Генератор случайных чисел (asm x86)

Материал из Национальной библиотеки им. Н. Э. Баумана  
Последнее изменение этой страницы: 19:48, 9 августа 2015.

Мультипликативный конгруэнтный метод генерации последовательности случайных чисел

Мультипликативный конгруэнтный метод задает последовательность неотрицательных целых чисел Xi(Xi<m){\displaystyle X\_{i}(X\_{i}<m)}, получаемых по формуле:

Xn+1=(a⋅Xn)modm{\displaystyle X\_{n+1}=(a\cdot X\_{n})\mod m}

На значения накладываются ограничения:

* X0{\displaystyle X\_{0}} - нечетно;
* a=52p+1(p=0,1,2,…){\displaystyle a=5^{2p+1}(p=0,1,2,\ldots )} или a=2m+3(m=3,4,5,…){\displaystyle a=2^{m}+3(m=3,4,5,\ldots )} - обе эти записи означают, что младшая цифра a{\displaystyle a} при представлении a{\displaystyle a} в восьмеричной системе счисления должна быть равна 3 или 5 (проще говоря, остаток от деления a{\displaystyle a} на 8 должен быть равен 3 или 5);
* m=2l(l>4){\displaystyle m=2^{l}(l>4)}

При соблюдении этих ограничений длина периода будет равна m/4{\displaystyle m/4}

;+-------------------------------------------------------------------------+

;| Программа: rand\_mult\_cong\_l.asm. Генератор линейной (мультипликативной) |

;| конгруэнтной последовательности случайных чисел (c=0) |

;+-------------------------------------------------------------------------+

;| Вход: X0. a. m. |

;+-------------------------------------------------------------------------+

;| Выход: dl - значение очередного случайного числа. |

;+-------------------------------------------------------------------------+

.data

m db 128

а db 11

X db 3 ; начальное значение

.code

;------ первое число в последовательности х=3

cycl: mov al.x ; вычисляем очередное случайное число Х=(а\*Х) mod m

mul a ; а\*х в ah:al

div m ; в ah случайное число

mov x. ah

;------ вывод в файл - командная строка rand\_mu1t\_cong.exe > p.txt

mov dl. ah

mov ah. 02

int 21h

jmp cycl

end\_cycl: ;...

Если m{\displaystyle m} является степенью 2, как в данном случае, то вместо команды DIV можно использовать две команды сдвига.

;+---------------------------------------------------------------------------+

;| Программа: rand\_mult\_cong\_l.asm. Генератор линейной (мультипликативной) |

;| конгруэнтной последовательности случайных чисел (c=0) |

;+---------------------------------------------------------------------------+

;| Вход: X0. a. m. - в соответствии с указанными ограничениями. |

;+---------------------------------------------------------------------------+

;| Выход: dl - значение очередного случайного числа. |

;+---------------------------------------------------------------------------+

cycl:

;------ вычисляем очередное случайное число Х=(а\*Х) mod m

mov al.x

mul а ; а\*х в ah:al

shrd ax. ax. cl

xor al. al

rol ax. cl ; в al случайное число

;------ вывод в файл - командная строка rand\_mu1t\_cong.exe > p.txt

;...

Полный пример этой программы (rand\_mult\_cong\_2.asm) вы найдете среди файлов, прилагаемых к книге.

Используя эти программы, можно получить последовательность случайных чисел, содержащую 32 значения — это ее период. Чтобы увеличить период, необходимо каким-либо способом сгенерировать значения a{\displaystyle a} или x{\displaystyle x}, удовлетворяющие приведенным выше ограничениям. Так, значение a{\displaystyle a} можно вычислить, используя фрагмент:

.data

divider db 8

.code

;------ вычисляем a исходя из соотношения, а mod 8 - 5

; одним из способов получить значение а (m>=а)

; удовлетворяем условию а mod 8=5

m2: mov al. а

xor ah. ah

div divider

cmp ah. 5 ; остаток равен 5?

je m1

cmp ah. 3 ; остаток равен 3?

je m1

inc a

jmp m2

m1: ; теперь a найдено до конца

;...

Изменить (увеличить) период можно, корректируя значение m{\displaystyle m}, для чего требуется исправить соответствующие команды в программах rand\_mult\_cong\_l.asm и rand\_mult\_cong\_2.asm, ориентированные на определенную разрядность регистров. Существует другая возможность увеличения периода — использование смешанного конгруэнтного метода генерации последовательности случайных чисел.

Смешанный конгруэнтный метод генерации последовательности случайных чисел

Соотношение смешанного конгруэнтного метода выглядит так:

Xn+1=(a⋅Xn+c)modm{\displaystyle X\_{n+1}=(a\cdot X\_{n}+c)\mod m}

где n≥0{\displaystyle n\geq 0}.

При правильном подборе начальных значений элементов, кроме увеличения периода последовательности случайных чисел, уменьшается корреляция (зависимость) получаемых случайных чисел.

На значения накладываются ограничения:

* X0≥0{\displaystyle X\_{0}\geq 0};
* a=21+1{\displaystyle a=2^{1}+1}, где l≥2{\displaystyle l\geq 2}
* c>0{\displaystyle c>0} взаимно просто с m{\displaystyle m} (это выполнимо, если c{\displaystyle c} — нечетно, а m=2p{\displaystyle m=2^{p}}, где p≥2{\displaystyle p\geq 2}
* m=2p(p≥2){\displaystyle m=2^{p}(p\geq 2)} и m{\displaystyle m} кратно 4.

;+---------------------------------------------------------------------------+

;| Программа: rand\_mix\_cong\_l.asm. Генератор линейной (смешанной) |

;| конгруэнтной последовательности случайных чисел (c>0) |

;+---------------------------------------------------------------------------+

;| Вход: X0. a. c. m. - в соответствии с указанными ограничениями. |

;+---------------------------------------------------------------------------+

;| Выход: dl - значение очередного случайного числа. |

;+---------------------------------------------------------------------------+

.data

m db 128 ; 128=2^7

а db 9

x db 3 ; начальное значение

с dw 3

.code

mov cl. 7 ; значение степени m = 27 в cl

;------вычисляем очередное случайное число Х=(а\*Х) mod m.

; первое число в последовательности х=3

сусl: mov al. x

mul a ; а\*х в ah:al

add ax. c

shrd ax. ax. cl

xor al. al

rol ax. cl ; в al случайное число

;------вывод в файл - командная строка rand\_mult\_cong.exe > p.txt

;...

end\_cycl: ;...

Величина периода случайной последовательности, получаемой с помощью данной программы, составляет 128. Сегмент кода программы rand\_mix\_cong\_l.asm можно оформить в виде процедуры. Начальное значение можно выбирать двумя способами: задавать константой в программе или генерировать случайным образом. В последнем случае достаточно опереться на такты системного таймера, как в следующей макрокоманде:

;+---------------------------------------------------------------------------+

;| Макрокоманда: rCMOS. Чтение значений CMOS |

;+---------------------------------------------------------------------------+

;| Вход: al - адрес ячейки, значение которой читаем. |

;+---------------------------------------------------------------------------+

;| Выход: al - прочтенное значение. |

;+---------------------------------------------------------------------------+

rCMOS macro

out 70h. al

xor al. al

in al. 71h ; вводим в регистр AL из порта

; значение ячейки CMOS

endm

Применение — получить значение секунд из CMOS для x\_start:

.code

mov al. 0h

rCMOS

mov x.al ; x=x\_start

;...

Таким способом можно получить начальное значение из диапазона 0-59. Для получения большего по величине начального значения подходит величина размером 32 бита из области данных BIOS по адресу 0040:006с. Здесь содержится счетчик прерываний от таймера. Извлечь это значение можно при помощи следующего программного фрагмента:

push ds

push word ptr 40h

pop ds

mov eax. dword ptr ds:006ch

pop ds

Заменяя команду MOV командой MOV AX, word ptr ds:006ch или MOV AL, byte ptr ds:006ch, можно использовать младшие 8 или 16 битов значения из этой области BIOS. Команда MOV AL, byte ptr ds:006ch позволяет случайным образом загрузить в регистр AL значение из диапазона 0-0ffh.

При работе под Windows в качестве начального можно использовать значения из счетчика меток реального времени TSC [39].

Попытки создать программный датчик случайных чисел без опоры на какие-либо теоретические выкладки обречены на провал. Рассмотрим еще несколько способов генерации случайных чисел.

Аддитивный генератор случайных чисел

Генератор, формирующий очередное случайное число в соответствии с отношением

Xn+1=(Xn+Xn−k)modm{\displaystyle X\_{n+1}=(X\_{n}+X\_{n-k})\mod m}

называется аддитивным.

D трехтомнике Кнута [5] обсуждаются подобные генераторы и рекомендован следующий вариант этой формулы:

Xn+1=(Xn−24+Xn−55)modm{\displaystyle X\_{n+1}=(X\_{n-24}+X\_{n-55})\mod m}

Здесь n≥55,m=2l,X0,…,X54{\displaystyle n\geq 55,m=2^{l},X\_{0},\ldots ,X\_{54}}— произвольные числа, среди которых есть и нечетные. При этих условиях длина периода последовательности равна 2l−1(255−1){\displaystyle 2^{l}-1(2^{55}-1)}.

Для генерации первых 55 случайных чисел можно использовать один из рассмотренных выше генераторов. Возьмем датчик линейной (смешанной) конгруэнтной последовательности случайных чисел (c>0{\displaystyle c>0}).

;+-----------------------------------------------------------------------------+

;| Программа; rand\_add.asm. Аддитивный генератор случайных чисел. |

;+-----------------------------------------------------------------------------+

;| Вход; Х0. а. с. m. |

;| случайная последовательность длиной 55 значений, получаемая с помощью |

;| программы генерации высокослучайных двоичных наборов (rand\_mix\_cong l.asm). |

;| N = 700 - количество элементов в последовательности + 1. |

;| L=7 - значение степени m=2^7 |

;+-----------------------------------------------------------------------------+

;| Выход: dl - Значение очередного случайного числа. |

;+-----------------------------------------------------------------------------+

.data

N = 700 ; число элементов в последовательности + 1

L = 7 ; значение степени m=2^7

m db 128 ; 128 = 2^7

mm dw 256 ; 256 = 2^8

а db 9

с dw 3

x db 3. N dup (Offh) ; массив значений x (начальное значение

; равно 3) длиной N + 1

.code

;------ далее фрагменты из rard\_mix\_cong\_l.asm

xor si. si

mov ecx, 54 ; счетчик цикла для формирования

; начальной последовательности

mov al. x ; первое число в последовательности (х=3)

;------вычисляем очередное случайное число Х=(а\*Х) mod m

cycl: mul a ; a\*x в ah:al

add ax. с

shrd ax. ax. L ; L - значение степени m=2^7

xor al. al

rol ax. L ; в al случайное число

;------ вывод в массив х и файл - командная строка

; rand\_mult\_cong.exe > p.txt

inc si

mov x[si]. al

mov dl. al

mov ah. 02

int 21h

loop cycl

;------ далее продолжаем формирование случайной последовательности.

; но уже аддитивным методом

mov ecx. N - 55

cycl2: inc si

xor dx. dx

mov al. x[si-24]

mov dl. x[si-55]

add ax. dx

xor dx. dx

div mm

mov x[si]. dl

mov ah. 02

int 21h

loop cycl2

exit:

;...

Судя пo результатам, этот метод достаточно хорош. В частности, он неплох тем, что позволяет задавать длину последовательности без оглядки на значение m{\displaystyle m}.

Следующий рассматриваемый нами датчик можно использовать в программах на языке ассемблера для получения случайной последовательности 0 и 1.