



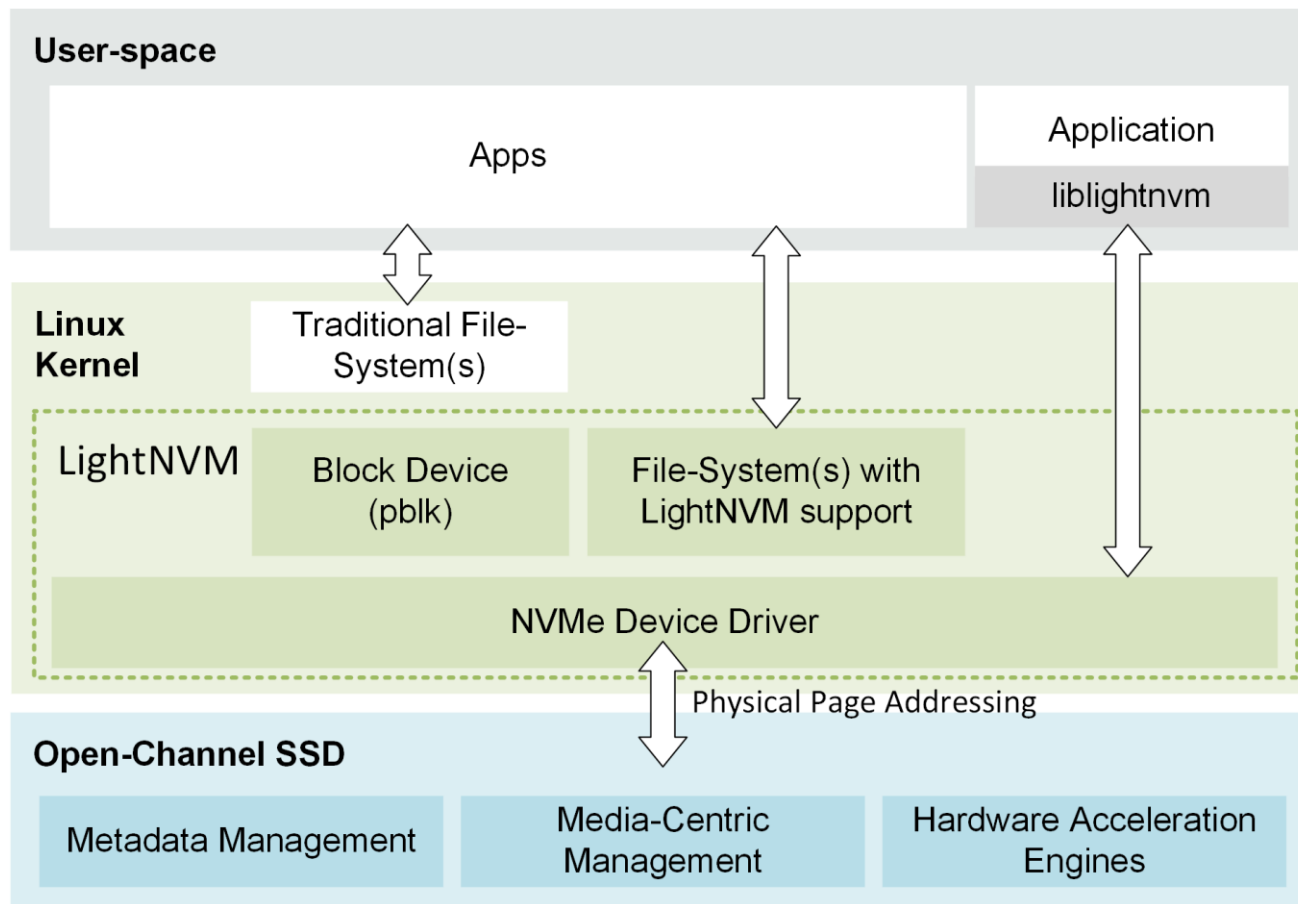
# D-Tux 연구주제 - LightNVM -

Choi GunHee

choi\_gunhee@dankook.ac.kr

# LightNVM

- LightNVM

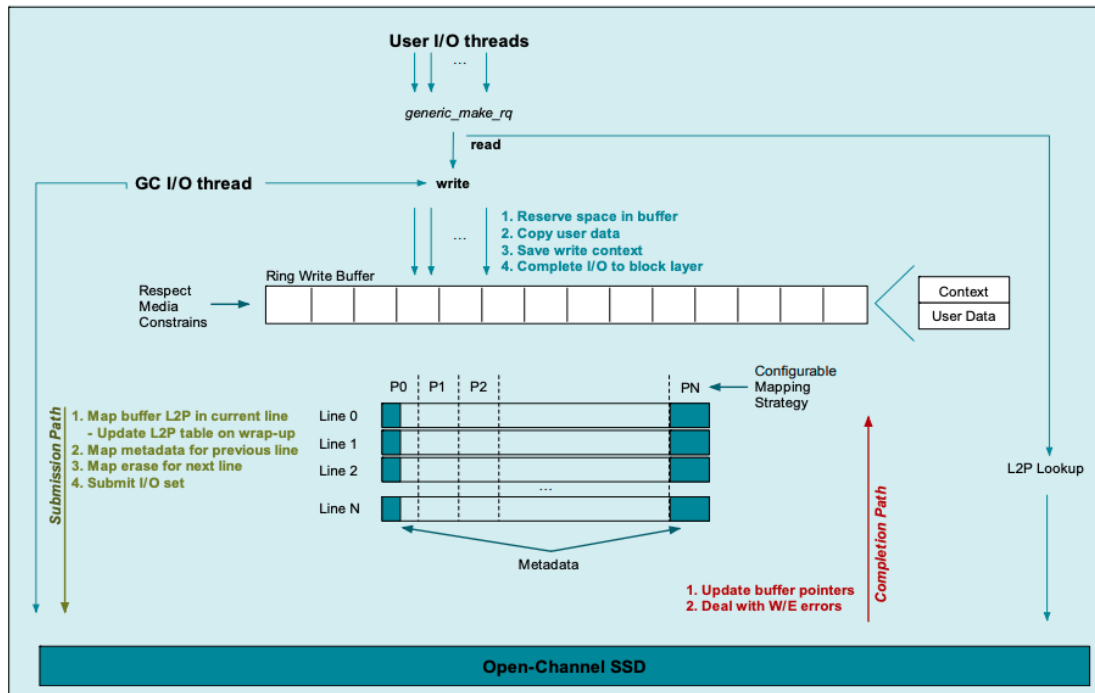
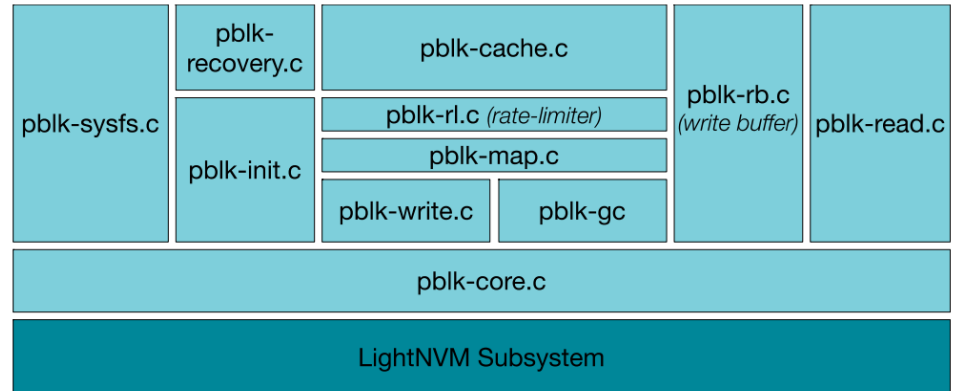


# LightNVM

## ● LightNVM

### ○ 소스코드 위치

- /drivers/lightnvm
- /drivers/nvme/host/lightnvm
- /include/linux/lightnvm.h



# LightNVM

## ● FEMU 설치

- 링크 : <https://github.com/ucare-uchicago/femu>
- git clone <https://github.com/ucare-uchicago/femu.git>

```
choigunhee@choigunhee-univ-server93:~$ git clone https://github.com/ucare-uchicago/femu.git
```

```
Cloning into 'femu'...
```

```
remote: Enumerating objects: 380443, done.
```

```
remote: Total 380443 (delta 0), reused 0 (delta 0), pack-reused 380443
```

```
Receiving objects: 100% (380443/380443), 259.48 MiB | 2.72 MiB/s, done.
```

```
Resolving deltas: 100% (303456/303456), done.
```

```
Checking connectivity... done.
```

```
choigunhee@choigunhee-univ-server93:~$ cd femu/
```

```
choigunhee@choigunhee-univ-server93:~/femu$ ls
```

accel.c	COPYING	gdbstub.c	Makefile	qemu-bridge-helper.c	qom	trace-events
arch_init.c	COPYING.LIB	<b>gdb-xml</b>	Makefile.objs	qemu-doc.texi	qtest.c	translate-all.c
atomic_template.h	cpu-exec.c	<b>gen_cscope.sh</b>	Makefile.target	qemu-ga.texi	README.md	translate-all.h
<b>audio</b>	cpu-exec-common.c	HACKING	memory.c	qemu-img.c	<b>replay</b>	translate-common.c
<b>backends</b>	cpus.c	hax-stub.c	memory_ldst.inc.c	qemu-img-cmds.hx	replication.c	<b>ui</b>
balloon.c	cpus-common.c	hmp.c	memory_mapping.c	qemu-img.texi	replication.h	user-exec.c
<b>block</b>	cputlb.c	hmp-commands.hx	<b>migration</b>	qemu-io.c	<b>roms</b>	user-exec-stub.c
block.c	<b>crypto</b>	hmp-commands-info.hx	module-common.c	qemu-io-cmds.c	rules.mak	<b>util</b>
blockdev.c	<b>default-configs</b>	hmp.h	monitor.c	qemu-nbd.c	<b>scripts</b>	VERSION
blockdev-nbd.c	device-hotplug.c	<b>hw</b>	<b>nbd</b>	qemu-nbd.texi	<b>slirp</b>	version.rc
blockjob.c	device_tree.c	<b>include</b>	<b>net</b>	qemu.nsi	softmmu_template.h	vl.c
bootdevice.c	<b>disas</b>	<b>io</b>	numa.c	qemu-options.h	spice-qemu-char.c	xen-common.c
<b>bsd-user</b>	disas.c	ioport.c	os-posix.c	qemu-options.hx	<b>stubs</b>	xen-common-stub.c
bt-host.c	dma-helpers.c	iothread.c	os-win32.c	qemu-options-wrapper.h	<b>target</b>	xen-hvm.c
bt-vhci.c	<b>docs</b>	kvm-all.c	page_cache.c	qemu-option-trace.texi	<b>tcg</b>	xen-hvm-stub.c
Changelog	dump.c	kvm-stub.c	<b>pc-bios</b>	qemu.sasl	tcg-runtime.c	xen-mapcache.c
<b>chardev</b>	exec.c	<b>libdecnumber</b>	<b>po</b>	qemu-seccomp.c	tcic.c	
CODING_STYLE	<b>femu-scripts</b>	LICENSE	<b>qapi</b>	qemu-tech.texi	<b>tests</b>	
<b>configure</b>	<b>fire</b>	<b>linux-headers</b>	qapi-schema.json	<b>qga</b>	thunk.c	
_config.yml	<b>fpu</b>	<b>linux-user</b>	qdev-monitor.c	qmp.c	tpm.c	
<b>contrib</b>	<b>fsdev</b>	MAINTAINERS	qdict-test-data.txt	<b>qobject</b>	<b>trace</b>	

```
choigunhee@choigunhee-univ-server93:~/femu$ █
```

# LightNVM

## ● FEMU 설치

- git checkout ba56426057f57f7eab839e860c7672d5289e25c5
- cd build-femu

```
choigunhee@choigunhee-univ-server93:~/femu$ git checkout ba56426057f57f7eab839e860c7672d5289e25c5
Note: checking out 'ba56426057f57f7eab839e860c7672d5289e25c5'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

```
HEAD is now at ba56426... Add irqfd support
```

```
choigunhee@choigunhee-univ-server93:~/femu$ cd build-femu/
```

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ git checkout ba56426057f57f7eab839e860c7672d5289e25c5
```

```
HEAD is now at ba56426... Add irqfd support
```

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ █
```

# LightNVM

## ● FEMU 설치

- `cp ../femu-scripts/femu-copy-script.sh .`

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ cp ../femu-scripts/femu-copy-scripts.sh .
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ ls
femu-copy-scripts.sh
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ █
```

- `./femu-copy-scripts.sh`

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ ./femu-copy-scripts.sh
```

```
==> Copying following FEMU script to current directory:
```

```
--> pkgdep.sh
--> femu-compile.sh
--> run-whitebox.sh
--> run-blackbox.sh
--> run-nossd.sh
--> pin.sh
--> ftk
--> vssd1.conf
```

```
Done!
```

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ █
```

# LightNVM

- FEMU 설치

- sudo ./pkgdep.sh

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ ls
femu-compile.sh femu-copy-scripts.sh ftk pin.sh pkgdep.sh run-blackbox.sh run-nossd.sh run-whitebox.sh vssd1.conf
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ sudo ./pkgdep.sh
[sudo] password for choigunhee: █
```

- ./femu-compile.sh

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ clear
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ ./femu-compile.sh
Install prefix      /usr/local
BIOS directory     /usr/local/share/qemu
binary directory   /usr/local/bin
library directory  /usr/local/lib
module directory   /usr/local/lib/qemu
libexec directory  /usr/local/libexec
include directory  /usr/local/include
config directory   /usr/local/etc
local state directory /usr/local/var
Manual directory   /usr/local/share/man
ELF interp prefix  /usr/gnemul/qemu-%M
Source path        /home/choigunhee/femu
```

# LightNVM

- FEMU 설치

- cd qemu-system-x86\_64

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu/x86_64-softmmu$ ls
```

arch_init.d	config-target.h	cpus.o	exec.o	hw	memory_mapping.o	qtest.d	translate-all.o
arch_init.o	config-target.h-timestamp	cputlb.d	fpu	ioport.d	memory.o	qtest.o	translate-common.d
balloon.d	config-target.mak	cputlb.o	gdbstub.d	ioport.o	migration	target	translate-common.o
balloon.o	cpu-exec-common.d	disas.d	gdbstub.o	kvm-all.d	monitor.d	tcg	xen-common-stub.d
bootdevice.d	cpu-exec-common.o	disas.o	hax-stub.d	kvm-all.o	monitor.o	tcg-runtime.d	xen-common-stub.o
bootdevice.o	cpu-exec.d	dump.d	hax-stub.o	Makefile	numa.d	tcg-runtime.o	xen-hvm-stub.d
config-devices.mak	cpu-exec.o	dump.o	hmp-commands.h	memory.d	numa.o	trace	xen-hvm-stub.o
config-devices.mak.old	cpus.d	exec.d	hmp-commands-info.h	memory_mapping.d	qemu-system-x86_64	translate-all.d	

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu/x86_64-softmmu$ █
```



# LightNVM

- FEMU 설치

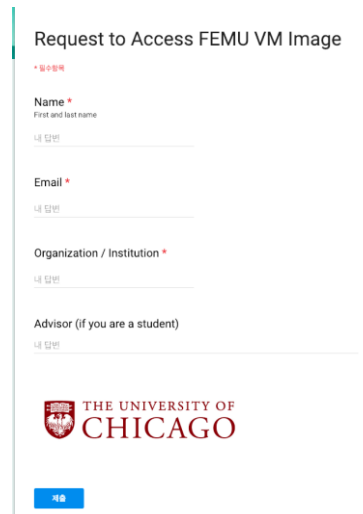
- Image 준비

- 커스텀 이미지 준비 링크 :

[https://help.ubuntu.com/community/Installation/QemuEmulator#Installation\\_of\\_an\\_operating\\_system\\_from\\_ISO\\_to\\_the\\_QEMU\\_environment](https://help.ubuntu.com/community/Installation/QemuEmulator#Installation_of_an_operating_system_from_ISO_to_the_QEMU_environment)

- 제공 이미지 다운로드 링크 :

[https://docs.google.com/forms/d/e/1FAIpQLSdCyNTU7n-hwW1ODJ3i\\_q1vmS6eTT-V3c4vCL8ouYocNLhxA/viewform](https://docs.google.com/forms/d/e/1FAIpQLSdCyNTU7n-hwW1ODJ3i_q1vmS6eTT-V3c4vCL8ouYocNLhxA/viewform)



The screenshot shows a Google Form titled "Request to Access FEMU VM Image". It includes a red asterisk indicating it is a required field. The form has four text input fields, each with a "내 답변" (My Answer) placeholder. The fields are labeled: "Name \* (First and last name)", "Email \*", "Organization / Institution \*", and "Advisor (if you are a student)". At the bottom of the form is the University of Chicago logo and a blue "48" button.

# LightNVM

- FEMU 설치
  - Image 준비
  - wget 이용

```
choigunhee@choigunhee-univ-server93:~$ wget --2019-01-07 16:19:29-- http://people.cs.uchicago.edu/~drexler/femu-vm.tar.xz
Resolving people.cs.uchicago.edu (people.cs.uchicago.edu)... 128.135.164.80
Connecting to people.cs.uchicago.edu (people.cs.uchicago.edu)|128.135.164.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3081634808 (2.9G) [application/x-xz]
Saving to: 'femu-vm.tar.xz'

femu-vm.tar.xz          100%[=====>]  2.87G  1.28MB/s  in 51m 3s

2019-01-07 17:10:33 (983 KB/s) - 'femu-vm.tar.xz' saved [3081634808/3081634808]

choigunhee@choigunhee-univ-server93:~$ █
```

# LightNVM

## ● FEMU 설치

- Image 준비
- mkdir images
- cp femu-vm.tar.xz images/

```
choigunhee@choigunhee-univ-server93:~$ ls
Desktop    Downloads    femu          Music        Public        Videos
Documents  examples.desktop  femu-vm.tar.xz  Pictures    Templates
choigunhee@choigunhee-univ-server93:~$ mkdir images
choigunhee@choigunhee-univ-server93:~$ cp femu-vm.tar.xz images/
```

- xz -d femu-vm.tar.xz
- tar -xf femu-vm.tar

```
choigunhee@choigunhee-univ-server93:~/images$ ls
femu-vm.tar.xz
choigunhee@choigunhee-univ-server93:~/images$ xz -d femu-vm.tar.xz
choigunhee@choigunhee-univ-server93:~/images$ tar -xf femu-vm.tar
choigunhee@choigunhee-univ-server93:~/images$ ls
femu-vm.tar  u14s.md5sum  u14s.qcow2
choigunhee@choigunhee-univ-server93:~/images$ █
```

# LightNVM

## ● FEMU 실행

- femu/build-femu/run-whitebox.sh
- NVMEIMGSZ : SSD용량
- Inum\_lun : LNU 개수
- Inum\_ch : LNU 당 channel 개수

```
# image directory
IMGDIR=$HOME/images
# virtual machine disk image
OSIMGF=$IMGDIR/u14s.qcow2
# virtual NVMe disk image
NVMEIMGF=$IMGDIR/vssd1.raw
# virtual NVMe disk size: 1GB
NVMEIMGSZ=16G
```

```
# every time we create a new SSD image file
sudo rm -rf $IMGDIR/vssd1.raw
```

```
sudo x86_64-softmmu/qemu-system-x86_64 \
  -name "FEMU-whitebox-SSD" \
  -enable-kvm \
  -cpu host \
  -smp 4 \
  -m 4G \
  -device virtio-scsi-pci,id=scsi0 \
  -device scsi-hd,drive=hd0 \
  -drive file=$OSIMGF,if=none,aio=native,cache=none,format=qcow2,id=hd0 \
  -drive file=$NVMEIMGF,if=none,aio=threads,format=raw,id=id0 \
  -device nvme,drive=id0,serial=serial0,id=nvme0,namespaces=1,lver=1,lmetasize=16,ll2pmode=0,nlba=5,lba_index=3,mdts=10,lnum_ch=2,lnum_lun=8,lnum_pln=2,lsec_s
ize=4096,lsecs_per_pg=4,lpgs_per_blk=512,ldebug=0,femu_mode=0 \
  -net user,hostfwd=tcp::8080-:22 \
  -net nic,model=virtio \
  -nographic \
  -qmp unix:./qmp-sock,server,nowait
```

# LightNVM

- FEMU 실행

- sudo ./run-whitebox.sh

```
choigunhee@choigunhee-univ-server93:~/femu/build-femu$ sudo ./run-whitebox.sh  
[sudo] password for choigunhee: █
```

```
GNU GRUB  version 2.02~beta2-36ubuntu3.17
```

```
+-----+  
|*Ubuntu, with Linux 4.16.0|  
| Ubuntu, with Linux 4.16.0 (recovery mode)|  
| Ubuntu, with Linux 4.14.0|  
| Ubuntu, with Linux 4.14.0 (recovery mode)|  
| Ubuntu, with Linux 4.4.0-121-generic|  
| Ubuntu, with Linux 4.4.0-121-generic (recovery mode)|  
|  
|  
|  
|  
|  
+-----+
```

# LightNVM

- FEMU 실행
  - ID : femu
  - Password : femu

```
--  
→ ~ ssh -p 8080 femu@220.149.250.93  
femu@220.149.250.93's password:  
--
```

Ubuntu 16.04.4 LTS fvm ttyS0

fvm login: femu

Password:

Last login: Mon Apr 23 15:40:55 CDT 2018 from 10.0.2.2 on pts/0

Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.16.0 x86\_64)

\* Documentation: <https://help.ubuntu.com>

\* Management: <https://landscape.canonical.com>

\* Support: <https://ubuntu.com/advantage>

169 packages can be updated.

100 updates are security updates.

femu@fvm:~\$ █

femu@220.149.250.93's password:

Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.16.0 x86\_64)

\* Documentation: <https://help.ubuntu.com>

\* Management: <https://landscape.canonical.com>

\* Support: <https://ubuntu.com/advantage>

169 packages can be updated.

100 updates are security updates.

New release '18.04.1 LTS' available.

Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Jan 7 02:50:26 2019

femu@fvm:~\$ █

# LightNVM

## ● FEMU 실행

- nvme cli 설치
- 링크 : <https://github.com/linux-nvme/nvme-cli>
- cd git/nvme-cli

```
femu@fvm:~$ ls
git
femu@fvm:~$ cd git/
femu@fvm:~/git$ ls
fio  linux  nvme-cli
femu@fvm:~/git$ cd nvme-cli/
femu@fvm:~/git/nvme-cli$ ls
CONTRIBUTING.md  argconfig.c  define_cmd.h  intel-nvme.h  memblaze-nvme.c  nvme-ioctl.h  nvme-print.h  parser.h  suffix.h  wdc-utils.c
Documentation     argconfig.h  fabrics.c     json.c        memblaze-nvme.h  nvme-lightnvm.c  nvme.c        plugin.c  tests      wdc-utils.h
LICENSE           cmd.h        fabrics.h     json.h        netapp-nvme.c    nvme-lightnvm.h  nvme.control.in  plugin.h  toshiba-nvme.c
Makefile          cmd_handler.h  huawei-nvme.c  linux        netapp-nvme.h    nvme-models.c    nvme.h         regress  toshiba-nvme.h
NVME-VERSION-GEN  common.h      huawei-nvme.h  lnvm-nvme.c   nvme-builtin.h   nvme-models.h    nvme.spec.in    scripts  wdc-nvme.c
README.md         completions   intel-nvme.c  lnvm-nvme.h   nvme-ioctl.c     nvme-print.c     parser.c        suffix.c  wdc-nvme.h
femu@fvm:~/git/nvme-cli$
```

# LightNVM

- FEMU 실행
  - nvme cli 설치
  - make
  - sudo make install

```
femu@fvm:~/git/nvme-cli$ make
NVME_VERSION = 1.5.151.g139c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c argconfig.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c suffix.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c parser.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c nvme-print.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c nvme-ioctl.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c nvme-lightnvm.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c fabrics.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c json.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c plugin.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c intel-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c lnvm-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c memblaze-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c wdc-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c wdc-utils.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c nvme-models.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c huawei-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c netapp-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c toshiba-nvme.c
cc -D_GNU_SOURCE -D__CHECK_ENDIAN__ -O2 -g -Wall -Werror -std=gnu99 -DNVME_VERSION='"1.5.151.g139c"' -c nvme.c -o nvme argcor
me-ioctl.o nvme-lightnvm.o fabrics.o json.o plugin.o intel-nvme.o lnvm-nvme.o memblaze-nvme.o wdc-nvme.o wdc-utils.o nvme-
oshiba-nvme.o
femu@fvm:~/git/nvme-cli$ sudo make install
[sudo] password for femu:
install -d /usr/local/sbin
install -m 755 nvme /usr/local/sbin
make -C Documentation install-no-build
make[1]: Entering directory '/home/femu/git/nvme-cli/Documentation'
install -d -m 755 /usr/local/share/man/man1
install -m 644 nvme-effects-log.1 nvme-set-feature.1 nvme-resv-release.1 nvme-fw-download.1 nvme-get-ns-id.1 nvme-intel-lc
etry-log.1 nvme-write.1 nvme-flush.1 nvme-ns-descs.1 nvme-intel-id-ctrl.1 nvme-dsm.1 nvme-write-zeroes.1 nvme-wdc-purge-mc
nvme-admin-passthru.1 nvme-get-feature.1 nvme-dir-send.1 nvme-delete-ns.1 nvme-wdc-id-ctrl.1 nvme-lnvm-info.1 nvme-write-ve
.1 nvme-wdc-purge.1 nvme-subsystem-reset.1 nvme-lnvm-diag-bbtbl.1 nvme-resv-register.1 nvme-connect-all.1 nvme-intel-int
ire.1 nvme-huawei-list.1 nvme-io-passthru.1 nvme-dir-receive.1 nvme-fw-log.1 nvme-huawei-id-ctrl.1 nvme-id-ns.1 nvme-conne
ity-recv.1 nvme-endurance-log.1 nvme-lnvm-factory.1 nvme-compare.1 nvme-list.1 nvme-intel-smart-log-add.1 nvme-help.1 nvme
1 nvme-smart-log.1 nvme-wdc-get-crash-dump.1 nvme-security-send.1 nvme-intel-market-name.1 nvme-wdc-drive-essentials.1 nvm
m-diag-set-bbtbl.1 nvme-intel-temp-stats.1 nvme-wdc-smart-add-log.1 nvme-sanitize.1 nvme-wdc-drive-log.1 nvme-lnvm-init.1
e-detach-ns.1 nvme-wdc-clear-pcie-corr.1 nvme-id-ctrl.1 nvme-set-property.1 nvme-toshiba-vs-internal-log.1 nvme-ns-rescan.
e-disconnect.1 nvme-list-ctrl.1 nvme-toshiba-vs-smart-add-log.1 nvme-format.1 nvme-lnvm-id-ns.1 nvme-read.1 nvme-discover.
nvme-resv-report.1 nvme-sanitize-log.1 nvme.1 /usr/local/share/man/man1
make[1]: Leaving directory '/home/femu/git/nvme-cli/Documentation'
install -d /usr/local/share/bash_completion.d
install -m 644 -T ./completions/bash-nvme-completion.sh /usr/local/share/bash_completion.d/nvme
femu@fvm:~/git/nvme-cli$
```



# LightNVM

- FEMU 실행
  - LightNVM 사용
  - sudo su
  - nvme list
  - nvme Invm-list

```
femu@fvm:~$ sudo su
```

```
[sudo] password for femu:
```

```
root@fvm:/home/femu# nvme list
```

Node	SN	Model	Namespace	Usage	Format	FW Rev
/dev/nvme0n1	serial0	FEMU NVMe Ctrl	1	17.18 GB / 17.18 GB	4 KiB + 0 B	1.0

```
root@fvm:/home/femu#
```

- nvme Invm-init -d nvme0n1

```
root@fvm:/home/femu# nvme Invm-init -d nvme0n1
```

# LightNVM

- FEMU 실행

- LightNVM 사용

- `nvme lnvm-create -d nvme0n1 -t pblk -n target0 -b 0 -e 15`

```
root@fvm:/home/femu# nvme lnvm-create -d nvme0n1 -t pblk -n target0 -b 0 -e 15
```

```
root@fvm:/home/femu# dmesg | tail
```

```
[ 2.526103] ip (1623) used greatest stack depth: 12680 bytes left
[ 2.705798] new mount options do not match the existing superblock, will be ignored
[ 2.909697] new mount options do not match the existing superblock, will be ignored
[ 3.283246] new mount options do not match the existing superblock, will be ignored
[ 3.674324] new mount options do not match the existing superblock, will be ignored
[ 4.053668] new mount options do not match the existing superblock, will be ignored
[ 9.815759] apt-check (2142) used greatest stack depth: 12448 bytes left
[102.737981] random: crng init done
[ 879.917604] pblk init: L2P CRC: 6a94538b
[ 879.999803] pblk(target0): luns:16, lines:4, secs:262144, buf entries:1024
root@fvm:/home/femu# █
```

# LightNVM

## ● FEMU 실행

- LightNVM 사용
- mkfs.ext4 /dev/target0
- mount /dev/target0 /mnt
- df -h

```
root@fvm:/home/femu# mkfs.ext4 /dev/target0
```

```
mke2fs 1.42.13 (17-May-2015)
```

```
Discarding device blocks: done
```

```
Creating filesystem with 3719168 4k blocks and 930240 inodes
```

```
Filesystem UUID: f1b3e748-6b1e-45aa-8999-648d46b27f10
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (32768 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

```
root@fvm:/home/femu# mount /dev/target0 /mnt/
```

```
root@fvm:/home/femu# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	2.0G	0	2.0G	0%	/dev
tmpfs	395M	7.3M	388M	2%	/run
/dev/sda1	40G	4.9G	33G	13%	/
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
tmpfs	395M	0	395M	0%	/run/user/1000
/dev/target0	14G	36M	14G	1%	/mnt

```
root@fvm:/home/femu#
```

```
root@fvm:/home/femu# mkfs.ext4 /dev/target0
```

```
mke2fs 1.42.13 (17-May-2015)
```

```
Discarding device blocks: done
```

```
Creating filesystem with 225280 4k blocks and 56336 inodes
```

```
Filesystem UUID: 0333358a-f638-46cb-9d48-ed97f1f3a354
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840
```

```
Allocating group tables: done
```

```
Writing inode tables: done
```

```
Creating journal (4096 blocks): done
```

```
Writing superblocks and filesystem accounting information: █
```

# LightNVM

- LightNVM 관련자료

- 공식사이트 1 : <http://lightnvm.io/>
- 공식사이트 2 : <https://openchannelssd.readthedocs.io/en/latest/>
- LightNVM Command-Line : <http://lightnvm.io/liblightnvm/index.html>
- pblk Command-Line : <http://lightnvm.io/pblk-tools/>
- NVMe Command-CLI : <https://www.mankier.com/package/nvme-cli>
  
- CNEX FTL 정리 :  
<https://javigongon.files.wordpress.com/2018/07/linuxfast2018.pdf>
  
- OCSSD Spec 문서
  - 1.2 : <http://lightnvm.io/docs/Open-ChannelSSDInterfaceSpecification12-final.pdf>
  - 2.0 : [http://lightnvm.io/docs/OCSSD-2\\_0-20180129.pdf](http://lightnvm.io/docs/OCSSD-2_0-20180129.pdf)

# LightNVM

- LightNVM 관련자료

- LightNVM's QEMU Git : <https://github.com/OpenChannelSSD/qemu-nvme>
- FEMU git : <https://github.com/ucare-uchicago/femu>

- 기타 참고 자료

- [https://www.snia.org/sites/default/files/SDC/2018/presentations/SSS\\_NVM\\_PM\\_NVDIMM/Bjorling\\_Matias\\_Introduction\\_to\\_Open-Channel\\_Denali\\_Solid\\_State\\_Drives.pdf](https://www.snia.org/sites/default/files/SDC/2018/presentations/SSS_NVM_PM_NVDIMM/Bjorling_Matias_Introduction_to_Open-Channel_Denali_Solid_State_Drives.pdf)
- [https://www.flashmemorysummit.com/English/Collaterals/Proceedings/2017/20170809\\_S202D\\_Song.pdf](https://www.flashmemorysummit.com/English/Collaterals/Proceedings/2017/20170809_S202D_Song.pdf)
- <https://events.static.linuxfound.org/sites/events/files/slides/Vault2016-Website.pdf>