Processes

Operating Systems

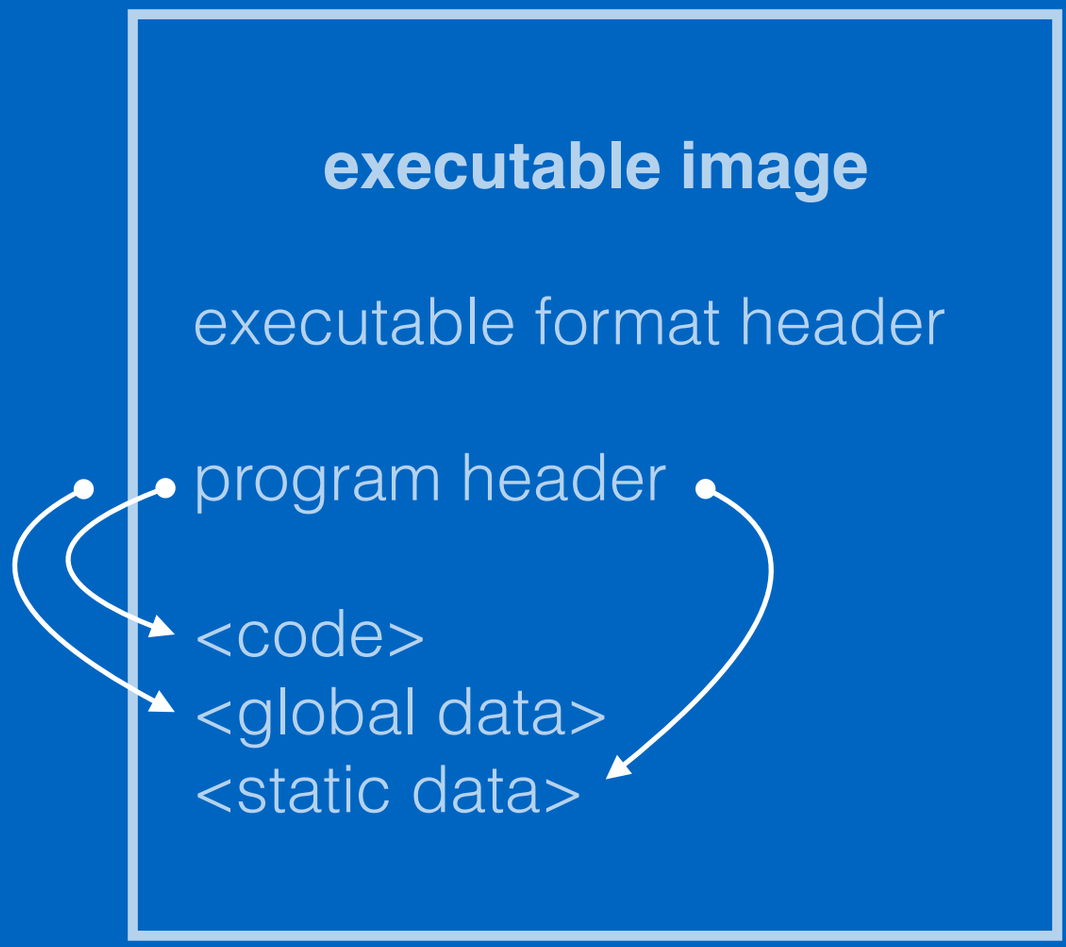
```
top - 14:31:50 up 17 min,  1 user,  load average: 0.38, 0.21, 0.17
Tasks: 243 total,   1 running, 242 sleeping,   0 stopped,   0 zombie
%Cpu0  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu8  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu9  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu10 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu11 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu12 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu15 :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
GiB Mem:   220.414 total,    0.846 used,  219.567 free,    0.015 buffers
GiB Swap:    0.000 total,    0.000 used,    0.000 free.    0.256 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0   34.2m   5.0m   3.6m S   0.0  0.0   0:12.28 systemd
    2 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.24 ksoftirqd/0
    5 root       0 -20    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.48 kworker/u128:0
    7 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.46 rcu_sched
    8 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 rcu_bh
    9 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.06 rcuos/0
   10 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 rcuob/0
   11 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.42 migration/0
   12 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:04.25 watchdog/0
   13 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:06.80 watchdog/1
   14 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.63 migration/1
   15 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 ksoftirqd/1
   17 root       0 -20    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 kworker/1:0H
   18 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.02 rcuos/1
   19 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 rcuob/1
   20 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 watchdog/2
   21 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 migration/2
   22 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 ksoftirqd/2
   24 root       0 -20    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 kworker/2:0H
   25 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 rcuos/2
   26 root      20   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.00 rcuob/2
   27 root      rt   0    0.0m   0.0m   0.0m S   0.0  0.0   0:00.10 watchdog/3
```

executable format

defines the structure of an executable stored on a disk or in memory

usually produced by a program linker


executable image
executable format header
program header
<code>
<global data>
<static data>

common executable formats

PE (Windows)
ELF (Linux…)
Mach-O (Mac OS X)

process descriptor

process structure

process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…

cpu state

registers
program counter
stack pointer
…

memory map

process page table

kernel maintains a list of process descriptors.

process structure

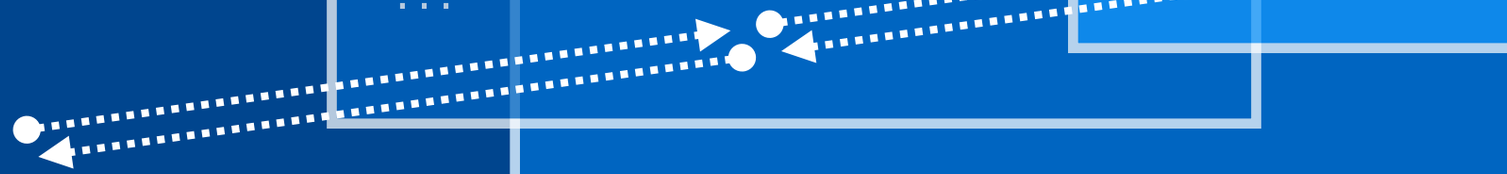
process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…

process structure

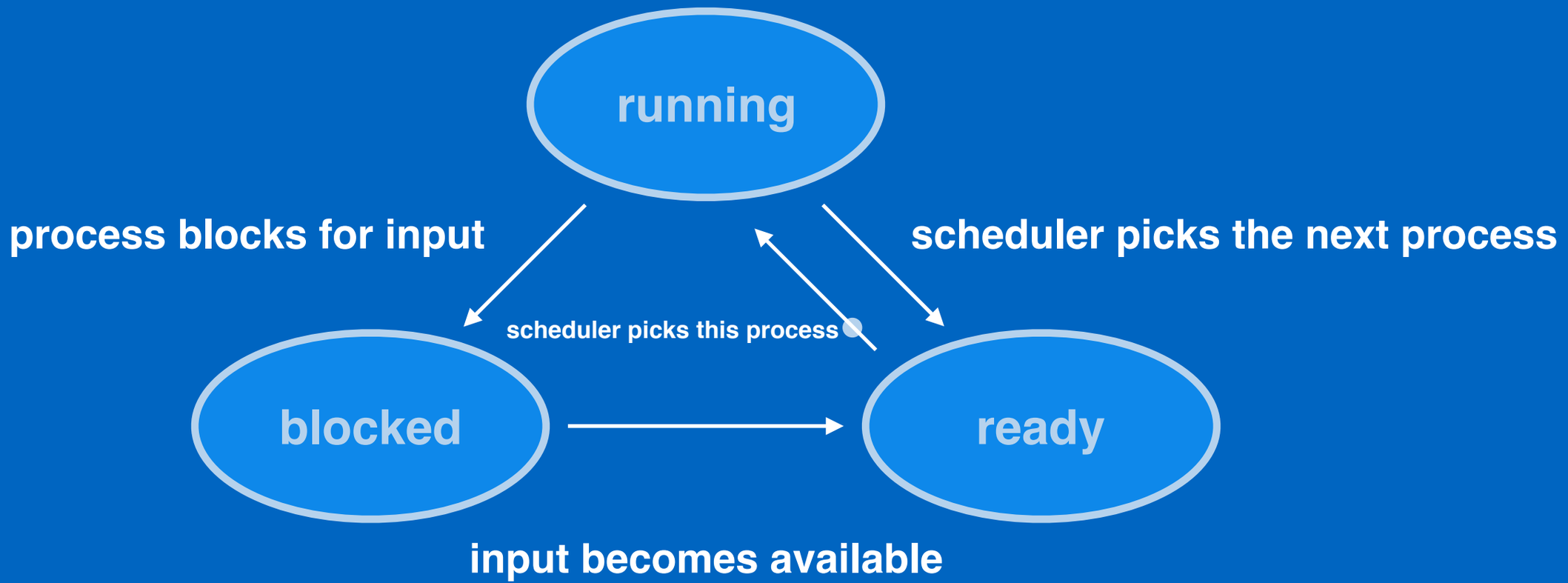
process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…

process structure

process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…



process states


running
process blocks for input
scheduler picks the next process
scheduler picks this process
blocked
ready
input becomes available

context switch

reasons to switch
between executable units

a new process is created

the process exits

the process blocks
waits for input/output
waits to enter a critical section

a hardware interrupt occurs

input/output device interrupts

clock interrupts

steps to switch a context

1. save the current CPU state to memory

process structure

ocess identification
ocess state
ocess priority

u state
tual memory
ened files

de
bal data
atic data

rent
rking directory
it status

process structure

process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…

CPU

reg0
reg1
instruction pointer
stack pointer
…

memory

2. load a CPU state for the next executable unit from memory

process structure

ocess identification
ocess state
ocess priority

u state
tual memory
ened files

de
bal data
atic data

rent
rking directory
it status

process structure

process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…

CPU

reg0
reg1
instruction pointer
stack pointer
…

memory

threads

thread (or a lightweight process) is an executable unit that shares certain elements with its owning process.

shared memory address space
shared opened files
shared global data
...

private state
private registers
private stack
...

process structure

process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
threads
working directory
exit status
…

thread structure

thread identification
thread state
thread priority
…
cpu state
termination status
…
static data
…

thread structure

thread identification
thread state
thread priority
…
cpu state
termination status
…
static data
…

thread structure

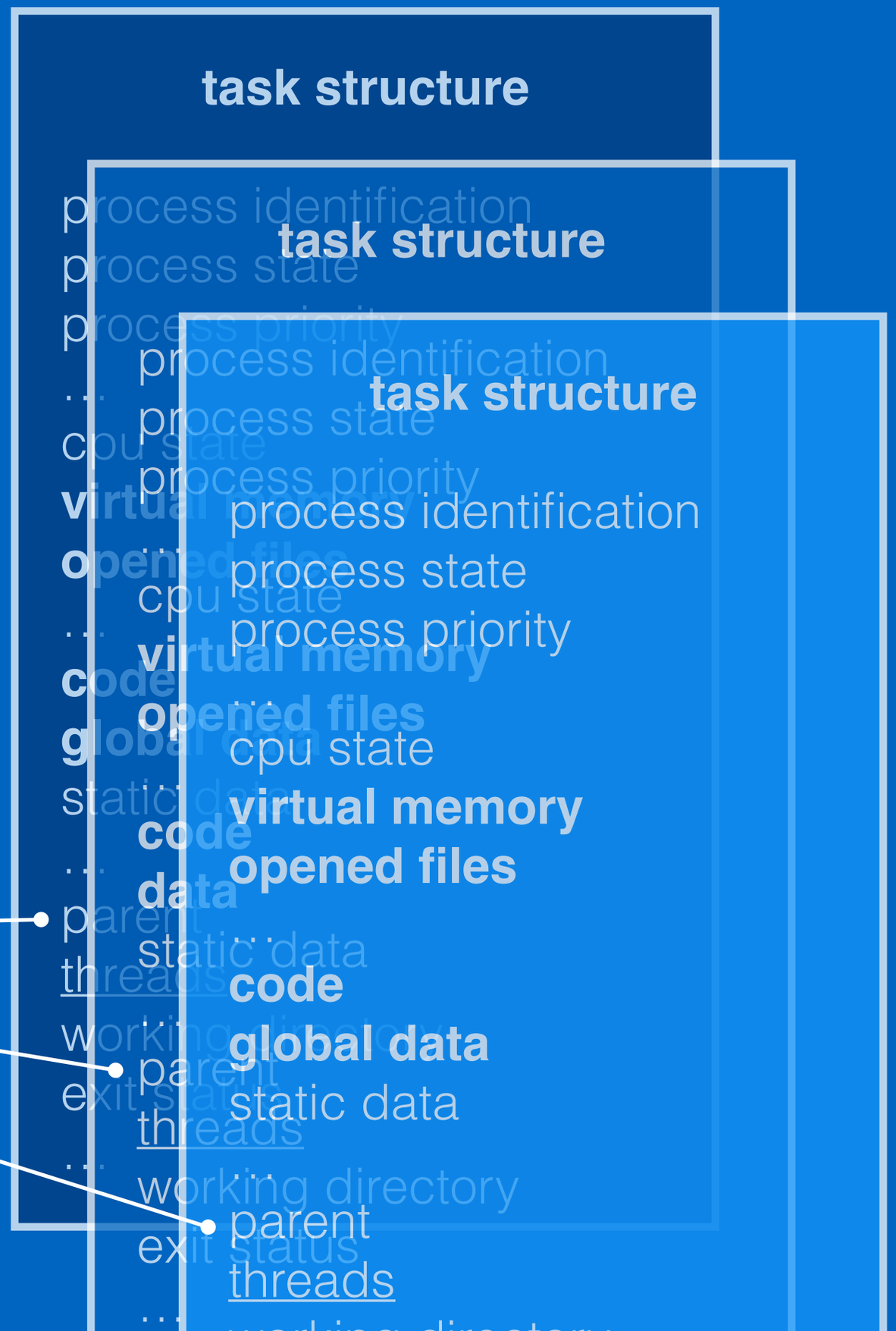
thread identification
thread state
thread priority
…
cpu state
termination status
…
static data
…
termination status
…

same kernel data structure can be used for process descriptors and threads.

such approach simplifies scheduling and context switching logic.

task structure

process identification
process state
process priority
…
cpu state
virtual memory
opened files
…
code
global data
static data
…
parent
working directory
exit status
…



Reading

Operating Systems Design and Implementation,
Third Edition by Andrew S. Tanenbaum

Chapter 2 (2.1, 2.4)

Supplemental Reading

Understanding the Linux kernel, Third Edition by Daniel P. Bovet and Marco Cesati

Chapter 3

Supplemental Reading

Windows Internals, Part 1 (6th Edition) by Mark E. Russinovich and David A. Solomon

Chapter 5

Supplemental Reading

Mac OS X and iOS internals : to the apple's core
by Jonathan Levin

Chapter 4