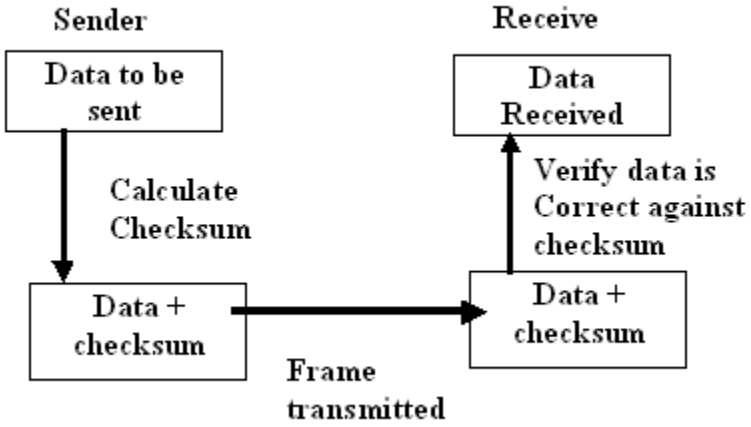
Cyclic Redundancy Check (CRC)

by hEYWIRE

For anyone that cracks programs I'm sure you have come across this. In some programs they check the Cyclic Redundancy Check (CRC) and if not correct give you an error or tell you the program is infected with a virus. When patching programs with CRC checking you have to patch the routine of the CRC check. But have you ever wondered how the CRC works?

Well let me give you a small bit of background information about CRC. CRC is an international standard approach to error detection. It protects the data with a checksum or **cyclic redundancy check**. CRC was first developed by the CCITT (Comité Consultatif International Telegraphique et Telephonique) now called ITU – T (International Telecommunications Union) or something like that!

The Cyclic Redundancy Check is the most powerful of the redundancy checking techniques (Yes theirs more but this is the most common one used), the CRC is based on binary division. In CRC a sequence of redundant bits, called the CRC or the CRC remainder is appended to the end of a data stream. The resulting data becomes exactly divisible by a second, predetermined binary number. At its destination, the incoming data is divided by the same number. The diagram below will show you the Sequence of Events that takes place when using CRC.



As you can see from the diagram it's fairly easy to follow but to get/verify the CRC or checksum is a different matter. You will notice that FRAME TRANSMITTED is used. This is a Number of 0's added to data unit (total equals highest number of polynomial generator) The Polynomial Generator is used to calculate a binary equivalent. – Present powers are identified as 1 and a absent powers as 0.The standard polynomials used by protocols are CRC-12, CRC-16, CRC-32. E.g. CRC- 16 $x^{16} + x^{15} + x^2 + 1$ A CRC generator (divisor) is usually represented not as a string of 1s and 0s but as an algebraic polynomial. A CRC Generator uses modulo-2 division.

I think its time to but all this theory to work. The following is the procedure for sending the data below with the given polynomial generator.

Data to be sent: 11011

Polynomial generator: $x^4 + x^2 + 1$

1. Take the polynomial generator and work out its corresponding binary representation by identifying the absent powers

Polynomial = $x^4 + x^2 + 1$

= $x^4 + 0^3 + x^2 + 0^1 + 1$

Divisor = 10101 ;this is the digital representation of the polynomial generator.

2. A Frame Check Sequence (FCS) which equates to the highest power of the polynomial generator – (4 in the example above and represented as four zeros - 0000). The FCS is added to the data resulting in

3. Ok so we now have the data to be sent which is 11011 and the polynomial generator which is $x^4 + 0^3 + x^2 + 0^1 + 1$ or in its binary form 10101 and the Frame Check Sequence (FCS) which is 0000. Now all we need to do is get the CRC. The CRC is calculated by dividing the binary representation of the polynomial into the DATA + FCS using modulo-2 arithmetic. Below is a table, which will help you out with modulo-2 arithmetic.

Modulo-2 Table	
1 + 0 =	1
0 + 1 =	1
1 + 1 =	0
0 + 0 =	0

```
 PG   Data  FCS
10101 11011 0000
      10101
       11100
       10101
        10010
        10101
          11100
          10101
            1001
```

So we calculated the CRC by dividing the binary representation of the polynomial into the data and the FCS. So we end up with 1001 which is the CRC. So it's easy enough to find the CRC. Now we have the CRC for the data which has been sent all theirs left to do is check the CRC and find out if the data has been received correctly. We do this exactly like the CRC Generator. If the remainder is all 0s, the CRC is dropped and the data is not corrupted, if the remainder is not = 0s, the data is corrupted.

```
 PG   Data  FCS
10101 11011 1001
      10101
       11101
       10101
        10000
        10101
           10101
           10101
             0000
```

Well as you can see from the above calculation the data is not corrupt because the remainder is all 0's, and you now know how the CRC for programs is calculated. If you wanted you could try make a program that will get the CRC of any program or insert a CRC check for any programs you code. Below are a few excrises you can try do.

1: Data: 11011; Polynomial x3+x2+1. What is the checksum?

2: Data: 11001; Polynomial x3+x2+1. What is the checksum?

3: Data: 110101; Polynomial x4+x2+1. What is the checksum?

That's it for this Tutorial, hope you learned something, if there is any spelling mistakes or grammar errors then forgive me, Im only human. Peace hEYWIRE