

V1.0

[illegible]

```

//  V1.0  Novus OS V1.6-FinalSpec  NKM
  x86_64 UEFI
  /

```

1.

--	--

NKM Novus Kernel Module Novus OS
 Win32 GPU DRM

.nkm

□ NUC□□□□□□□□□□

項目	NUC項目	NKM項目
項目	項目	項目
項目	項目 Ring3	項目 Ring0
項目	項目	項目 Ed25519 項目
項目	entry_point 項目	nkm_init() + nkm_exit() 項目
項目	項目	項目 nkm_unload 項目
項目	項目	項目

2.

--	--	--	--	--	--

0x0000	NKM Header (nkm_hdr_t)	208
0x00D0	Module Dependencies	dep_count × nkm_dep_t (16B/)
...	Segment Table	seg_count × nkm_segment_t (32B/)
...	Symbol Table	sym_count × nkm_sym_t (24B/)
...	Relocation Table	rel_count × nkm_rel_t (16B/)
...	String Table	
...	.text/.data/.rodata	4KiB

3. NKM Header

208 0x00D0 0x0000 packed

“ 128 3.2 — reserved[32] 0xB0 32 0xD0 6.3 0x50..0xB0 208 3.3

3.1

```
#define NKM_MAGIC 0x4E4B4D21 // "NKM!"

typedef struct __attribute__((packed)) {
    uint8_t  magic[4];           // 0x00 "NKM!" = 4E 4B 4D 21
    uint16_t abi_version;        // 0x04 bit[0..7]=minor, bit[8..15]=major
    uint16_t header_size;        // 0x06 = 208 (0xD0)
    uint64_t min_kernel_ver;     // 0x08
    uint32_t module_type;        // 0x10 /FS/...
```

```

uint16_t seg_count;           // 0x14  0000
uint16_t dep_count;          // 0x16  00000000
uint32_t symtab_count;        // 0x18  00000000
uint32_t rel_count;           // 0x1C  00000000
uint32_t strtab_offset;       // 0x20  00000000
uint32_t rel_offset;          // 0x24  00000000
uint32_t init_offset;         // 0x28  nkm_init() 0 .text 0000
uint32_t exit_offset;         // 0x2C  nkm_exit() 0 .text 0000
uint8_t  module_name[32];     // 0x30  0000 '\0' 0000
uint8_t  module_hash[32];     // 0x50  0000 SHA256 0000000000
uint8_t  module_sig[64];      // 0x70  Ed25519 00
uint8_t  reserved[0x20];      // 0xB0  00 32 00
} nkm_hdr_t;                  // 0000 208 00 (0xD0)

```

3.2 00000000

00	00	00	0000
0x00	4	magic	00 NKM! 00000000
0x04	2	abi_version	0000 =major0000 =minor0000 0x0100
0x06	2	header_size	00 208 0xD0 00000000
0x08	8	min_kernel_ver	00000000 0000 >= min_kernel_ver
0x10	4	module_type	0 \$3.3
0x14	2	seg_count	0000 1~16
0x16	2	dep_count	00000000 0=0000
0x18	4	symtab_count	0000
0x1C	4	rel_count	0000
0x20	4	strtab_offset	00000000
0x24	4	rel_offset	00000000
0x28	4	init_offset	nkm_init() 0 .text 00000000
0x2C	4	exit_offset	nkm_exit() 0 .text 00000000
0x30	32	module_name	0000 \0 0000 31 00

Offset	Size	Field	Details
0x50	32	module_hash	SHA256 hash, 64 hex
0x70	64	module_sig	Ed25519 signature, 64 hex
0xB0	32	reserved	Reserved, 0

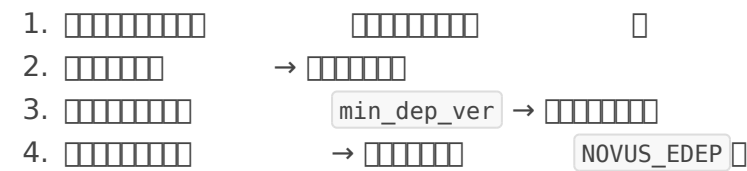
3.3 module_type

Index	Enum	Description
0	NKM_TYPE_DRIVER	VGA/DP/PCI
1	NKM_TYPE_FS	
2	NKM_TYPE_COMPAT	Win32
3	NKM_TYPE_CRYPT0	
4	NKM_TYPE_OTHER	

4.

Header dep_count 16

```
typedef struct __attribute__((packed)) {
    uint32_t dep_name_off; // 0x00 String Table
    uint64_t min_dep_ver; // 0x04
    uint32_t reserved; // 0x0C 0
} nkm_dep_t; // 16
```



5. / /

	NKM
vaddr	0000 00000000 000000000000
0000	00 Ring00000 VM_USER TEXT 0000 DATA 00
0000	0000 00000000 00 nksym 00000000 SDK
0000	0 R_X86_64_64 64 0000000000000000 PC32

00000000 nksym 00000000 vmlinux 0000

```
mm_alloc, mm_free, vm_map, vm_unmap
sched_spawn, sched_exit
vfs_register_fs, vfs_read, vfs_write
hal_display_probe, hal_display_flush
cap_check, log_print, err_report
pci_enumerate, blk_read, blk_write
```

6. Ed25519



6.1 0000

- 0000 SHA256
- 0000 Ed25519 RFC 8032
- 0000 32 00
- 0000 64 00

6.2 000000

```
// 00000000000000000000000000000000
static const ed25519_pubkey_t kernel_trusted_keys[4] = {
    /* 001 */, /* 002 */, /* 003 */, /* 004 */
};
static const uint8_t trusted_key_count = 1; // 0001000000000000
```

- 4 个 16 字节
- 16 字节 = 128 位
- 128 位
- 128 字节

6.3 模块哈希和签名

“模块的 SHA256 哈希 = 模块哈希 + 模块签名”

module_sig[64] 字节

module_hash[32] 字节

模块哈希和签名

1. 初始化 temp_buf
2. 将 temp_buf 填充 0x50 到 96 字节 (32+64) 0
 - 将 module_hash 和 module_sig 填充进去
3. computed_hash = SHA256(temp_buf, 128)
4. 将 computed_hash 与 module_hash 比较
 - 相等 → 模块可信

6.4 模块验证

1. 计算 computed_hash (6.3)
2. 将 computed_hash 与 nkm_hdr.module_hash 比较
3. 将 module_sig[64] 与 Ed25519 公钥比较
4. 将 nkm_hdr.module_hash[32] 与 computed_hash 比较
5. 遍历 kernel_trusted_keys[]
 - if ed25519_verify(module_hash, module_sig, pubkey[i]):
 - 模块可信
6. 模块可信 → 加载

6.5 模块哈希和签名

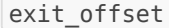
模块哈希和签名

“模块哈希” “模块签名”

“模块哈希+模块签名”

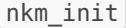
7.

 NKM 

 init_offset /  exit_offset 

```
//   
//  0   
int nkm_init(void);  
  
//   
//  0   
int nkm_exit(void);
```



-  ABI  x86_64 SysV
-  nkm_init  0 → 
-  nkm_exit  IRQ

8.

1. nkm_load(path) 
2.  cap_check(mod.load) → NOVUS_EPERM
3. VFS  208  Header
4.  magic == "NKM!"
5.  abi_version 
6.  min_kernel_ver <= 
7. 
8. 
 - a.  0x50..0xB0  0
 - b.  SHA256
 - c.  module_hash  → NOVUS_ECORRUPT
9.  Ed25519  → NOVUS_EMODULE
 9 
10. 
11. 

12. `nkysym` `nkysym`

13. `nk_init()`

14. `init` `0` → `nkysym`

15. `init` `0` → `nkysym`
- ## 9. `nkysym`
1. `nk_unload(name)` `nkysym`

2. `cap_check(mod.unload)`

3. `nkysym` `dep_count` `nkysym`
`nkysym` → `nkysym` `NOVUS_EBUSY`

4. `nk_exit()`

5. `nkysym`

6. `nkysym`

7. `nkysym`

10. `nkysym`

- `nkysym` `nkysym` `nkysym`
 - `nkysym` `mod.load` `CAP` `nkysym`
 - `nkysym` `NKM_FLAG_ALLOW_UNSIGNED` `nkysym`
 - `nkysym` `WARNING` `nkysym` `PID` `nkysym`
 - `nkysym` `nkysym`
 - `nkysym`

11. `nkysym`

<code>nkysym</code>	<code>nkysym</code>
<code>NOVUS_OK (0)</code>	<code>nkysym</code>
<code>NOVUS_EPERM (-4)</code>	<code>nkysym</code> <code>mod.load</code> / <code>mod.unload</code> <code>CAP</code>
<code>NOVUS_EINVAL (-1)</code>	<code>nkysym</code>
<code>NOVUS_ECORRUPT (-7)</code>	SHA256 <code>nkysym</code>

項目	値
NOVUS_EMODULE (-8)	Ed25519 000000
NOVUS_EBUSY (-5)	00000000
NOVUS_ENOTFOUND (-3)	000000
NOVUS_EDEP (-12)	0000
NOVUS_EIO (-6)	000000
NOVUS_ENOMEM (-2)	000000

12. 0000000000000000

12.1 Header 0000

```
magic:      4E 4B 4D 21  ("NKM!")
abi_version: 00 01      (major=1, minor=0)
header_size: D0 00      (208)
min_kernel_ver: 00 00 00 00 00 00 00 01  (0000 >= 1.0)
module_type: 00 00 00 00  (DRIVER)
seg_count:   02 00      (2)
dep_count:   00 00      (0)
symtab_count: 01 00 00 00
rel_count:   02 00 00 00
strtab_offset: 00 02 00 00  (0x200)
rel_offset:  C0 01 00 00  (0x1C0)
init_offset: 10 00 00 00  (.text+0x10)
exit_offset: 30 00 00 00  (.text+0x30)
module_name: "nvme.ko" 00 00...
module_hash: <3200SHA256>
module_sig:  <6400Ed2551900>
```

12.2 00000000 C

```
#include <novus/nkm.h>
#include <novus/mm.h>
#include <novus/log.h>
```

[illegible][illegible]