Index

Numbers

32-bit addressing
  address size, 43
  page size, 44
64-bit addressing
  address size, 43
  page size, 44

A

_ABI_SOURCE compiler variable, 136
address range, 43
address space, 12, 43-48
  copy on write, 16
  copy-on-write pages, 48
  defining addresses, 45
  duplicated by **fork()**, 16
  functions that change, 85
  heap segment, 44
  interrogating, 48
  limits of, 46
  lowest used address, 44
  of VME bus devices, 166
  read-only pages, 47
  replaced by **exec()**, 16
  resident set size, 47
  shared by lightweight processes, 17
  stack segment, 44
  text segment, 44
  virtual size of, 45
affinity scheduling, 72
  compared to static assignment, 77
affinity value, 72
aio_cancel(), 140
aio_error(), 141, 142
  example code, 212
aio_fsync(), 140
aio_read(), 140
  example code, 213
  from callback, 143
  implies **aio_init()**, 138
aio_return(), 142
  example code, 213
aio_sgi_init(), 138
  example code, 210
aio_suspend(), 142
  example code, 214
aio_write(), 140
  example code, 213
  from callback, 143
  implies **aio_init()**, 138
aircraft simulator, 3
Application Binary Interface. *See* MIPS ABI
asynchronous I/O, 7, 19, 134-147
  aiocb structure, 137, 142
  difference between 5.3 and 5.2, 136
  example code, 203-??, 204-??
  in IRIX 6.0, 136
  initializing, 138
  multiple operations to one file, 146
  not compatible with guaranteed rate, 155

notification methods, 142
POSIX 1003.1b-1993, 136
request priority no longer supported, 137
signal use, 142
average data rate, 6

B

backing store, 45, 47
barrier, 34-35
starting Frame Scheduler, 117
barrier(), 34
example code, 211
batch priority, 70
brk(), 45
modifies address space, 85
bus
processor. *See* processor bus
bus,VME. *See* VME bus

C

cache
address mapping in Challenge/Onyx, 53
affinity scheduling, 72
architecture, 11
effect of miss, 52
management, 52-54
multiprocessor conflicts, 54
warming up in first frame, 102
cache coherency, 11
cacheflush(), 85
cache line, 52
CD-ROM audio library, 163
Challenge/Onyx architecture, 9
cache address mapping, 53
cache management in, 52
clock_gettime(), 64
concurrent execution, 11
copy on write page statue, 16
CPU
assign interrupt to, 81
assign process to, 83
CPU 0 not used by Frame Scheduler, 96
isolating from sprayed interrupts, 81
isolating from TLB interrupts, 84
making nonpreemptive, 86
relation to bus and memory, 10
restricting to assigned processes, 24, 82
current directory, 15
cycle counter, 40
as Frame Scheduler time base, 108
drift rate of, 40, 66
example program, 177-184
in interval timer management, 60
mapping into memory, 65
precision of, 40, 65
used for timestamp, 65

D

data collection system, 2, 5-7
average data rate, 6
input rate, 6
peak data rate, 5
requirements on, 5
data segment
locking, 50
DAT audio library, 163
date command, 64, 65
deadline scheduling, 23, 74
degrading priority, 22, 69
/dev/ei, 173
/dev file system, 158
device

defined in */dev*, 158
device numbers, 158
opening, 19, 158
device driver, 13, 14
as Frame Scheduler time base, 126-131
entry points to, 159
for VME bus master, 169
generic SCSI, 162
in synchronous input, 134
reference pages, 159
tape, 162
device interrupt, 13
device service time, 14, 88, 91
not guaranteed, 91
device special file, 158
direct disk output, 148
disk output
synchronous direct, 148
synchronous unbuffered, 147
dispatch cycle time, 88, 92
dlopen(), 85
DMA engine for VME bus, 170
performance, 170
DMA mapping, 167
DMA to VME bus master devices, 169
drift rate of cycle counter, 66
dslib, 162
DSO, 85
DSO, text segment for, 44
dynamic shared object. *See* DSO

E

/etc/autoconfig command, 81
exec(), 16
external interrupt, 42, 173-??
with Frame Scheduler, 108

F

fasthz tuning parameter, 61
effect of truncation, 62
fcntl()
example code, 152
file, mapping into memory, 51
file descriptor
of a device, 158
returned by **open()**, 19
with asynchronous I/O, 137
with guaranteed-rate I/O, 153
fork(), 16
defines address space, 45
example, 16
new address space copy-on-write, 48
rate guarantee not inherited, 154
frame interval, 3
frame rate, 3
of plant control simulator, 4
of virtual reality simulator, 4
Frame Scheduler, 25-29, 95-131
advantages, 27
and cycle counter, 108
and external interrupt, 108
and R4000 timer, 107
and vertical sync, 108
background discipline, 111
continuable discipline, 112
CPU 0 not used by, 96
definition of frame, 26
design process, 27
device driver initialization, 128
device driver interface, 126-131
device driver interrupt, 131
device driver termination, 130
device driver use, 126
example code, 225-??
exception handling, 118-121
frs_run flag, 102

frs_yield flag, 102
FRS control process, 98, 104
interface to, 98-101
interval timers not used with, 125
major frame, 96
minor frame, 96
multiple synchronized, 104
overrun exception, 110, 118
overrunnable discipline, 111
pausing, 105
process outline for multiple, 115
process outline for single, 114
process structure, 101
realtime discipline, 110
scheduling disciplines, 110-113
scheduling rules of, 102
signals produced by, 123, 124
software interrupt to, 109
starting up, 105
time base selection, 26, 107
underrunable discipline, 111
underrun exception, 110, 118
using consecutive minor frames, 112
warming up cache, 102
frs_create_master(), 114, 116, 127
frs_create_slave(), 117
frs_destroy(), 115, 116, 117
frs_driver_export(), 127
frs_enqueue(), 102, 110, 114, 116, 117
frs_handle_driverintr(), 131
frs_join(), 101, 105, 114, 116, 117
frs_resume(), 105
frs_setattr()
example code, 120
frs_start(), 105, 114, 116, 117
frs_stop(), 105
frs_userinter(), 109
frs_yield, 101
frs_yield()
with overrunable discipline, 112
FRS control process, 98, 104
design of, 114-117
receives signals, 124
fsync(), 135, 148

G

gang scheduling, 23, 73
getpagesize(), 44, 48
getrlimit(), 46
gettimeofday(), 40, 63
example code, 184-186
grio_remove_request(), 154
grio_request(), 153
example code, 224
GRIO. *See* guaranteed-rate I/O
ground vehicle simulator, 4
group ID, 15
guaranteed-rate I/O, 7, 150-156
creating a real-time file, 152
example code, 221-225
hard guarantee, 155
requesting a guarantee, 153
requires XFS, 150
requires XLV volume, 151
soft guarantee, 155
tied to PD and I-node, 154
video on demand guarantee, 156

H

hardware latency, 88, 89
hardware simulator, 5
heap segment, 44, 45
HZ value in timer management, 61

I

inline functions and cache management, 53
input
  synchronous, 134
interchassis communication, 41-42
interprocess communication, 29-38
interrupt
  assign to CPU, 81
  clock comparator, 60
  controlling distribution of, 25
  device, 13
  external. *See* external interrupt
  group. *See* interrupt group
  isolating CPU from, 81
  latency, 14
  periodic timer, 60
  propogation delay, 89
  response time. *See* interrupt response time
  spraying, 14, 81
  TLB, 13, 84
  validity fault, 47
  vertical sync, 82, 108
  VME bus, 14, 89
interrupt group, 107
  Frame Scheduler passes to device driver, 128
  not used with cycle counter, 108
  to distribute external interrupt, 109
  to distribute vertical sync, 108
interrupt response time, 14, 87-93
  200 microsecond guarantee, 87
  components, 88
  device service not guaranteed, 91
  device service time, 91
  dispatch cycle, 92
  hardware latency, 89
  kernel service not guaranteed, 90
  restrictions on processes, 90
  software latency, 90
interrupts
  unavoidable from timer, 80
interval timer, 38, 55-63
  cycle counter used to manage, 60
  example, 57
  management by kernel, 59
  not used with Frame Scheduler, 39, 125
ioctl()
  and device driver, 159
IPL statement, 81
IRIS InSight, xxi
IRIX
  kernel, 10
IRIX overview, 9-19
irix.sm configuration file, 81
itimer. *See* interval timer

K

kernel
  address space limits in, 46
  affinity scheduling, 72
  critical section, 90
  deadline scheduling, 74
  degrading priority, 69
  gang scheduling, 73
  interrupt response time, 90
  multiprocessor use, 10, 11
  optimizations in, 22
  originates signals, 36
  priority assignment, 69
  process management, 15
  real-time features, 21-25
  scheduling, 68
  scheduling assumptions, 17
  scheduling queues, 70
  tick, 68
  timer management, 59
  time slice, 68

kernel address space, 43

L

latency
  hardware, 88, 89
  software, 88, 90
lightweight process
  created with **sproc()**, 17
  less work to create, 17
  preferred for real-time use, 17
limits command, 46
linked lists and cache management, 53
lio_listio(), 140
Locality of Reference, 52
locality of reference, 11
lock, 33-34
  defined, 33
  effect of gang scheduling, 73
  metering, 34
  set by spinning, 33
  used by kernel, 11
locking virtual memory, 24
lseek()
  with asynchronous I/O, 137
  with guaranteed-rate I/O, 156

M

major device number, 158
major frame, 96
malloc(), 45
MAP_AUTOGROW flag, 85
MAP_LOCAL flag, 85
memalign(), 53
memory, 43-54
  address ranges of, 43
  backing store for, 45
  hierarchy, 11
  interrogating size of, 48
  locking pages in, 49-52
  main, 10
  page, 44
  page size, 12
  shared. *See* shared memory
  virtual, 12
memory mapping, 45
  for I/O, 133
  locking mapped file, 51
  of cycle counter, 65
metering lock use, 34
metering semaphore use, 32
minor device number, 158
minor frame, 96, 102
MIPS ABI
  asynchronous I/O, 136
mmap(), 85
mpadmin command
  assign clock processor, 79
  make CPU nonpreemptive, 87
  query fasthz CPU, 80
  restrict CPU, 83
  set fasthz CPU, 80
  unrestrict CPU, 83
mpin(), 49, 52
mprotect(), 85
multiprocessor architecture, 10
  affinity scheduling, 72
  and Frame Scheduler, 104
munmap(), 85
munpin(), 50, 52
mutual exclusion primitive, 35

N

NDPHIMAX constant, 69
NDPHIMIN constant, 69
ndpri_hilim tuning parameter, 71
ndpri_lolim tuning parameter, 71
newbarrier(), 34
"nice" value, 69
NOINTR statement, 81
nondegrading batch priority, 70
nondegrading priorities, 22
nondegrading real-time priority, 71
npri command, 70
  deadline scheduling, 74
  nondegrading priority, 71

O

open(), 19
  example code, 152
  of a device, 158
operator
  affected by transport delay, 3
  in virtual reality simulator, 4
  of simulator, 2
output
  synchronous, 135
  to disk is buffered, 147
overrun in data collection system, 5
overrun in Frame Scheduler, 110

P

page
  copy on write, 48
  locking, 49
  read-only, 47
page fault, 12
  causes TLB interrupt, 85
  prevent by locking memory, 24, 49
page size, 12, 44
page validation, 47
peak data rate, 5
performance effects of cache, 52
performance tools, 54
PIO access to VME devices, 168
PIO address mapping, 167
pixie command, 54
plant control simulator, 4
plock(), 49
  example of, 50
poll(), 32
power plant simulator, 4
prctl(), 48
priority, 69-72
  degrading, 22, 69
  looping process can halt system, 72
  nondegrading, 22
  nondegrading batch, 70
  nondegrading real-time, 71
  ranges of, 69
process, 15-19
  address space, 44
  assigned to processor, 76
  assign to CPU, 83
  attributes, 15
  attributes initialized by **exec()**, 16
  blocked by I/O, 18
  composition, 15
  created with **fork()**, 16
  FRS control, 98
  lightweight. *See* lightweight process
  mapping to CPU, 24
  "nice" value, 69
  priority of, 69

time slice, 68
process control, 2
process creation, 16
process group, 23
and gang scheduling, 74
process ID, 15
process id
identifies rate guarantee, 154
processor bus
capacity, 10
diagram, 10
processor set, 23, 76-78
contradiction, 78
process scheduling, 17
prof command, 54
propogation delay. *See* hardware latency
ps command, 45
*ps*command, 47
pset command, 70, 76-78
and restricted CPU, 82
contradictions, 78
punlock(), 50

Q

queue, scheduling, 69, 70

R

R4000 timer, 107
REACT, xxi
REACT/Pro, xxi
read()
and device driver, 159
synchronous, 134
with guaranteed-rate I/O, 153
real-time priority, 71
real-time program
and Frame Scheduler, 25
and scheduler assumptions, 18
data collection, 2, 5-7
defined, 1
disk I/O by, 133
frame rate, 2
lightweight processes preferred, 17
process control, 2
simulator, 1, 2-5
types of, 1-7
reflective shared memory, 42
resident set size, 47
response time. *See* interrupt response time
restricting a CPU, 82
rlimit kernel parameter, 46
rtnetd daemon, priority of, 72
runon command, 83

S

schedctl(), 70, 71, 74, 75, 84
example code, 71, 74, 200-203
with Frame Scheduler, 100
scheduling, 68-76
affinity type, 72
assumptions, 17
deadline type, 23, 74
degrading priority, 22
gang type, 23, 73
nondegrading priority, 22
scheduling discipline, 78
See also Frame Scheduler scheduling disciplines
scheduling queue, 69, 70
processor set assigned to, 77
SCSI interface, 160-163
generic device driver, 162

segment
heap, 44
locking, 50
lowest address, 44
stack, 44
text, 44
semaphore, 31-33
defined, 31
IRIX implementation, 32
metering, 32
pollable, 32
portable implementation, 32
used by kernel, 11
used with interval timer, 58
semget(), 32
semop(), 33
setitimer(), 38, 56
setitimer()example code, 58
setrlimit(), 46
sginap(), 34
example code, 212
shared memory, 29-31
IRIX implementation, 29
portable implementation, 30
reflective, 42
usconfig(), 30
usinit(), 30
shmat(), 30
shmctl(), 30, 85
shmget(), 30, 85
sigaction()
example code, 57, 217
with asynchronous I/O, 143
SIGALRM
from interval timer, 39, 57
SIGBUS
on reference to undefined page, 45
sigevent structure, 137
SIGKILL
possible when locking pages, 50
signal, 36-38
delivery priority, 37
generated from interval timer, 56
generated in asynchronous I/O, 142
latency, 122
SIGALRM, 39, 57
SIGBUS, 45
SIGKILL, 50
signal numbers, 37
SIGSEGV, 47
SIGUSR1, 124
SIGUSR2, 124
with Frame Scheduler, 122
signal handler
as process attribute, 15
when setting up Frame Scheduler, 114, 116, 117
SIGRTMIN on dequeue, 124
SIGSEGV
on attempt to change read-only page, 47
sigsuspend(), 142
SIGUSR1
on underrun, 124
SIGUSR2
on overrun, 124
sigwait(), 142
simulator, 1, 2-5
aircraft, 3
control inputs to, 2, 4
frame rate of, 2, 4
ground vehicle, 4
hardware, 5
operator of, 2
plant control, 4
state display, 2
virtual reality, 4
world model in, 2
sockets, 41

software latency, 88, 90
spin lock, 33
sproc(), 17
  CPU assignment inherited, 84
  modifies address space, 85
  rate guarantee not inherited, 154
sprocsp(), 85
  example code, 210
stack segment, 44, 45
  locking, 50
striped volume, 156
structures and cache management, 53
swap, 45, 47
swapctl(), 48
synchronous disk output, 147
sysconf(), 48
sysmp(), 48, 84
  assign process to CPU, 83
  example code, 79, 80, 83, 84, 87
  isolate TLB interrupts, 84
  make CPU nonpreemptive, 87
  number of CPUs, 82
  restrict CPU, 83
  run process on any CPU, 84
  set fasthz CPU, 80
sys/param.h, 68
sys/schedctl.h, 69
syssgi
  set flush interval, 150
syssgi(), 48, 65
systune command, 46, 61

T

tape device, 162
telemetry, 2
test_and_set, 35
text segment, 44
  loaded from program file, 47
  locking, 50
  read-only, 47
tick, 68
  disabling, 86
time base for Frame Scheduler, 107
timer interrupts unavoidable, 80
timer management, 59
time slice, 68
timestamp, 40, 63-66
  comparing methods, 66
  from **clock_gettime()**, 64
  from cycle counter, 65
  from **gettimeofday()**, 63
TLB, 13
TLB update interrupt, 13, 84
translation lookaside buffer. *See* TLB
transport delay, 3

U

udmalib, 170-171
underrun, in Frame Scheduler, 110
usconfig(), 30
user ID, 15
usinit(), 30
  arena for barrier, 34
  arena for lock, 34
  arena for semaphore, 32
usnewlock(), 34
usnewsema(), 32
uspsema, 32
uspsema()
  example code, 217
ussetlock(), 34
usunsetlock(), 34

usvsema, 32
usvsema()
  example code, 57, 216

V

validity fault, 47
vertical sync interrupt, 82, 108
video on demand (VOD). *See* guaranteed-rate I/O, video on demand
virtual address space. *See* address space
virtual memory, 12
  loading pages, 47
  locking, 24
  page fault, 12
  See also memory
virtual page number, 44
virtual reality simulator, 4
virtual size, 45
VME bus, 164-171
  address space mapping, 166
  and process scheduling, 19
  assign interrupt to CPU, 81
  configuration, 165
  data input rate, 6
  DMA mapping, 167
  DMA to master devices, 169
  hardware latency of, 89
  interrupt levels, 14
  performance, 168, 169, 170
  PIO access, 168
  PIO address mapping, 167
  udmalib, 170
VPN. *See* virtual page number

W

write()
  and device driver, 159
  direct, 148
  synchronous, 135, 148
  with guaranteed-rate I/O, 152, 153