

Penetrate – NVMe SSD Test Software

Cell Computers Ltd.

Contents

- Overview
- Applications
- Features
- Functions
- Language Features
- Licenses

Contents

- Overview
- Applications
- Features
- Functions
- Language Features
- Licenses

Overview

- Test NVMe SSDs/semi-finished SSDs
- Assist diagnosing issues
- Tailor testcases by scripting
- The package includes
 - Script language interpreter – Peanut
 - Run on a Linux terminal
 - Peanut extension modules
 - NVMe
 - C library
 - System
 - Peanut scripts

Contents

- Overview
- **Applications**
- Features
- Functions
- Language Features
- Licenses

Applications

- NVMe protocol verification
- Data integrity regression
- IO workload generation
- Performance measurement
- Vendor command/protocol modeling
- Developing IC/module verification

Contents

- Overview
- Applications
- **Features**
- Functions
- Language Features
- Licenses

Features

- Lightweight
 - Proprietary user-space NVMe driver
- High efficiency
- Referring C/C++ syntax and supporting conditional inclusions
- Check without running
- Multitasking
- NVMe 2.0 support
- Enterprise/Data Center specifics
- Manipulating developing devices
- Unlimited log files

Contents

- Overview
- Applications
- Features
- **Functions**
- Language Features
- Licenses

Command Level Operations

- Script APIs to issue commands
- NVMe 2.0 and backward compatible APIs
 - Admin Command Set
 - I/O Command Set
 - NVM Command Set
 - ZNS Command Set
 - KV Command Set

Register Accesses

- Read configuration-space registers
- Read controller registers

Multitasking

- Create multi tasks
- Access multiple devices in parallel
- Access multiple controllers (functions) in parallel
- Sending commands to a submission queue is thread-safe

Query Information

- Query devices information
 - Device number
 - Function number per device
- Query queues information
 - Max and current IOCQ/IOSQ number
 - Max IOQ size
 - IOCQ/IOSQ id
- Query command issued/completed number
 - Controller
 - Submission queue
 - OP code
- Query commands information
 - Complete/Timeout
 - Status
 - CQE CDW1
 - Execution time
 - Full CQE CDWs

Initialize, Rescan, and Reinitialize

- Initialization API and queue creations with parameters:
 - Sizes of ACQ/ASQ
 - Sizes of IOCQ/IOSQ
 - Number of IOSQs
 - Number of IOSQs per IOCQ
- Rescan the device after a “PHY-reestablishment” reset
- Same API for a reinitialization after a reset

DMA Data Buffer

- Allocate/Free DMA data buffer dynamically
- Read/Write DMA data buffer
- PRP and SGL types
- Dump data to screen/file

Namespace and Metadata

- Support multi-namespaces
 - Independent LBA size
 - Independent metadata size
 - Independent metadata transport
 - Extended LBA
 - Separate metadata buffer

SR-IOV Virtual Function

- Enable and disable virtual functions
- Initialization
- Virtualization Management command
- Namespace Management command
- Namespace Attachment command

Resets

- Function Level Reset
- Controller Reset
- NVM Subsystem Reset
- PCIe Hot Reset
- Power-on Reset

Trace

- Command Record
- Event Record
- Partial or Full (all the latest)
- Dump Trace to screen/file

No.	Start Timestamp	Q	CID	Command C		Execution Time (us)	CDW0	CDW1	CDW2	CDW3	CDW4	CDW5	CDW6	CDW7	CDW8	CDW9	CDW10	CDW11	CDW12	CDW13	CDW14	CDW15	Status
				Event	P																		
0001	24/03/13 15:01:36.084871	0	00B7	Virt Mgmt	1	603	00B7001C	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00030009	00000000	00000000	00000000	00000000	00000000	0000h
0002	24/03/13 15:01:35.275431	0	00B6	Virt Mgmt	1	151	00B6001C	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00030108	00000002	00000000	00000000	00000000	00000000	0000h
0003	24/03/13 15:01:35.275279	0	00B5	Virt Mgmt	1	150	00B5001C	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00030008	00000002	00000000	00000000	00000000	00000000	0000h
0004	24/03/13 15:01:35.273324	0	00B4	Ns Atchmt	1	1952	00B40015	00000004	00000000	00000000	00000000	00000000	A262A000	00000001	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0000h
0005	24/03/13 15:01:35.248364	0	00B3	Ns Mgmt	1	24927	00B3000D	00000000	00000000	00000000	00000000	00000000	A262A000	00000001	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0000h

No.	Start Timestamp	Q	CID	Command C		Execution Time (us)	CDW0	CDW1	CDW2	CDW3	CDW4	CDW5	CDW6	CDW7	CDW8	CDW9	CDW10	CDW11	CDW12	CDW13	CDW14	CDW15	Status
				Event	P																		
0001	24/03/13 15:01:36.095239	0	0004	Identify	1	184	00040006	00000000	00000000	00000000	00000000	00000000	A2634000	00000001	00000000	00000000	00000001	00000000	00000000	00000000	00000000	00000000	0000h
0002	24/03/13 15:01:36.094182	0	0003	Create IO SQ	1	177	00030001	00000000	00000000	00000000	00000000	00000000	A2630000	00000001	00000000	00000000	00FF0001	00010001	00000000	00000000	00000000	00000000	0000h
0003	24/03/13 15:01:36.092279	0	0002	Create IO CQ	1	104	00020005	00000000	00000000	00000000	00000000	00000000	A262C000	00000001	00000000	00000000	03FF0001	00000001	00000000	00000000	00000000	00000000	0000h
0004	24/03/13 15:01:36.091213	0	0001	Set Ftrs	1	259	00010009	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000007	00000000	00000000	00000000	00000000	00000000	0000h
0005	24/03/13 15:01:36.090158	0	0000	Get Ftrs	1	553	0000000A	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000107	00000000	00000000	00000000	00000000	00000000	0000h
0006	24/03/13 15:01:35.275589			FL Rst		308530																	

Command Execution Time

- Individual command's execution time
- Execution time statistics
- QoS statistics

Callback for Illegal CQEs

- Registered through the initialization API
- Callback interface: `CallbackName(argc, argv[])`

Contents

- Overview
- Applications
- Features
- Functions
- **Language Features**
- Licenses

Language Features

- Referring C and C++ syntax
- Extensible
- Multitasking programming
- Namespace resolution
- Untyped elementary variables
- Compound data
 - Array
 - Structure
 - Object programming
- Call by value and call by reference
- Function types: Built-in function, Extension function, User function
- Conditional inclusion
- Check without running

```
#load <pxm/pxm_clib.pxm>
#load <pxm/pxm_nvme.pxm>

main(argc, argv[])
{
    var i;

    c::printf("Hello");
    for (i = 1; i < argc; ++i)
        c::printf(" %s", argv[i]);
    print("!\n");
}
```

Language Features

```
2 struct Point {
3     var x, y;
4     ...
5     vect2Point(struct Point P, struct Point &V)
6     {
7         V.x = P.x - x;
8         V.y = P.y - y;
9     }
10 } ;
11
12 struct Line {
13     struct Point P;
14     struct Point Q;
15     ...
16     len()
17     {
18         struct Point V;
19         var LL, Lh, Ll, Lm, i;
20
21         P.vect2Point(Q, V);
22         LL = V.x * V.x + V.y * V.y;
```

```
39 main()
40 {
41     struct Point Q, R;
42     struct Point V;
43     struct Line L;
44
45     Q.x = -2;
46     Q.y = -3;
47     R.init(5, 6);
48     ...
49     L.init(Q, R);
50     print("length of L, from Q to R, is about ", L.len());
51 }
```

Contents

- Overview
- Applications
- Features
- Functions
- Language Features
- **Licenses**

Licenses

- Source License

Thanks For Your Time.