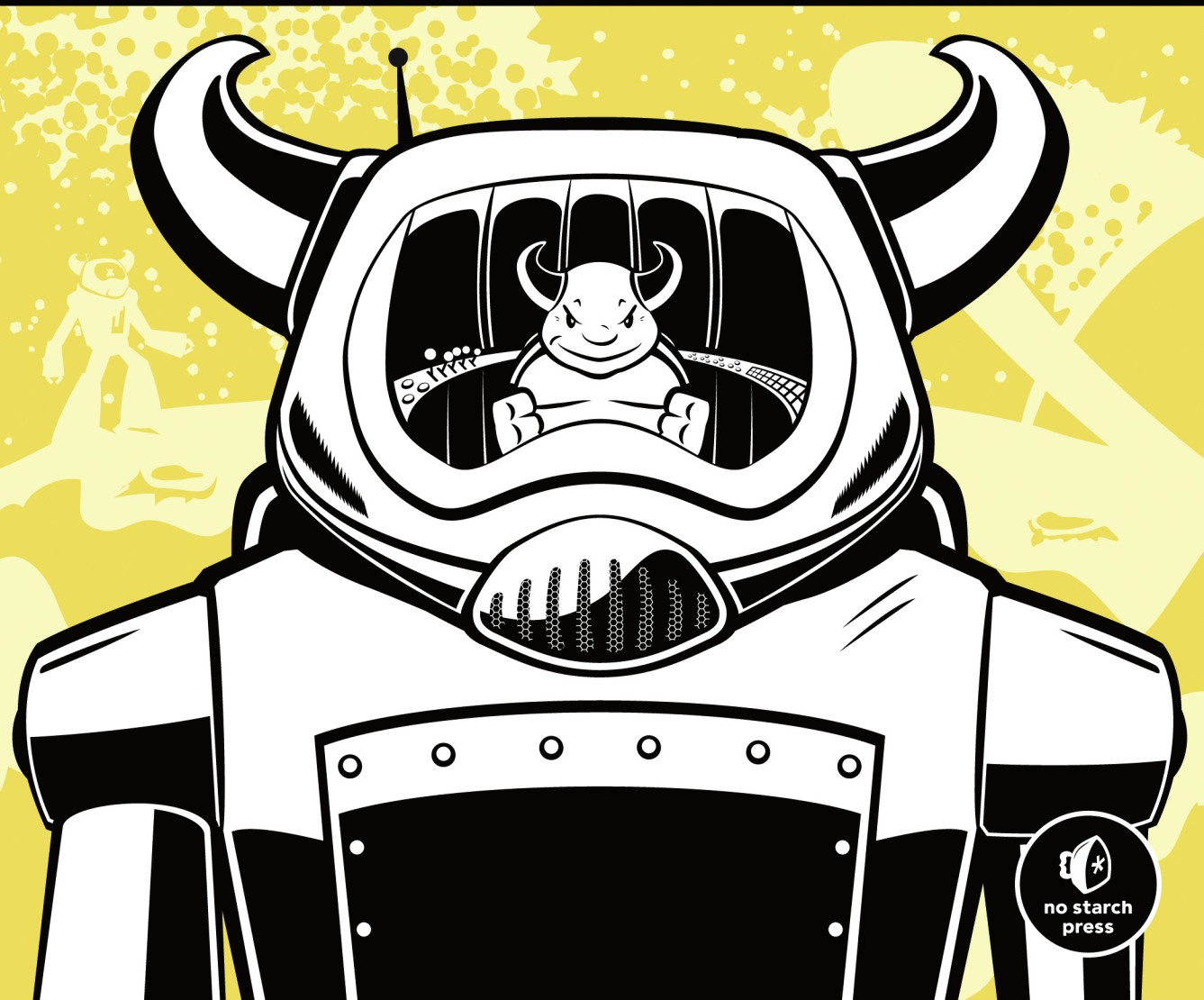


# FREEBSD<sup>®</sup> DEVICE DRIVERS

A GUIDE FOR THE INTREPID

JOSEPH KONG



# CONTENTS IN DETAIL

<b>ABOUT THE AUTHOR AND THE TECHNICAL REVIEWER</b>	<b>xvii</b>
--	-------------

<b>FOREWORD by John Baldwin</b>	<b>xix</b>
---------------------------------	------------

<b>ACKNOWLEDGMENTS</b>	<b>xxi</b>
------------------------	------------

<b>INTRODUCTION</b>	<b>xxiii</b>
---------------------	--------------

Who Is This Book For? .....	xxiii
Prerequisites .....	xxiv
Contents at a Glance .....	xxiv
Welcome Aboard! .....	xxv

<b>1</b>	
<b>BUILDING AND RUNNING MODULES</b>	<b>1</b>

Types of Device Drivers.....	1
Loadable Kernel Modules.....	2
Module Event Handler .....	2
DECLARE_MODULE Macro .....	3
Hello, world! .....	5
Compiling and Loading .....	6
Character Drivers .....	7
d_foo Functions.....	7
Character Device Switch Table .....	8
make_dev and destroy_dev Functions .....	9
Mostly Harmless.....	10
echo_write Function.....	12
echo_read Function .....	13
echo_modevent Function .....	14
DEV_MODULE Macro.....	15
Don't Panic.....	15
Block Drivers Are Gone.....	15
Conclusion .....	16

<b>2</b>	
<b>ALLOCATING MEMORY</b>	<b>17</b>

Memory Management Routines.....	17
malloc_type Structures .....	19
MALLOC_DEFINE Macro .....	19
MALLOC_DECLARE Macro.....	20

Tying Everything Together .....	20
Contiguous Physical Memory Management Routines .....	22
A Straightforward Example .....	23
Conclusion .....	25

### **3**

## **DEVICE COMMUNICATION AND CONTROL**

### **27**

ioctl .....	28
Defining ioctl Commands .....	29
Implementing ioctl .....	30
echo_write Function .....	34
echo_set_buffer_size Function .....	35
echo_ioctl Function .....	36
echo_modevent Function .....	36
Don't Panic.....	37
Invoking ioctl .....	37
sysctl .....	40
Implementing sysctls, Part 1 .....	41
sysctl Context Management Routines .....	44
Creating Dynamic sysctls.....	44
SYSCTL_STATIC_CHILDREN Macro .....	47
SYSCTL_CHILDREN Macro .....	47
Implementing sysctls, Part 2 .....	47
sysctl_set_buffer_size Function .....	50
Don't Panic.....	52
Conclusion .....	52

### **4**

## **THREAD SYNCHRONIZATION**

### **53**

A Simple Synchronization Problem .....	54
A More Complex Synchronization Problem .....	54
race_new Function .....	58
race_find Function .....	58
race_destroy Function .....	59
race_ioctl Function .....	59
race_modevent Function.....	60
The Root of the Problem .....	61
Preventing Race Conditions .....	65
Mutexes .....	65
Spin Mutexes.....	65
Sleep Mutexes .....	66
Mutex Management Routines.....	66
Implementing Mutexes .....	68
race_modevent Function.....	71
Don't Panic.....	72
Shared/Exclusive Locks.....	73
Shared/Exclusive Lock Management Routines.....	73
Implementing Shared/Exclusive Locks .....	75
Reader/Writer Locks .....	78
Reader/Writer Lock Management Routines .....	78

Condition Variables.....	79
Condition Variable Management Routines.....	80
General Guidelines .....	81
Avoid Recursing on Exclusive Locks .....	81
Avoid Holding Exclusive Locks for Long Periods of Time .....	82
Conclusion .....	82

## **5 DELAYING EXECUTION 83**

Voluntary Context Switching, or Sleeping.....	83
Implementing Sleeps and Condition Variables .....	85
sleep_modevent Function.....	88
load Function.....	89
sleep_thread Function .....	90
sysctl_debug_sleep_test Function .....	91
unload Function .....	91
Don't Panic.....	92
Kernel Event Handlers.....	92
Callouts .....	94
Callouts and Race Conditions .....	96
Taskqueues.....	96
Global Taskqueues .....	97
Taskqueue Management Routines.....	97
Conclusion .....	98

## **6 CASE STUDY: VIRTUAL NULL MODEM 99**

Prerequisites .....	100
Code Analysis .....	100
nmdm_modevent Function .....	103
nmdm_clone Function .....	104
nmdm_alloc Function .....	105
nmdm_outwakeup Function .....	106
nmdm_task_tty Function.....	106
nmdm_inwakeup Function .....	108
nmdm_modem Function .....	108
nmdm_param Function.....	109
nmdm_timeout Function.....	111
bits_per_char Function .....	111
Don't Panic.....	112
Conclusion .....	112

## **7 NEWBUS AND RESOURCE ALLOCATION 113**

Autoconfiguration and Newbus Drivers.....	113
device_foo Functions .....	114
Device Method Table.....	115
DRIVER_MODULE Macro .....	116

Tying Everything Together .....	117
foo_pci_probe Function.....	120
foo_pci_attach Function .....	120
d_foo Functions.....	121
foo_pci_detach Function .....	121
Don't Panic.....	122
Hardware Resource Management .....	122
Conclusion .....	124

## **8 INTERRUPT HANDLING 125**

Registering an Interrupt Handler.....	125
Interrupt Handlers in FreeBSD .....	126
Implementing an Interrupt Handler.....	127
pint_identify Function.....	132
pint_probe Function .....	132
pint_attach Function.....	133
pint_detach Function.....	134
pint_open Function .....	134
pint_close Function .....	135
pint_write Function .....	136
pint_read Function.....	136
pint_intr Function.....	137
Don't Panic.....	138
Generating Interrupts on the Parallel Port.....	138
Conclusion .....	139

## **9 CASE STUDY: PARALLEL PORT PRINTER DRIVER 141**

Code Analysis .....	141
lpt_identify Function.....	146
lpt_probe Function.....	146
lpt_detect Function.....	147
lpt_port_test Function .....	148
lpt_attach Function .....	148
lpt_detach Function .....	150
lpt_open Function .....	151
lpt_read Function .....	153
lpt_write Function .....	154
lpt_intr Function.....	156
lpt_timeout Function.....	158
lpt_push_bytes Function.....	158
lpt_close Function .....	159
lpt_ioctl Function .....	160
lpt_request_ppbus Function.....	162
lpt_release_ppbus Function.....	162
Conclusion .....	163

<b>10</b>	<b>MANAGING AND USING RESOURCES</b>	<b>165</b>
I/O Ports and I/O Memory .....		165
Reading from I/O Ports and I/O Memory .....		166
Writing to I/O Ports and I/O Memory .....		167
Stream Operations .....		169
Memory Barriers .....		172
Tying Everything Together .....		172
led_identify Function .....		177
led_probe Function .....		177
led_attach Function .....		178
led_detach Function .....		178
led_open Function .....		179
led_close Function .....		180
led_read Function .....		180
led_write Function .....		181
Conclusion .....		182
<b>11</b>	<b>CASE STUDY: INTELLIGENT PLATFORM MANAGEMENT INTERFACE DRIVER</b>	<b>183</b>
Code Analysis .....		183
ipmi_pci_probe Function .....		185
ipmi_pci_match Function .....		186
ipmi_pci_attach Function .....		187
ipmi2_pci_probe Function .....		189
ipmi2_pci_attach Function .....		189
Conclusion .....		191
<b>12</b>	<b>DIRECT MEMORY ACCESS</b>	<b>193</b>
Implementing DMA .....		194
Initiating a DMA Data Transfer .....		196
Dismantling DMA .....		196
Creating DMA Tags .....		197
Tearing Down DMA Tags .....		198
DMA Map Management Routines, Part 1 .....		199
Loading (DMA) Buffers into DMA Maps .....		199
bus_dma_segment Structures .....		199
bus_dmamap_load Function .....		200
bus_dmamap_load_mbuf Function .....		201
bus_dmamap_load_mbuf_sg Function .....		201
bus_dmamap_load_uio Function .....		202
bus_dmamap_unload Function .....		202
DMA Map Management Routines, Part 2 .....		202
A Straightforward Example .....		203
Synchronizing DMA Buffers .....		205
Conclusion .....		205

<b>13</b>	<b>STORAGE DRIVERS</b>	<b>207</b>
disk Structures.....		207
Descriptive Fields .....		208
Storage Device Methods .....		209
Mandatory Media Properties .....		209
Optional Media Properties .....		209
Driver Private Data .....		210
disk Structure Management Routines .....		210
Block I/O Structures .....		210
Block I/O Queues .....		212
Tying Everything Together .....		213
at45d_attach Function .....		217
at45d_delayed_attach Function .....		218
at45d_get_info Function.....		219
at45d_wait_for_device_ready Function .....		220
at45d_get_status Function .....		220
at45d_strategy Function.....		221
at45d_task Function .....		221
Block I/O Completion Routines .....		223
Conclusion .....		223

<b>14</b>	<b>COMMON ACCESS METHOD</b>	<b>225</b>
How CAM Works .....		226
A (Somewhat) Simple Example .....		227
mfip_attach Function.....		234
mfip_detach Function.....		235
mfip_action Function.....		236
mfip_poll Function .....		238
mfip_start Function .....		238
mfip_done Function .....		240
SIM Registration Routines .....		242
cam_simq_alloc Function.....		242
cam_sim_alloc Function .....		242
xpt_bus_register Function .....		243
Action Routines .....		243
XPT_PATH_INQ .....		243
XPT_RESET_BUS .....		245
XPT_GET_TRAN_SETTINGS .....		246
XPT_SET_TRAN_SETTINGS .....		249
XPT_SCSI_IO .....		250
XPT_RESET_DEV .....		255
Conclusion .....		255

<b>15</b>	<b>USB DRIVERS</b>	<b>257</b>
About USB Devices.....		257
More About USB Devices.....		258
USB Configuration Structures.....		259
Mandatory Fields.....		260
Optional Fields.....		260
USB Transfer Flags.....		261
USB Transfers (in FreeBSD).....		262
USB Configuration Structure Management Routines.....		264
USB Methods Structure.....		265
Tying Everything Together.....		266
ulpt_probe Function.....		270
ulpt_attach Function.....		271
ulpt_detach Function.....		273
ulpt_open Function.....		273
ulpt_reset Function.....		274
unulpt_open Function.....		275
ulpt_close Function.....		276
ulpt_ioctl Function.....		276
ulpt_watchdog Function.....		277
ulpt_start_read Function.....		277
ulpt_stop_read Function.....		278
ulpt_start_write Function.....		278
ulpt_stop_write Function.....		278
ulpt_write_callback Function.....		279
ulpt_read_callback Function.....		280
ulpt_status_callback Function.....		281
Conclusion.....		282

<b>16</b>	<b>NETWORK DRIVERS, PART 1: DATA STRUCTURES</b>	<b>283</b>
Network Interface Structures.....		283
Network Interface Structure Management Routines.....		286
ether_ifattach Function.....		287
ether_ifdetach Function.....		288
Network Interface Media Structures.....		289
Network Interface Media Structure Management Routines.....		289
Hello, world!.....		291
mbuf Structures.....		293
Message Signaled Interrupts.....		294
Implementing MSI.....		294
MSI Management Routines.....		297
Conclusion.....		297



## 17

### NETWORK DRIVERS, PART 2:

#### PACKET RECEPTION AND TRANSMISSION

299

Packet Reception .....	299
em_rxeof Function .....	300
em_handle_rx Function .....	303
Packet Transmission .....	304
em_start_locked Function .....	304
em_txeof Function .....	305
Post Packet Transmission .....	307
Conclusion .....	308

#### REFERENCES

309

#### INDEX

311