

УДК681.324

А.А. Никандров (6 курс, каф. САиУ), В.Е. Куприянов, проф., к.т.н.

ИССЛЕДОВАНИЕ СРЕДСТВ ОБРАБОТКИ ИСКЛЮЧЕНИЙ ПРИ ВЫПОЛНЕНИИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ С ПЛАВАЮЩЕЙ ТОЧКОЙ

При создании любой автоматизированной процедуры, разработчик стремится избежать ошибок, которые являются неизбежными при написании кода.

К категории информационных ошибок можно отнести ряд аппаратных ошибок, связанных с выполнением простых арифметических операций. Чаще всего такие ошибки возникают при попытке деления на ноль, а также при выходе за пределы разрядной сетки. Отсутствие их обработки приводит к немедленному аварийному завершению процесса. Чтобы исключить и/или обработать информационные ошибки, в ряде случаев приходится большую часть кода отводить для всевозможных проверок. В теории все выполняемые действия следует подвергать анализу на получаемый результат, причем до их выполнения.

Для решения проблем, вызываемых информационными ошибками, при создании процедуры следует использовать обработку исключений. Средства обработки исключений нужны для того, чтобы дать возможность в коде, при выполнении которого возникает исключение, которое он не может обработать, передать эту обработку некоторой процедуре. Эта процедура определит тип произошедшей ошибки и выполнит заданные пользователем действия.

Целью данной работы является исследование встроенных в интегрированную среду программирования Visual Fortran средств обработки ситуаций, вызывающих прерывания.

При выполнении вычислений с вещественными числами, необходимо осуществлять контроль возникновения переполнения и исчезновения порядка. Если вещественное число x не равно точно одному из чисел, представляемых в компьютере, то x будет представляться ближайшим к нему числом с плавающей точкой \hat{x} , называемым приближением x . При этом возникает ошибка округления, равная $|x - \hat{x}|$. Результат, если он настолько мал или велик, что не может быть обработан процессором, приводит к переполнению или исчезновению порядка. По умолчанию в случае исчезновения порядка результат заменяется нулем, а переполнения приводят к ошибке выполнения и аварийному завершению программы.

Факт замены значения переменной с очень малого на ноль может иметь очень серьезные последствия. Если эта переменная с новым (нулевым) значением будет использоваться в последующих вычислениях, то они будут далеки от истинных. Если же еще используется деление на эту переменную, то возникнет другая аппаратная ошибка - деление на 0.

Программист может выбрать способ обработки таких исключений, вызывая процедуру SETCONTROLFPQQ и задавая соответствующее значение параметра. Записанный в процессор статус возвращается процедурой GETSTATUSFPQQ библиотеки MSFLIB. При появлении исключения, если оно запрещено, возникает соответствующий ему сигнал прерывания.

По умолчанию сигнал прерывания приводит к завершению программы. Можно заменить встроенный обработчик сигналов прерываний на свой. Для этого необходимо, во-первых, написать функцию, реагирующую на прерывания, и, во-вторых, зарегистрировать ее, с использованием процедуры SIGNALQQ. Эта процедура предназначена для определения самостоятельно написанной процедуры, как процедуры, вызываемой при возникновении прерывания. Внутри этой функции можно задать действия, которые необходимо выполнить при возникновении каждого конкретного типа ошибок. Встроенные в средства обработки позволяют различать следующие типы ошибок:

- недопустимый результат (например, 0/0);
- результат является денормализованным числом;
- произошло деление на 0;

переполнение;
исчезновение порядка;
приближение.

Если обработчик прерываний создан программистом, то в нем можно предусмотреть оператор останова STOP. Если же такой оператор отсутствует, то вычисления будут продолжены. Кроме того, написанный программистом обработчик на новые прерывания реагировать не будет. Если все же есть необходимость продолжить вычисления при возникновении исключения, то нужно, применив SETCONTROLFPQQ, их разрешить. Разрешенное исключение прерываний не порождает, поэтому информацию о месте возникновения и характере исключения можно получить, анализируя результат, возвращаемый GETSTATUSFPQQ, или исследуя значения переменных, его вызвавших. Например, в случае переполнения и режиме разрешения прерываний, аварийная переменная получает значение NaN или #Indefinite. По результатам анализа значения можно принять решение о продолжении или завершении процесса вычислений.

ЛИТЕРАТУРА

1. Бартьев О.В. Фортран для профессионалов. Математическая библиотека IMSL. Москва, изд. Диалог-МИФИ. 2000.
2. Бартьев О.В. Современный Фортран. Москва. Изд. Диалог-МИФИ, 1998.