SSD

Calculate Lifetime

Most vendors beyond the low end consumer devices will have a TBW parameter on their spec sheet. This is the estimated number of Terabytes Written before failure of the SSD is likely. All vendors appear to grossly underestimate this value, but this is normally the extent of their warranties.

Using smartctl for various Unix systems, you can read the values of most SSD's and calculate amount of data written to date.

```
apt install smartmontools # Devuan Linux
pkg install smartmontools # FreeBSD
# record number of Logical Bytes that makes up a sector
smartctl /dev/sda -a |grep "Sector Size"
# record Raw Value (rightmost column) as Number of Writes
smartctl -a /dev/sda | grep '241 Host_Writes'
```

Now, multiply the above two values (rightmost column on both). That is the number of bytes written to the SSD since it was manufactured.

Nothing I'm aware of has less than a 10 TBW, and most high end consumer grade SSD's are rated at many times that. For example, a Samsung 850 Pro SATA 250G SSD is rated at 70 TBW, but tests show it can get up to 150 TBW.

Note that, using the same technology, a larger capacity will have a linearly greater TBW value. The TBW value is based on between 3000 and 100,000 writes per cell depending on the technology. If you have a larger capacity drive, it has more cells, so it would have more TBW.

You can find the make/model of the drive with the following two commands:

```
smartctl -a /dev/sda | grep '^Model Family:'
smartctl -a /dev/sda | grep '^Device Model:'
```

and go to the vendors web site to see what they estimate is the lifetime.

NOTE: some vendors do not include ID 241 in their output, so all you can do is guess. One good thing to try is simply take the output of smartctl -a and pipe it to less, then read through it (not that big).

Script

The following script is very rough, but kinda works. No parameters, but you have to comment/uncomment lines at the beginning depending on whether you want to manually set the drives, or have them discovered by Linux or BSD systems. Use at your own risk. I'll probably clean it up in the future.

getSmart.pl

```
#! /usr/bin/env perl

use warnings;
use strict;

my %allDrives;

# create a hash with the keys having values ada0, da0, etc
# comment/uncomment one of the three lines below

# manually enter the drives to look at
# %allDrives = map { $_ => 0 } qw/ada0 da0 da1 da2 da3 da4 da5 da6 da7
da8 da9/;
# automagically find the drives on a BSD system
%allDrives = map { &trim($_) => 0 } `geom disk list | grep 'Geom name:'
| cut -d':' -f 2`;
# automagically find the drives on a Linux system
#%allDrives = map { &trim($_) => 0 } `lsblk | grep disk | cut -d' ' -
f1`;

sub trim {
   my $value = shift;
   $value =~ s/^\s+|\s+$//g;
   return $value;
}

sub getTotalWrites {
   my $drive = shift;
   my @report = `smartctl -a /dev/$drive`;
   return -2 unless grep{ /(Solid State Device)|(SSD)/ } @report;
   my @temp = grep{ /^Sector Size:/ } @report;
   my $sectors = shift @temp;
#   print "The Value is [$sectors]\n"; die;
   if ( $sectors =~ m/^Sector Size\:\s+(\d+)\s+bytes/ ) {
      $sectors = $1;
      # print "Sectors is $sectors\n"; die;
      @temp =  grep{ /^241/ } @report;
      my $lbas = $temp[0];
      if ( $lbas =~ m/(\d+)\s*$/ ) {
         $lbas = $1;
         return $lbas * $sectors;
      } else {
         return -3;
      }
   } else {
      return -4;
   }
   return -1; # we could not find something
}
```

```
# output the drives and information as tab delimited,
foreach my $thisDrive ( sort keys %allDrives ) {
   #print "$thisDrive\n";
   $allDrives{$thisDrive} = getTotalWrites( $thisDrive );
   print "$thisDrive\t" . $allDrives{$thisDrive} . "\n" if
$allDrives{$thisDrive} > 0;
}
```

Links

- https://www.ontrack.com/blog/2018/02/07/how-long-do-ssds-really-last/ (this is a service company, and towards the bottom is some advertisement for their service, but the article appears to be very fair).

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/other/hardware/ssd

Last update: **2020/01/26 03:06**