического параметра
Setabs      endp
```

САТП добавляет к процедуре блок данных и команд, дополняющих его до полной программы.

До тела процедуры добавляются следующие команды.

include console.inc ; подключение библиотеки команд ввода-вывода.

.data ; описание необходимых переменных.

X! db ? ; при формировании программы все символы ‘!’ заменяются на пару случайных цифровых символов (одну и ту же во всей программе).

Q! dd ?

.code

Сюда вставляется тестируемая процедура.

После тела процедуры добавляются следующие команды.

Start:

Inint X! ;ввод исходных данных

k!=! ; константа, используемая для присвоения случайных значений регистрам.

for R,<eax,edi,esi,ebp,ebx,ecx,edx>

k!=k!+1

mov R,k!

endm

mov Q!,esp ; запоминается вершина стека.

push offset X!

call <name> ; слово <name> заменяется на фактическое имя процедуры.

outint X! ; вывод результата

cmp Q!,esp ; проверяется, что регистр esp не изменился, т.е. что фактические параметры удалены из стека

je <1> ; все метки заменяются специальными именами: L0, L1 и т.д.

outchar 6 ; в случае изменения значения регистра esp выводится код 6; его появление в строке результата означает неверное значение этого регистра.

exit <1>:

k!=k!-7

for R,<eax,edi,esi,ebp,ebx,ecx,edx>

k!=k!+1

cmp R,k!

jne <0> ; проверяется, что регистры не изменились после выполнения процедуры

endm

exit

<0>:outchar 5 ; код 5 выводится в случае, если значение какого-либо регистра не было сохранено в процедуре (изменилось по окончании ее работы).

Exit

end Start

Сформированная программа может выглядеть так:

```
include console.inc
.data
X23 db ?
Q23 dd ?
.code
Setabs proc
push ebp
mov ebp,esp
push ebx
mov ebx, [ebp+8]
cmp [ebx], byte ptr 0
jge m5
neg byte ptr[ebx]
m5: pop ebx
pop ebp
ret 4
Setabs endp
Start:
inint X23
k23=23
for R,<eax,edi,esi,ebp,ebx,ecx,edx>
k23=k23+1
mov R,k23
endm
mov Q23,esp
push offset X23
call Setabs
outint X23
cmp Q23,esp
je L1
outchar 6
exit
L1:
k23=k23-7
for R,<eax,edi,esi,ebp,ebx,ecx,edx>
k23=k23+1
cmp R,k23
jne L0
endm
exit
L0:outchar 5
exit
end Start
```

Эта программа компилируется и выполняется на заранее подготовленных наборах исходных данных. Выдаваемые результаты сравниваются с правильными.

Заключение и выводы.

Описанная САТП используется для автоматической проверки студенческих программ с 2020 г. Она показала свою эффективность по сравнению с тестированием

программ вручную. Раньше студенты приносили на проверку программы, содержащие большое количество ошибок – как синтаксических, так и логических. Преподавателю приходилось визуально проверять тексты программ и вводить с клавиатуры компьютера большие объемы исходных данных для тестирования программ. Теперь же САТП выполняет всю эту работу автоматически, и студенты приносят на проверку, в основном, уже готовые программы. К настоящему времени составлены тесты примерно к 400 задачам из [1] и [2].

САТП написана в среде DELPHI. Объем архивированного файла вместе с транслятором MASM 6.14 – около 10 MB.

Основные результаты опубликованы в [4] и [5].

Библиографический список

1. Баула В.Г. Введение в архитектуру ЭВМ и системы программирования. Учебно-методическое пособие. – [Электронный ресурс]. – Режим доступа: http://arch32.cs.msu.su (дата обращения: 26.03.2022).

2. Бордаченкова Е.А., Панферов А.А. Задания практикума. 1 курс. – М.: МАКС Пресс, 2016. – 48 с.

3. Бордаченкова Е.А. Задачи и упражнения по языку Ассемблера MASM. – М.: МАКС Пресс, 2020. – 92 с.

4. Новиков М.Д. Система автоматического тестирования студенческих программ на языке Ассемблера // Научная конференция «Ломоносовские чтения». Секция Вычислительной математики и кибернетики. – М.: Изд-во Московского ун-та, 2021. – С. 116-117.

5. Новиков М.Д. Автоматическое тестирование студенческих программ // Всероссийская научная конференция «Математические основы информатики и информационно-телекоммуникационных систем». Сборник Трудов. –Тверь: Изд-во Тверского государственного университета, 2021. – С. 235-240.

AUTOMATIC TESTING OF ASSEMBLY PROCEDURES

M.D. Novikov, *Candidate of Physical-Mathematical Sciences, Senior Researcher*
Lomonosov Moscow State University
(Russia, Moscow)

Abstract. *A system destined for an automatic testing of Assembly programs is described in the article. The system has been developing at the Faculty of Computational Mathematics and Cybernetic of Moscow State University from 2020. It tests tasks performed by first-year students of the faculty. Some aspects of testing Assembly procedures are presented in the article – passing actual parameters to a procedure, its performing and output the results.*

Keywords*: assembly language, procedures, functions, testing systems, programming, computer science.*