Collin Gros
09-09-2020
Programming #2

PROGRAMMING #2

We want to prove whether a language does short-circuit evaluation. To do this, we have to write an if statement using a logical and with the first part that is evaluated being false, and monitor the second part to see if it is even touched. The following table summarizes my results from my testing:

Language	**Short-Circuits?**
ADA	NO
Bourne Again Shell	YES
PHP	YES
PERL	YES

ADA

CODE:

```
collin@collin-fx:~/Documents/school/cs-471/prog-2/ada$ cat main.adb
-- collin gros
-- 09-09-2020
--
-- this program tests whether short-circuiting is present in ADA.
-- it uses two functions, one that returns true and the other that
-- returns false (each print whenever they are called) to prove
-- a function call is skipped when evaluating an if statement.
-- (short circuit)


with Ada.text_IO;
use Ada.text_IO;



-- main tests whether ada does short circuit evaluation
procedure main is
                -- retTrue: returns true and prints to stdout "retTrue!"
                function retTrue return Boolean is
                begin
                        Put("retTrue!");
                        New_Line;
                        return TRUE;
                end retTrue;

                -- retFalse: returns false and prints to stdout "retFalse!"
                function retFalse return Boolean is
                begin
                        Put("retFalse!");
                        New_Line;
                        return FALSE;
                end retFalse;
begin
        -- test whether ada does short circuiting with and
        -- if short circuiting, should do retFalse! and nothing else.
        if retFalse and retTrue then
                Put("executed FALSE AND TRUE");
                New_Line;
        end if;

        -- test whether ada does short circuiting with or
        -- if short circuiting, should do retTrue! and nothing else.
        if retTrue or retFalse then
                Put("executed TRUE OR FALSE");
                New_Line;
        end if;
end main;
```

OUTPUT:

```
collin@collin-fx:~/Documents/school/cs-471/prog-2/ada$ ls
ada-compile.txt  main  main.adb  main.ali
collin@collin-fx:~/Documents/school/cs-471/prog-2/ada$ ./main
retFalse!
retTrue!
retTrue!
retFalse!
executed TRUE OR FALSE
collin@collin-fx:~/Documents/school/cs-471/prog-2/ada$
```

The fact that retTrue! and retFalse! were both printed twice indicates that ADA does NOT short circuit conditional statements.

BOURNE AGAIN SHELL (BASH)

CODE:

```
collin@collin-fx:~/Documents/school/cs-471/prog-2/bash$ ls
after-iftest.png  iftest-1.sh  iftest-2.sh
collin@collin-fx:~/Documents/school/cs-471/prog-2/bash$ cat iftest-1.sh
#!/bin/bash
#
# collin gros
# 09-09-2020
#
# this program tests whether the bourne shell will do
# short-circuit evaluation in if statements
#


# makes a file, called-echoTrue, and echos true for substitution later
echoTrue ()
{
        echo true

        touch called-echoTrue
}


# if short circuiting, a file called called-echoTrue will NOT be created
if false && $(echoTrue); then
        exit
fi


collin@collin-fx:~/Documents/school/cs-471/prog-2/bash$ cat iftest-2.sh
#!/bin/bash
#
# collin gros
# 09-09-2020
#
# this program tests whether the bourne shell will do
# short-circuit evaluation in if statements
#


# makes a file, called-echoTrue, and echos true for substitution later
echoTrue ()
{
        echo true

        touch called-echoTrue
}


# will crete a file called called-echoTrue
if true && $(echoTrue); then
        exit
fi
```

OUTPUT:

```
collin@collin-fx:~$ ./iftest-1.sh
collin@collin-fx:~$ ls
Desktop    Downloads    iftest-2.sh  Pictures  Templates
Documents  iftest-1.sh  Music        Public    Videos
collin@collin-fx:~$ ./iftest-2.sh
collin@collin-fx:~$ ls
called-echoTrue  Documents  iftest-1.sh  Music     Public     Videos
Desktop          Downloads  iftest-2.sh  Pictures  Templates
collin@collin-fx:~$
```

The fact that the file was NOT created after running iftest-1.sh indicates that BASH does indeed implement short circuiting.

PHP

CODE:

```
collin@collin-fx:~/Documents/school/cs-471/prog-2/php$ ls
php2.png  php.png  test.php
collin@collin-fx:~/Documents/school/cs-471/prog-2/php$ cat test.php
<?php
# collin gros
# 09-09-2020
# this program proves php short circuits if statements by showing the
# function retTrue() is not called when a conditional of 0 && retTrue().

# returns true, and prints that we are inside
function retTrue()
{
    echo "in retTrue\n";
    return true;
}


# won't print anything
if (0 && retTrue()) {
    echo "true";
}

echo "hello";
```

OUTPUT:

PHP Sandbox, test PH | PHP Comments

https://sandbox.onlinephpfunctions.com

Test your PHP code with this code tester

You can test your PHP code here on many php versions.

Your script:

```php
<?php
# collin gros
# 09-09-2020
# this program proves php short circuits if statements by showing the
# function retTrue() is not called when a conditional of 0 && retTrue().

# returns true, and prints that we are inside
function retTrue()
{
    echo "in retTrue\n";
    return true;
}


# won't print anything because php short circuits
if (0 && retTrue()) {
    echo "true";
}

echo "hello";
```

Run on PHP version: 7.4.7

Output: Textbox

Execute code | Save or share your code

Result:

```
hello
```

The fact that "in retTrue" was not printed indicates that it was never evaluated in the if statement. Therefore, PHP implements short circuiting.

PERL

CODE:

```
collin@collin-fx:~/Documents/school/cs-471/prog-2/perl$ ls
test.pl
collin@collin-fx:~/Documents/school/cs-471/prog-2/perl$ cat test.pl
#!/usr/bin/perl
# collin gros
# 09-09-2020
#
# this program shows that perl short circuits if statements by proving
# perl does not move through a function if the first part of the statement
# is evaluated as false


# returns true and prints that we've gone through it
sub retTrue
{
        print "in retTrue\n";
        return 1;
}


# short circuits, retTrue is never called because the frist part of
# statement is eval as false, retTrue doesn't matter
if (0 and retTrue) {
        print "if\n";
}
```

OUTPUT:

```
collin@collin-fx:~/Documents/school/cs-471/prog-2/perl$ ls
test.pl
collin@collin-fx:~/Documents/school/cs-471/prog-2/perl$ ./test.pl
collin@collin-fx:~/Documents/school/cs-471/prog-2/perl$
```

The fact that "in retTrue" was not printed indicates that the second part of the if statement was never evaluated; therefore, perl implements short-circuiting.