CS 314 – Operating Systems

Caveat

- Expect to be busy
- Assignments take time
  - Plan accordingly
  - E.g. if 3 weeks are given for an assignment, do not wait until the last couple of days to start
- Keep up
  - Material is not difficult, but can be plentiful
- Know the syllabus
- Interact
  - If you have questions, ask

“Study the past if you would define the future”
-Confucius

- Former 314 students' thoughts about assignments
  - “Start early on the labs and homework”
  - “Get a partner for labs, do them incrementally”
  - “the programs... might make your brain melt”
  - “give yourself enough time to do the labs and ask a lot of questions if you are stuck”
  - “The programs are not as easy as one would hope”
  - “The labs take a lot of time.”
  - “Take the time to understand the assignments, then they will be much easier”

“Sweet is the memory of past troubles”
-Faulkner

- Former 314 students' thoughts about the environment:
  - “Know everything”
  - “get to know working in a UNIX environment early in the semester”
  - “Invest time in learning bash and C before any labs are assigned. If a lab is due in two weeks, work all of those two weeks on it.”

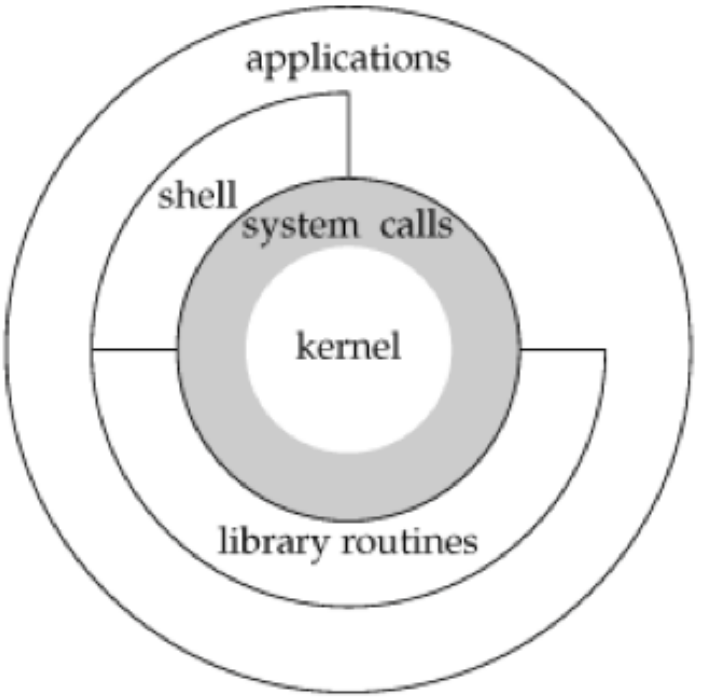
“There is no present or future – only the past, happening over and over again – now”

-Eugene O'Neill

- Former 314 students' thoughts about stuff in general:
  - “Go talk to Professor Crk, no one does and he wants to help”
  - “Go to office hours”
  - “CS 314 is a 2 semester course, they just don't mention that in the catalog description”
  - “The internet is a useful tool, but don't use it as your only tool”
  - “Read the book. Come to class. DON'T PROCRASTINATE”
  - “Don't put off assignments in this class like you might do in others”
  - “Look through the slides to determine what to read.”

The Simple Big Picture

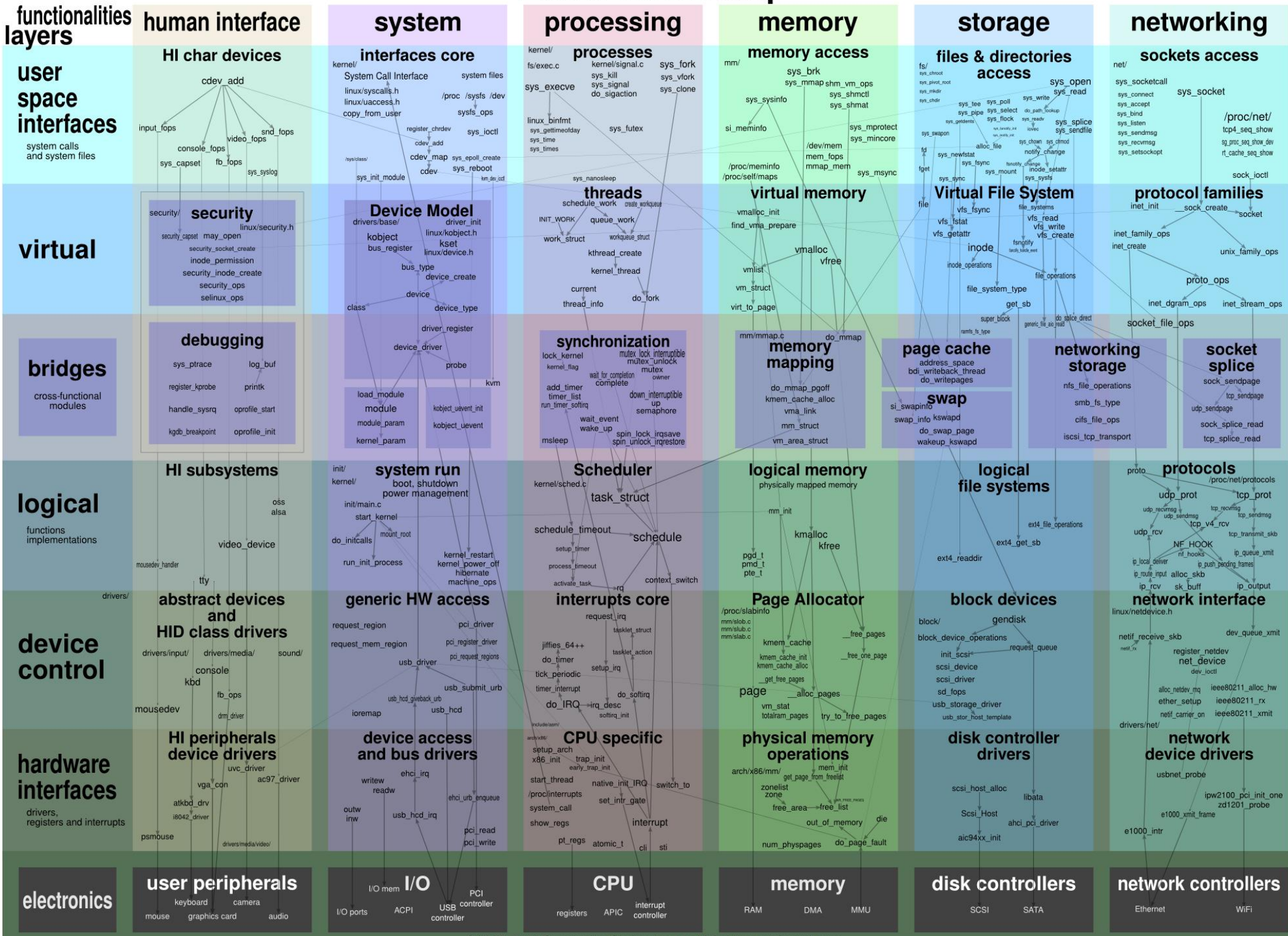
- Kernel: resource management
- System calls: how we ask the kernel action
- Shell/UI: how we interact with the system
- Library routines: how we build functionality around system calls
- Applications: programs whose timely and efficient execution users care about



OS – What are we talking about?

- OS is an extended machine
  - architecture is primitive and/or awkward
  - OS presents a virtual machine that should be easier to program
- OS does resource management
  - How do we provide the means for multiple programs to run?
  - How do we manage and protect memory/devices?
  - How do we multiplex resources in time and space?

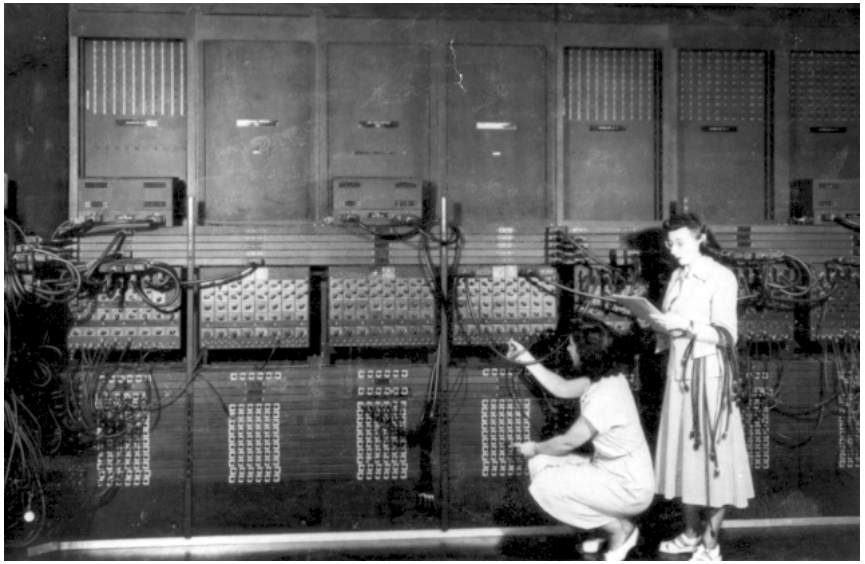
Linux kernel map



© 2007, 2010 Constantine Shulyupin www.MakeLinux.net/kernel_map

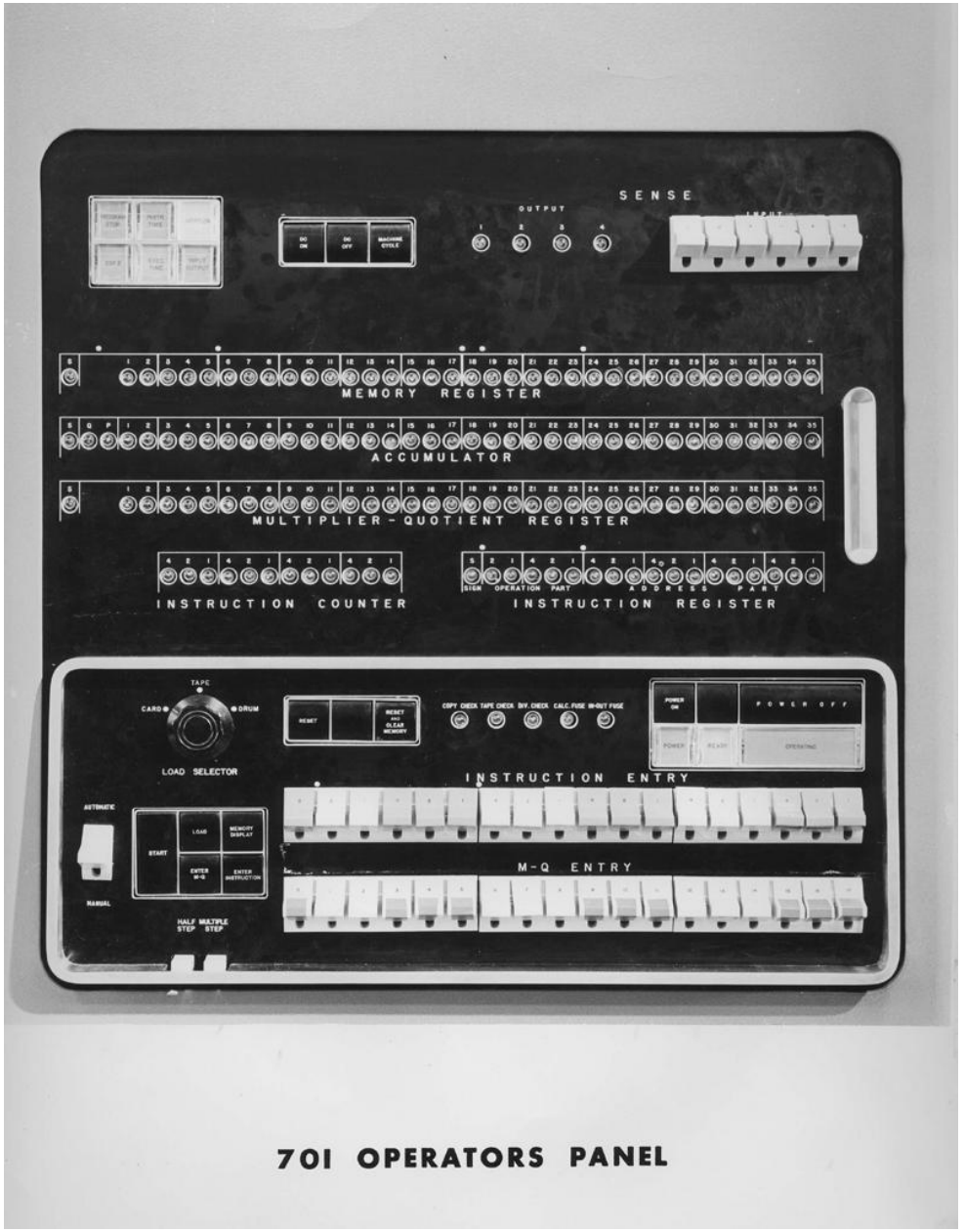
How did we get here? (A general history)

- All advances drive towards greater automation
- 1945-1955 (vacuum tubes)
  - No OS, no programming languages
  - Everything is a manual effort
  - Absolute machine language or directly wiring circuits
  - Programmers reserved time to insert a plugboard
  - Punch cards in 1950s



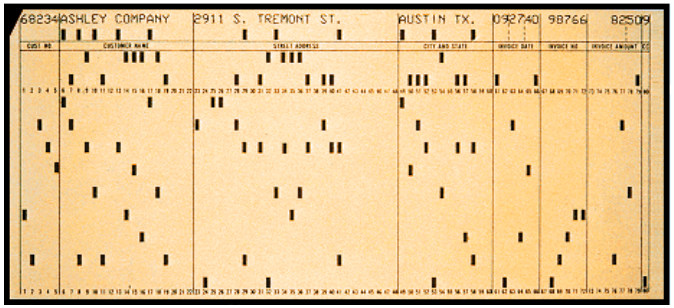
What the 50's looked like (IBM 701)

What the 50's looked like (IBM 701)

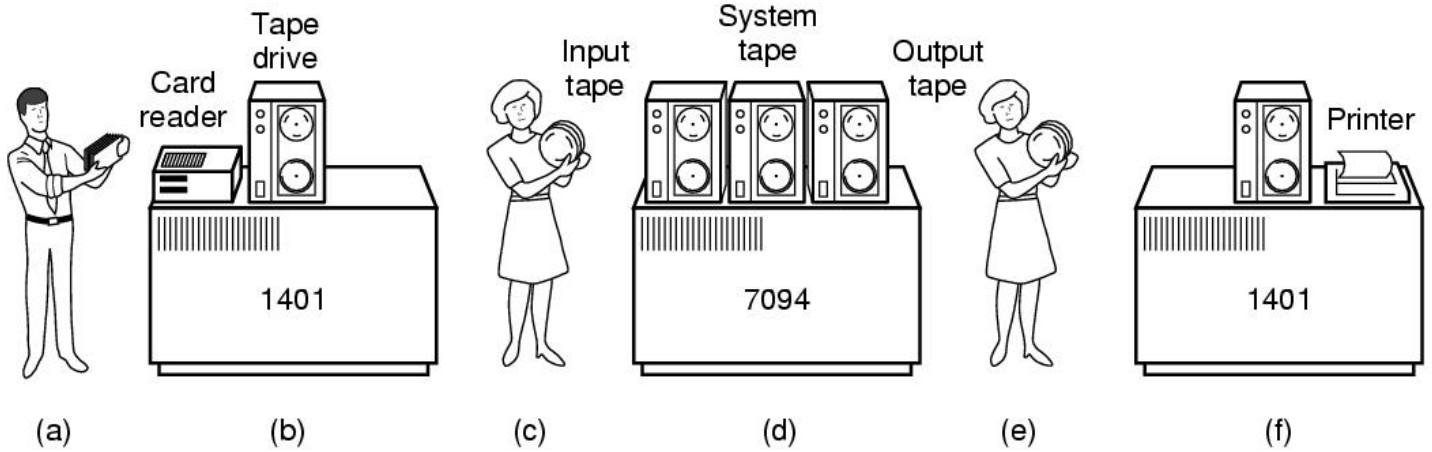


701 OPERATORS PANEL

and so on...



- 1955-1965 (transistors, batch systems)
  - first clear separation between designers, operators, programmers, etc.
  - programmers hand off stacks of punch cards to operators, do something else while waiting for output
  - operators carried tapes containing batches of jobs from card readers to the computing machine and again out to the output printer



What the 60's looked like (IBM 7094)

What the 60's looked like (IBM 7094)

and so on...

- 1965-1980 (Multiprogramming)
  - While a job is waiting for some I/O operation to complete, allow another job to use the CPU
  - Timesharing – allowing many simultaneous users
- 1980-Present
  - Personal computing
  - DOS, Windows, Linux, Android, OSX, etc.
- In short
  - all modern operating systems designs come from early ones
  - all solve essentially the same resource management problems: CPU sharing, memory partitioning/sharing, I/O device sharing, interfacing with users and programs

Linux kernel map

2.6.36

functionalities layers

human interface | system | processing | memory | storage | networking

user space interfaces
system calls and system files

virtual

bridges
cross-functional modules

logical
functions implementations

device control

hardware interfaces
drivers, registers and interrupts

electronics

HI char devices
cdev_add
input_fops
snd_fops
video_fops
console_fops
sys_capset
fb_fops
sys_syslog

interfaces core
kernel/
System Call Interface
system files
linux/syscalls.h
linux/uaccess.h
copy_from_user
/proc /sysfs /dev
sysfs_ops
register_chrdev
sys_ioctl
cdev_add
/sys/class/
cdev_map
sys_epoll_create
cdev
sys_reboot
sys_init_module
kvm_dev_ioctl

processes
kernel/
fs/exec.c
kernel/signal.c
sys_fork
sys_kill
sys_signal
sys_vfork
sys_execve
do_sigaction
sys_clone
linux_binfmt
sys_gettimeofday
sys_time
sys_times
sys_futex
sys_nanosleep

memory access
mm/
sys_brk
sys_mmap
shm_vm_ops
sys_sysinfo
sys_shmctl
sys_shmat
si_meminfo
sys_mprotect
sys_mincore
/dev/mem
mem_fops
mmap_mem
/proc/meminfo
/proc/self/maps
sys_msync

files & directories access
fs/
sys_chroot
sys_pivot_root
sys_mkdir
sys_chdir
sys_open
sys_read
sys_write
sys_tee
sys_poll
sys_pipe
sys_select
sys_flock
sys_splice
sys_sendfile
sys_swapon
alloc_file
notify_change
fd
sys_newfstat
sys_fsync
inode_setattr
fget
sys_mount
sys_sysfs
sys_sync

sockets access
net/
sys_socketcall
sys_connect
sys_accept
sys_bind
sys_listen
sys_sendmsg
sys_recvmsg
sys_setsockopt
sys_socket
/proc/net/
tcp4_seq_show
sg_proc_seq_show_dev
rt_cache_seq_show
sock_ioctl

threads
schedule_work
create_workqueue
INIT_WORK
queue_work
work_struct
workqueue_struct
kthread_create
kernel_thread
current
thread_info
do_fork

virtual memory
vmalloc_init
find_vma_prepare
vmalloc
vfree
vmlist
vm_struct
virt_to_page

Virtual File System
file
vfs_fsync
file_systems
vfs_fstat
vfs_read
vfs_write
vfs_getattr
vfs_create
fsnotify
inode
inode_operations
file_operations
file_system_type
get_sb
super_block
do_splice_direct
generic_file_aio_read
ramfs_fs_type

protocol families
inet_init
__sock_create
socket
inet_family_ops
inet_create
unix_family_ops
proto_ops
inet_dgram_ops
inet_stream_ops
socket_file_ops

security/
security
linux/security.h
security_capset
may_open
security_socket_create
inode_permission
security_inode_create
security_ops
selinux_ops

Device Model
drivers/base/
driver_init
kobject
linux/kobject.h
kset
bus_register
linux/device.h
bus_type
device_create
device
class
device_type
driver_register
device_driver
probe
kvm

synchronization
lock_kernel
mutex_lock_interruptible
mutex_unlock
kernel_flag
wait_for_completion
mutex
owner
complete
add_timer
timer_list
down_interruptible
run_timer_softirq
up
semaphore
wait_event
wake_up
spin_lock_irqsave
msleep
spin_unlock_irqrestore

mm/mmap.c
do_mmap
memory mapping
do_mmap_pgoff
kmem_cache_alloc
vma_link
mm_struct
vm_area_struct

page cache
address_space
bdi_writeback_thread
do_writepages

swap
si_swapinfo
swap_info
kswapd
do_swap_page
wakeup_kswapd

networking storage
nfs_file_operations
smb_fs_type
cifs_file_ops
iscsi_tcp_transport

socket splice
sock_sendpage
tcp_sendpage
udp_sendpage
sock_splice_read
tcp_splice_read

debugging
sys_ptrace
log_buf
register_kprobe
printk
handle_sysrq
oprofile_start
kgdb_breakpoint
oprofile_init

load_module
module
module_param
kernel_param
kobject_uevent_init
kobject_uevent

HI subsystems
oss
alsa
video_device
mousedev_handler
tty

init/
kernel/
system run
boot, shutdown
power management
init/main.c
start_kernel
mount_root
do_initcalls
run_init_process
kernel_restart
kernel_power_off
hibernate
machine_ops

Scheduler
kernel/sched.c
task_struct
schedule_timeout
schedule
setup_timer
process_timeout
activate_task
rq
context_switch

logical memory
physically mapped memory
mm_init
kmalloc
kfree
pgd_t
pmd_t
pte_t

logical file systems
ext4_file_operations
ext4_get_sb
ext4_readdir

proto
protocols
/proc/net/protocols
udp_prot
tcp_prot
udp_recvmsg
tcp_recvmsg
udp_sendmsg
tcp_sendmsg
tcp_v4_rcv
udp_rcv
tcp_transmit_skb
NF_HOOK
nf_hooks
ip_queue_xmit
ip_local_deliver
ip_push_pending_frames
ip_route_input
alloc_skb
ip_rcv
sk_buff
ip_output

drivers/
abstract devices and HID class drivers
drivers/input/
drivers/media/
sound/
console
kbd
fb_ops
mousedev
drm_driver

generic HW access
request_region
pci_driver
request_mem_region
pci_register_driver
pci_request_regions
usb_driver
usb_submit_urb
usb_hcd_giveback_urb
ioremap
usb_hcd

interrupts core
request_irq
tasklet_struct
jiffies_64++
tasklet_action
do_timer
setup_irq
tick_periodic
timer_interrupt
do_softirq
do_IRQ
irq_desc
softirq_init

Page Allocator
/proc/slabinfo
mm/slob.c
mm/slub.c
mm/slab.c
__free_pages
kmem_cache
__free_one_page
kmem_cache_init
kmem_cache_alloc
__get_free_pages
page
__alloc_pages
vm_stat
totalram_pages
try_to_free_pages

block devices
block/
gendisk
block_device_operations
request_queue
init_scsi
scsi_device
scsi_driver
sd_fops
usb_storage_driver
usb_stor_host_template

network interface
linux/netdevice.h
dev_queue_xmit
netif_receive_skb
netif_rx
register_netdev
net_device
dev_ioctl
alloc_netdev_mq
ieee80211_alloc_hw
ether_setup
ieee80211_rx
netif_carrier_on
ieee80211_xmit

HI peripherals device drivers
uvc_driver
vga_con
ac97_driver
atkbd_drv
i8042_driver
psmouse
drivers/media/video/

device access and bus drivers
writew
readw
ehci_irq
ehci_urb_enqueue
outw
inw
usb_hcd_irq
pci_read
pci_write

CPU specific
include/asm/
arch/x86/
setup_arch
x86_init
trap_init
early_trap_init
start_thread
native_init_IRQ
switch_to
/proc/interrupts
set_intr_gate
system_call
show_regs
interrupt
pt_regs
atomic_t
cli
sti

physical memory operations
arch/x86/mm/
mem_init
get_page_from_freelist
zonelist
zone
free_area
free_list
out_of_memory
die
num_physpages
do_page_fault

disk controller drivers
scsi_host_alloc
libata
Scsi_Host
ahci_pci_driver
aic94xx_init

drivers/net/
network device drivers
usbnet_probe
ipw2100_pci_init_one
zd1201_probe
e1000_xmit_frame
e1000_intr

user peripherals
keyboard
camera
mouse
graphics card
audio

I/O
I/O mem
PCI controller
I/O ports
ACPI
USB controller

CPU
registers
APIC
interrupt controller

memory
RAM
DMA
MMU

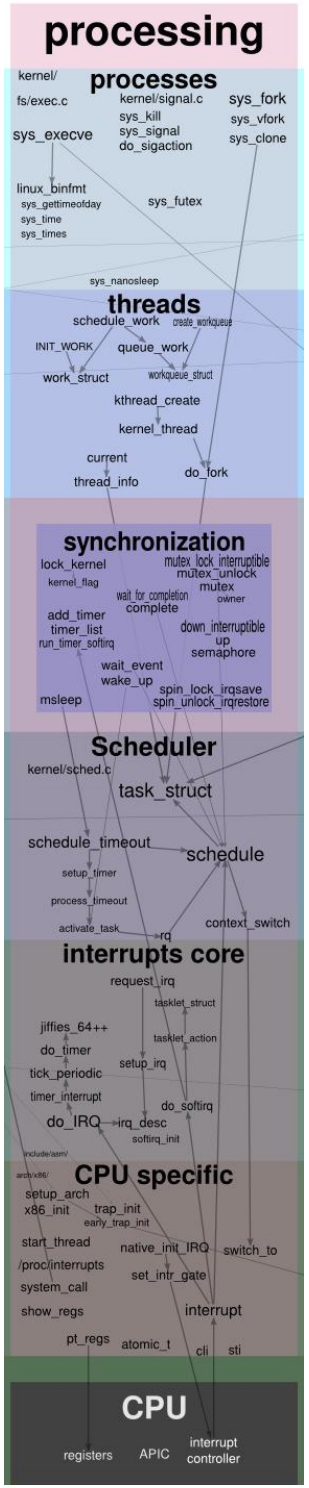
disk controllers
SCSI
SATA

network controllers
Ethernet
WiFi

© 2007, 2010 Constantine Shulyupin www.MakeLinux.net/kernel_map

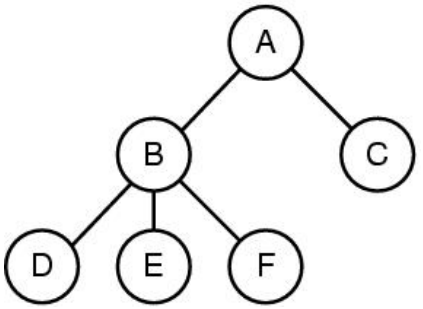
CPU Review

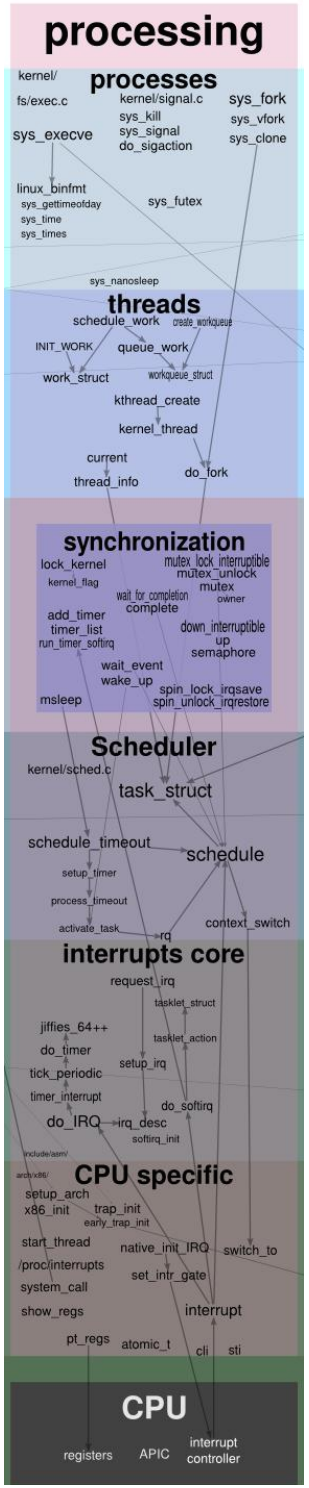
- Fetch/execute instructions
  - has unique instruction set
- General and special registers
  - Program counter (PC): keep track of current inst.
  - Stack pointer (SP): currently executing procedure
  - Program status word (PSW): what is the program allowed to do
    - user/kernel mode, CPU priority, etc...



Processes in General

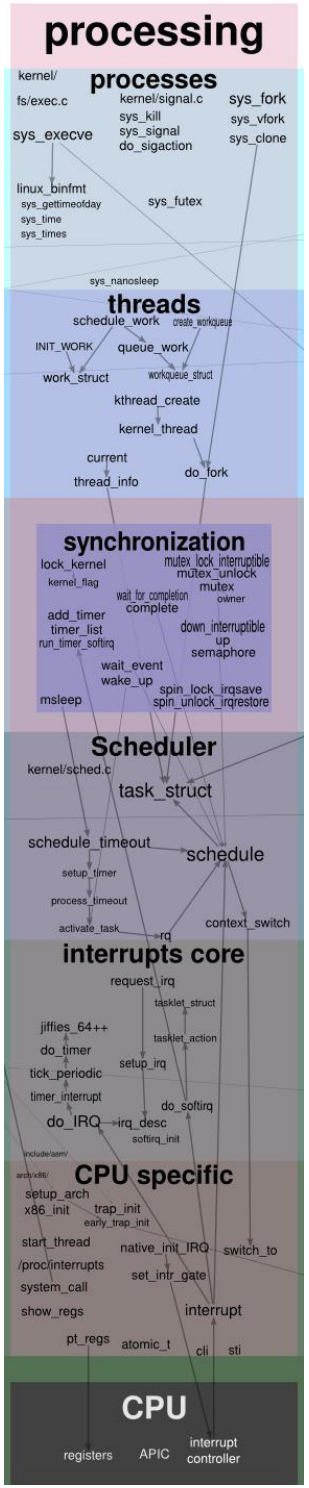
- Processes are programs in execution
- From the OS perspective
  - process = container defining an address space which contains:
    - executable code
    - program data
    - program stack
- OS manages processes
- Processes can reproduce
- Processes can communicate





Processes in General

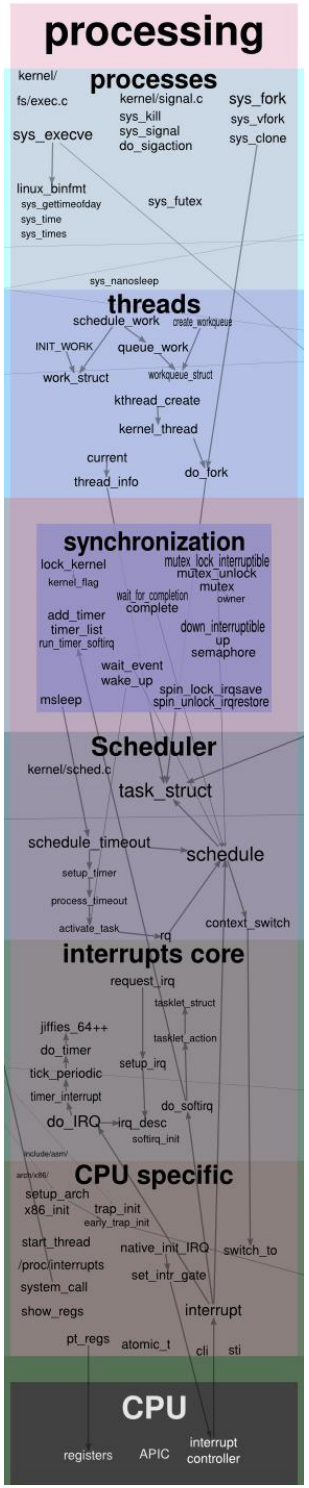
- Multiuser systems allow several processes to be active concurrently
- Systems allowing concurrent active processes are multiprogramming systems
- scheduler decides which process is allowed to progress
  - OS may be nonpreemptive
    - scheduler invoked only when CPU is voluntarily relinquished
  - or preemptive
    - scheduler invoked by OS periodically, a necessity for multiuser systems



Processes in General

- (in UNIX) Processes are under the illusion that they are the only process on the machine with exclusive access to the kernel
  - kernel itself is not a process, but a process manager
- Threads allow one process to have multiple PCs and SPs, or multiple processes to work within the same address space...
- Creating a new process in Unix is simple

```
int pid = fork();
if (pid == 0) {
    exec("foo");
} else {
    waitpid(pid, &status, options);
};
```



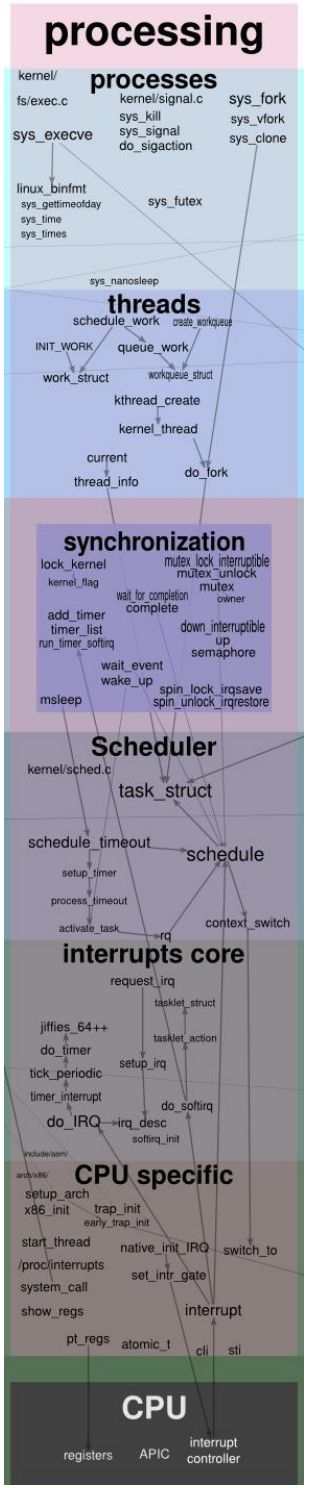
Processes in General

- In windows, it's arguably not:

```
BOOL CreateProcess(
    LPCTSTR lpApplicationName,
    LPTSTR lpCommandLine,
    LPSECURITY_ATTRIBUTES lpProcessAttributes,
    LPSECURITY_ATTRIBUTES lpThreadAttributes,
    BOOL bInheritHandles,
    DWORD dwCreationFlags,
    LPVOID lpEnvironment,
    LPCTSTR lpCurrentDirectory,
    LPSTARTUPINFO lpStartupInfo,
    LPPROCESS_INFORMATION lpProcessInformation
);

WaitForSingleObject(lpProcessInformation->hProcess,
    INFINITE);
```

- So... we'll stick with Unix-like systems
  - Note: this reason alone would be insufficient, but consider it illustrative of the situation



Linux kernel map

2.6.36

functionalities layers

human interface | system | processing | memory | storage | networking

user space interfaces
system calls and system files

virtual
bridges
cross-functional modules

logical
functions implementations

device control
hardware interfaces
drivers, registers and interrupts

electronics

HI char devices
cdev_add
input_fops
snd_fops
video_fops
console_fops
sys_capset
fb_fops
sys_syslog

interfaces core
kernel/
System Call Interface
system files
linux/syscalls.h
linux/uaccess.h
/proc /sysfs /dev
copy_from_user
sysfs_ops
register_chrdev
sys_ioctl
cdev_add
/sys/class/
cdev_map
sys_epoll_create
cdev
sys_reboot
sys_init_module
kvm_dev_ioctl

processes
kernel/
fs/exec.c
kernel/signal.c
sys_fork
sys_kill
sys_vfork
sys_execve
sys_signal
sys_clone
do_sigaction
linux_binfmt
sys_gettimeofday
sys_futex
sys_time
sys_times
sys_nanosleep

memory access
mm/
sys_brk
sys_mmap
shm_vm_ops
sys_sysinfo
sys_shmctl
sys_shmat
si_meminfo
sys_mprotect
sys_mincore
/dev/mem
mem_fops
mmap_mem
/proc/meminfo
/proc/self/maps
sys_msync

files & directories access
fs/
sys_chroot
sys_pivot_root
sys_mkdir
sys_chdir
sys_open
sys_read
sys_write
sys_tee
sys_poll
sys_pipe
sys_select
do_path_lookup
sys_getdents
sys_flock
sys_readv
sys_splice
sys_swapon
sys_sendfile
fd
alloc_file
sys_chown
sys_chmod
sys_newfstat
notify_change
fget
sys_fsync
sys_mount
inode_setattr
sys_sync
sys_sysfs

sockets access
net/
sys_socketcall
sys_connect
sys_accept
sys_bind
sys_listen
sys_sendmsg
sys_recvmsg
sys_setsockopt
sys_socket
/proc/net/
tcp4_seq_show
sg_proc_seq_show_dev
rt_cache_seq_show
sock_ioctl

security/
security
linux/security.h
security_capset
may_open
security_socket_create
inode_permission
security_inode_create
security_ops
selinux_ops

Device Model
drivers/base/
driver_init
kobject
linux/kobject.h
bus_register
kset
linux/device.h
bus_type
device_create
device
class
device_type
driver_register
device_driver
probe
kvm
load_module
module
module_param
kernel_param
kobject_uevent_init
kobject_uevent

threads
schedule_work
create_workqueue
INIT_WORK
queue_work
work_struct
workqueue_struct
kthread_create
kernel_thread
current
thread_info
do_fork

virtual memory
vmalloc_init
find_vma_prepare
vmalloc
vfree
vmlist
vm_struct
virt_to_page

Virtual File System
file
vfs_fsync
file_systems
vfs_fstat
vfs_read
vfs_write
vfs_getattr
vfs_create
fsnotify
inode
inode_operations
file_operations
file_system_type
get_sb
super_block
do_splice_direct
generic_file_aio_read
ramfs_fs_type

protocol families
inet_init
__sock_create
socket
inet_family_ops
inet_create
unix_family_ops
proto_ops
inet_dgram_ops
inet_stream_ops
socket_file_ops

debugging
sys_ptrace
log_buf
register_kprobe
printk
handle_sysrq
oprofile_start
kgdb_breakpoint
oprofile_init

synchronization
lock_kernel
mutex_lock_interruptible
mutex_unlock
kernel_flag
wait_for_completion
mutex
owner
complete
add_timer
timer_list
run_timer_softirq
down_interruptible
up
semaphore
wait_event
wake_up
spin_lock_irqsave
msleep
spin_unlock_irqrestore

mm/mmap.c
do_mmap
memory mapping
do_mmap_pgoff
kmem_cache_alloc
vma_link
mm_struct
vm_area_struct

page cache
address_space
bdi_writeback_thread
do_writepages

swap
si_swapinfo
swap_info
kswapd
do_swap_page
wakeup_kswapd

networking storage
nfs_file_operations
smb_fs_type
cifs_file_ops
iscsi_tcp_transport

socket splice
sock_sendpage
tcp_sendpage
udp_sendpage
sock_splice_read
tcp_splice_read

HI subsystems
oss
alsa
video_device
mousedev_handler
tty

init/
kernel/
system run
boot, shutdown
power management
init/main.c
start_kernel
mount_root
do_initcalls
run_init_process
kernel_restart
kernel_power_off
hibernate
machine_ops

Scheduler
kernel/sched.c
task_struct
schedule_timeout
schedule
setup_timer
process_timeout
activate_task
rq
context_switch

logical memory
physically mapped memory
mm_init
kmalloc
kfree
pgd_t
pmd_t
pte_t

logical file systems
ext4_file_operations
ext4_get_sb
ext4_readdir

protocols
proto
/proc/net/protocols
udp_prot
tcp_prot
udp_recvmsg
tcp_recvmsg
udp_sendmsg
tcp_sendmsg
tcp_v4_rcv
udp_rcv
tcp_transmit_skb
NF_HOOK
nf_hooks
ip_queue_xmit
ip_local_deliver
ip_push_pending_frames
ip_route_input
alloc_skb
ip_rcv
sk_buff
ip_output

drivers/
abstract devices and HID class drivers
drivers/input/
drivers/media/
sound/
console
kbd
fb_ops
mousedev
drm_driver

generic HW access
request_region
pci_driver
request_mem_region
pci_register_driver
pci_request_regions
usb_driver
usb_submit_urb
usb_hcd_giveback_urb
ioremap
usb_hcd

interrupts core
request_irq
tasklet_struct
jiffies_64++
tasklet_action
do_timer
setup_irq
tick_periodic
timer_interrupt
do_softirq
do_IRQ
irq_desc
softirq_init

Page Allocator
/proc/slabinfo
mm/slob.c
mm/slub.c
mm/slab.c
kmem_cache
__free_pages
kmem_cache_init
__free_one_page
kmem_cache_alloc
__get_free_pages
page
__alloc_pages
vm_stat
totalram_pages
try_to_free_pages

block devices
block/
gendisk
block_device_operations
request_queue
init_scsi
scsi_device
scsi_driver
sd_fops
usb_storage_driver
usb_stor_host_template

network interface
linux/netdevice.h
dev_queue_xmit
netif_receive_skb
netif_rx
register_netdev
net_device
dev_ioctl
alloc_netdev_mq
ieee80211_alloc_hw
ether_setup
ieee80211_rx
netif_carrier_on
ieee80211_xmit

HI peripherals device drivers
uvc_driver
vga_con
ac97_driver
atkbd_drv
i8042_driver
psmouse
drivers/media/video/

device access and bus drivers
writew
readw
ehci_irq
ehci_urb_enqueue
outw
inw
usb_hcd_irq
pci_read
pci_write

CPU specific
include/asm/
arch/x86/
setup_arch
x86_init
trap_init
early_trap_init
start_thread
native_init_IRQ
switch_to
/proc/interrupts
set_intr_gate
system_call
show_regs
interrupt
pt_regs
atomic_t
cli
sti

physical memory operations
arch/x86/mm/
mem_init
get_page_from_freelist
zonelist
zone
free_area
free_list
out_of_memory
die
num_physpages
do_page_fault

disk controller drivers
scsi_host_alloc
libata
Scsi_Host
ahci_pci_driver
aic94xx_init

drivers/net/
network device drivers
usbnet_probe
ipw2100_pci_init_one
zd1201_probe
e1000_xmit_frame
e1000_intr

user peripherals
keyboard
camera
mouse
graphics card
audio

I/O
I/O mem
PCI controller
I/O ports
ACPI
USB controller

CPU
registers
APIC
interrupt controller

memory
RAM
DMA
MMU

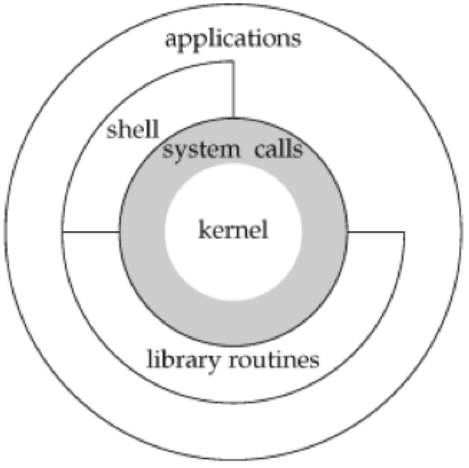
disk controllers
SCSI
SATA

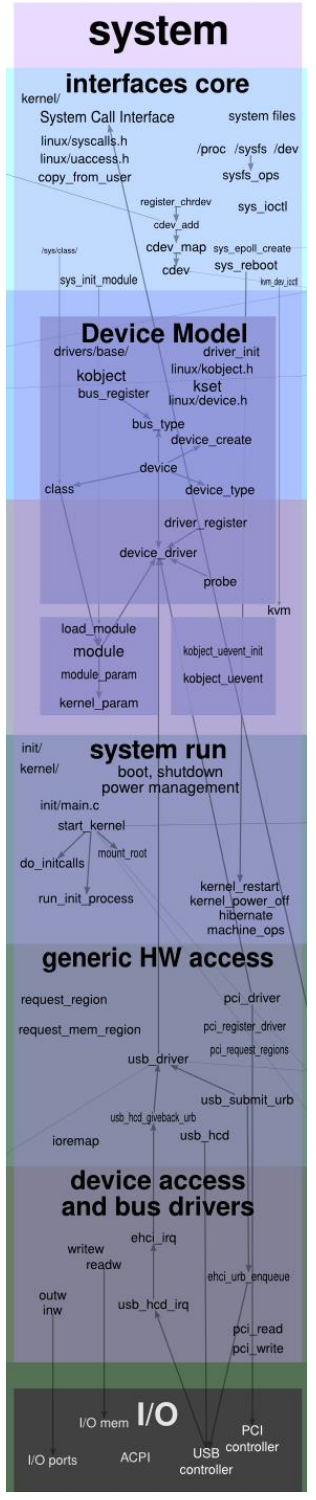
network controllers
Ethernet
WiFi

© 2007-2010 Constantine Shulyupin www.MakeLinux.net/kernel_map

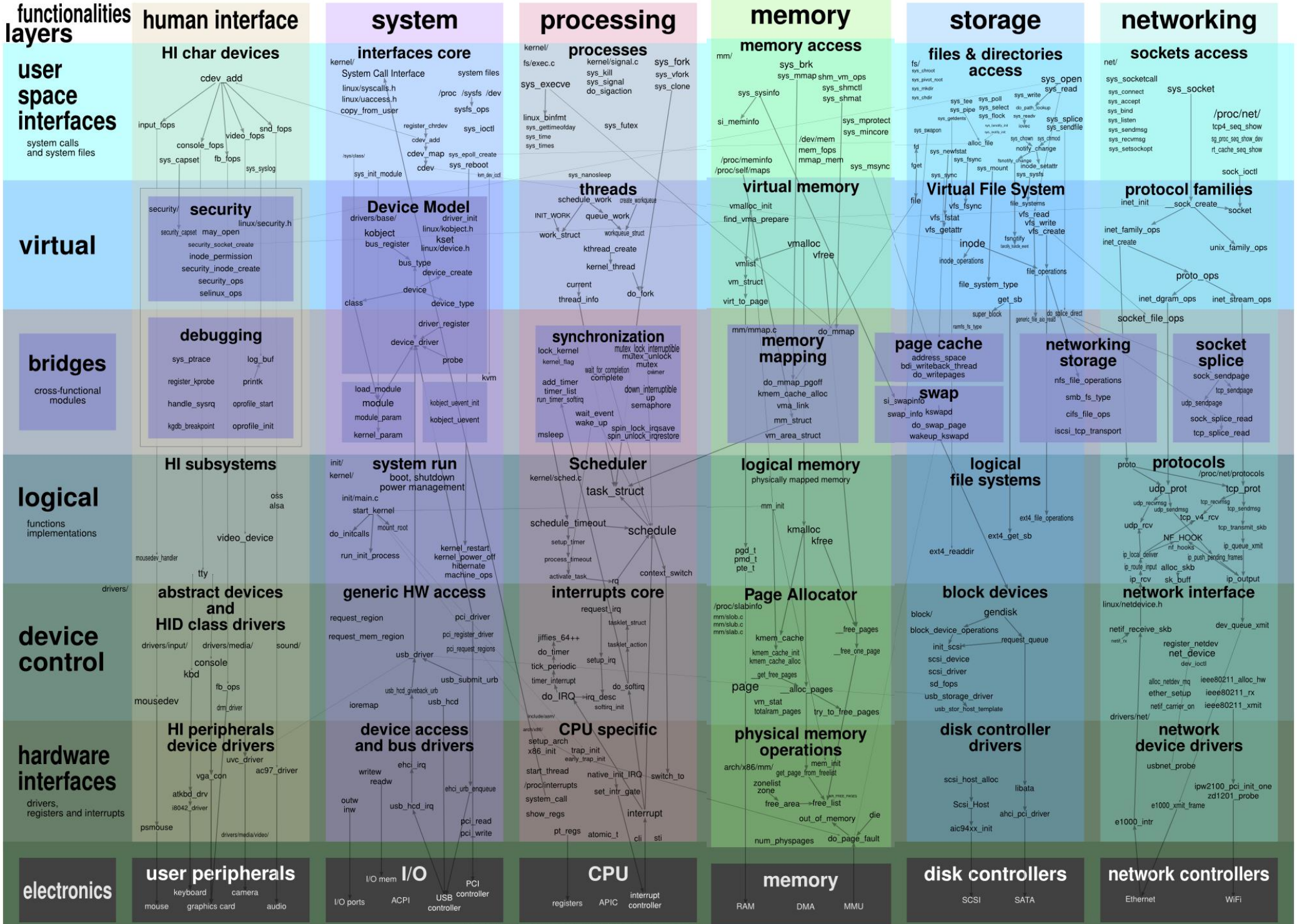
System Calls

- Processes make system calls to request OS services
  - privilege mode is changed from user mode to kernel mode
  - the intended kernel procedure is executed
  - on completion, kernel procedure returns HW to user mode and process continues where it left off





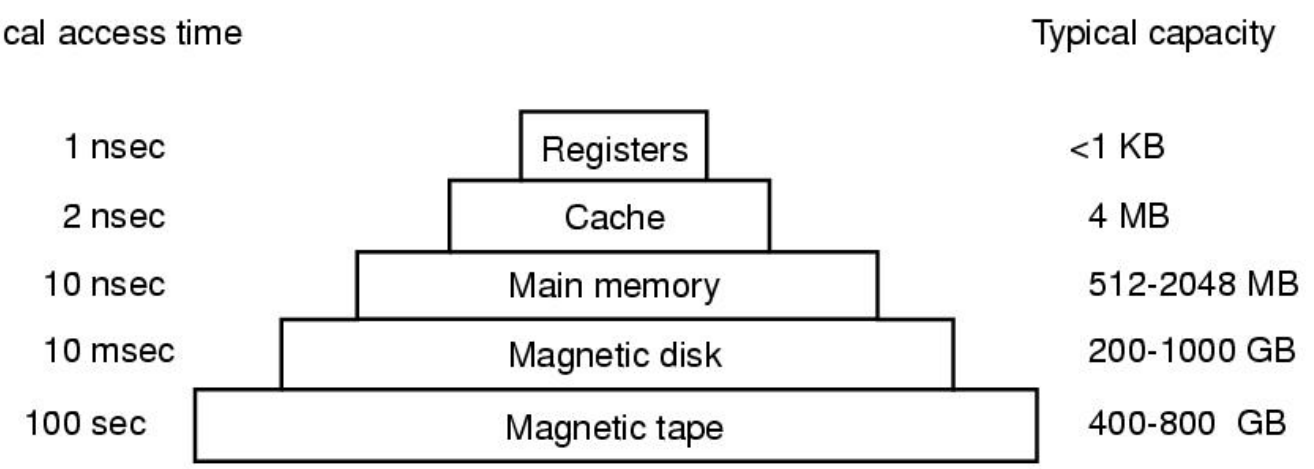
Linux kernel map

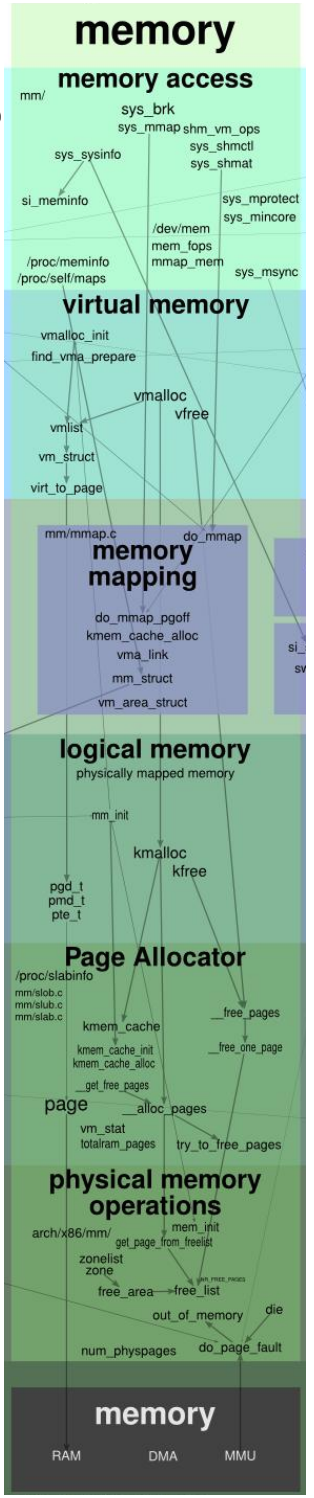


© 2007, 2010 Constantine Shulyupin www.MakeLinux.net/Kernel_map

Remember the Memory Hierarchy?

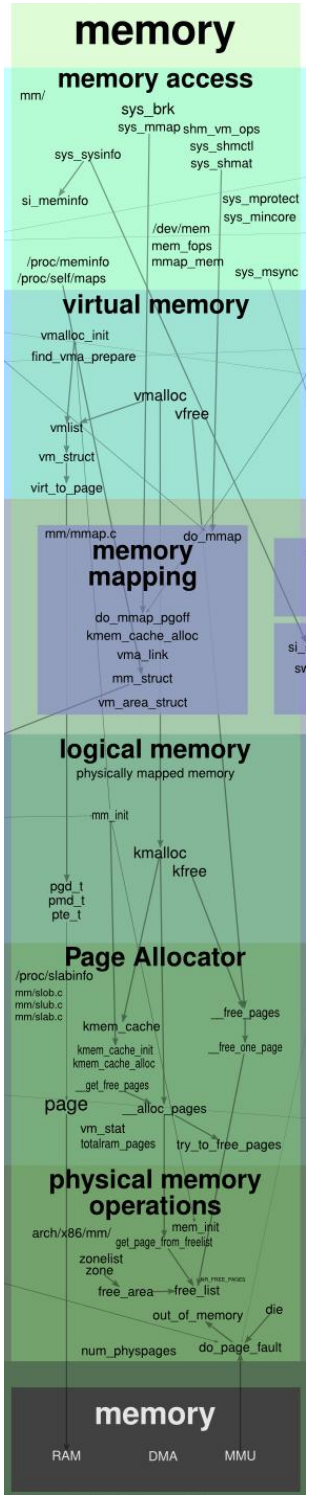
- Capacities and access times are approximate





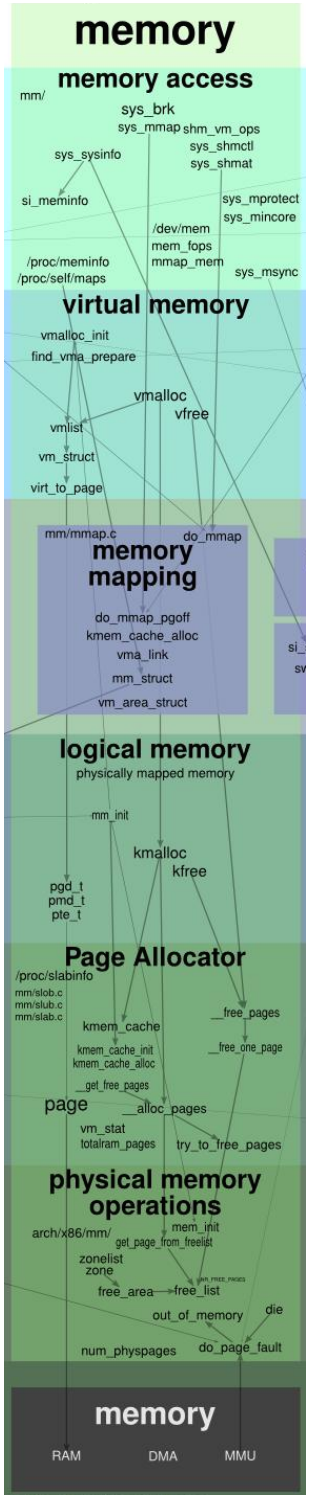
So what?

- Processes must have memory to play with
- Physical memory must be shared
  - processes must be protected from each other
  - the system must be protected from processes
  - processes must be protected from themselves?
- Process address spaces can be significantly larger than what is physically available!
  - what now?

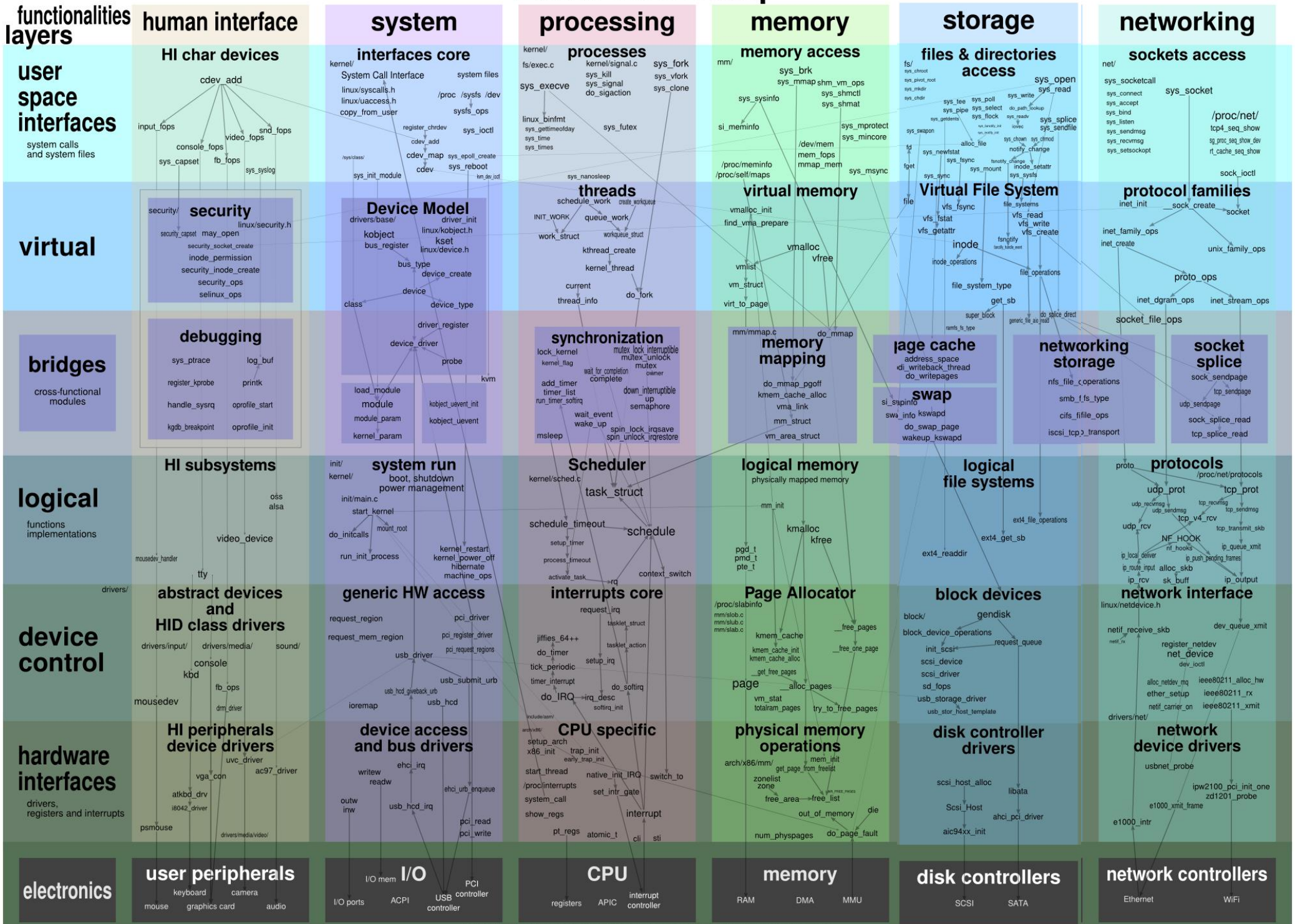


Cache, cache, cache...

- Registers are fast, expensive
- Memory access is slow (by comparison)
- Caching is critical
  - if the cache contains the needed word: 2 cycles
  - if not: many many cycles
- So?
  - When to cache?
  - What should be cached?
  - How is stuff evicted from the cache?
  - Where does evicted stuff go?
  - How can we make this efficient/fair/safe?



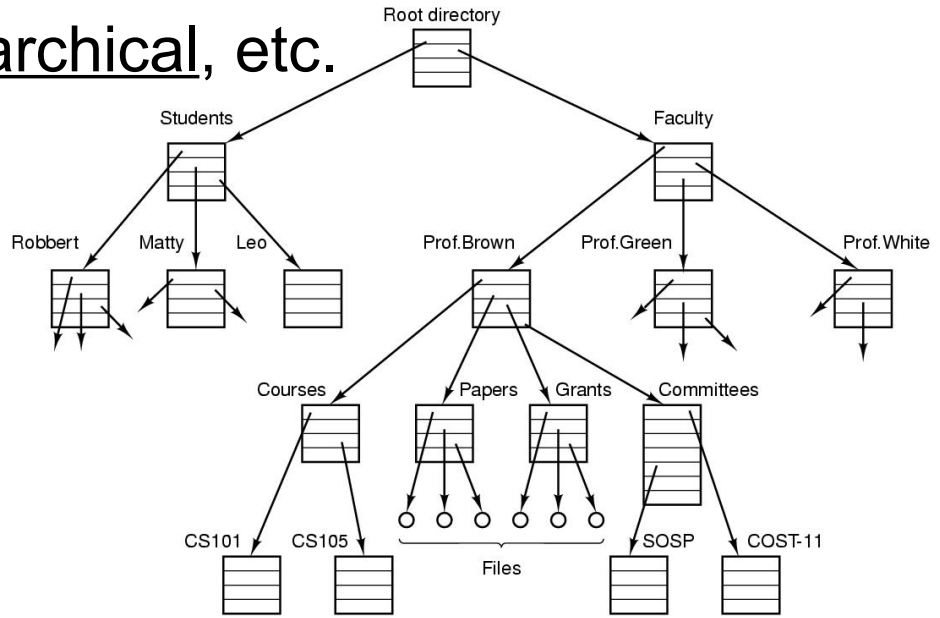
Linux kernel map

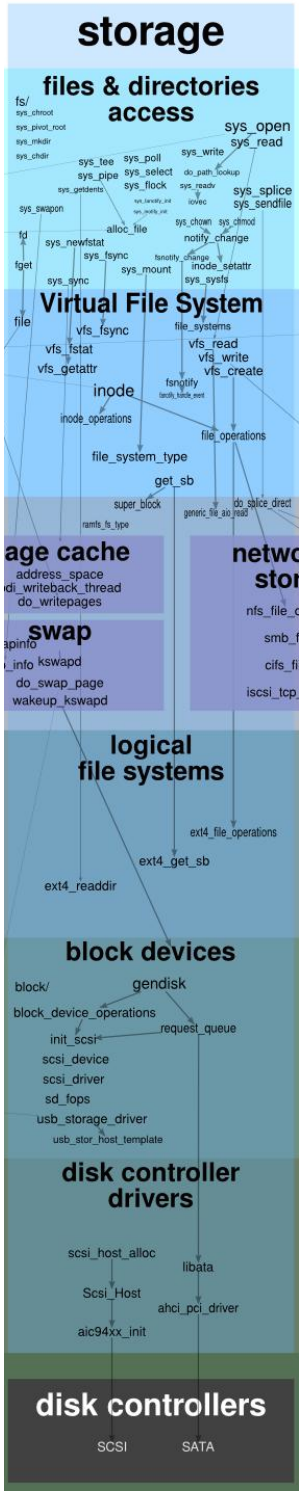


© 2007, 2010 Constantine Shulyupin www.MakeLinux.net/kernel_map

File Systems

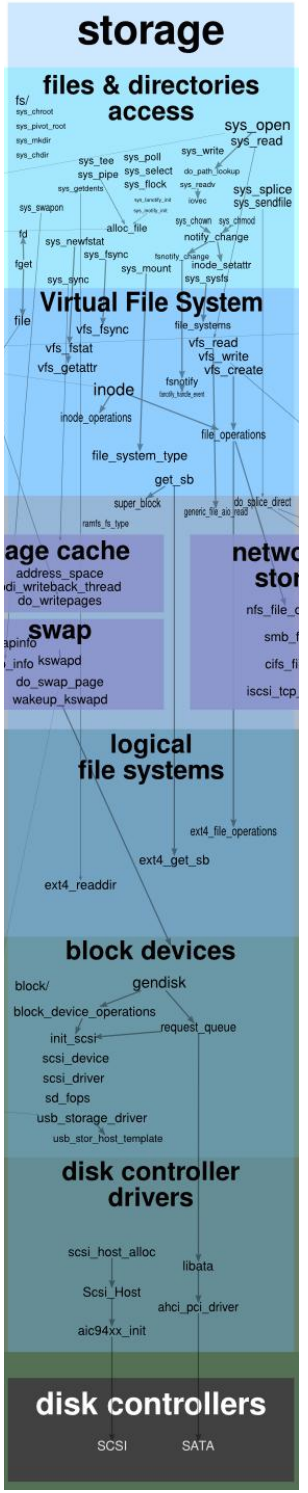
- How is data organized?
- Files, directories, how to define these in the OS, and how to interact with them?
  - associated system calls, structures, etc.
- Organizations
  - flat, hierarchical, etc.





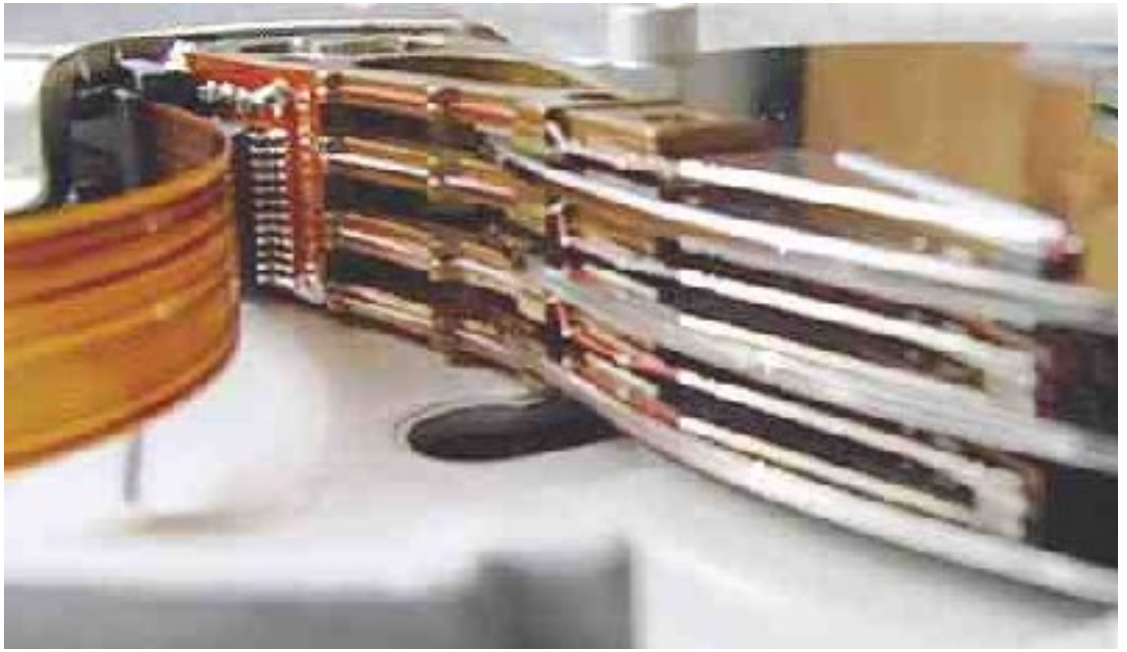
File Systems

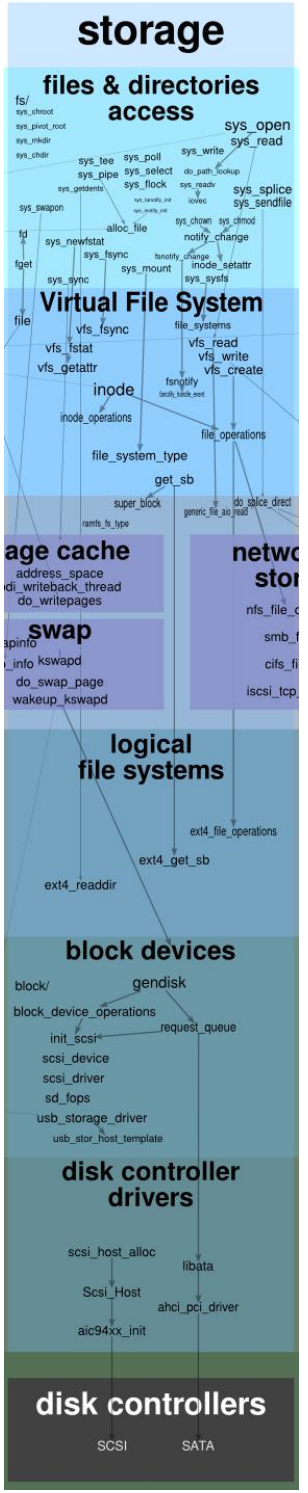
- Representation of files/directories in the OS
  - inodes
    - design choices directly affect performance
- System call interfaces for working with files
- Fun with hierarchical file systems
  - hard/soft links, mounting, pipes, etc.



Disks

- Data organization, optimization, performance
  - take advantage of the physical properties
- Effectively extend physical memory
- Need caches to hide their awful slowness





Linux kernel map

2.6.36

functionalities / layers

human interface | system | processing | memory | storage | networking

user space interfaces

system calls and system files

HI char devices

cdev_add
input_fops
snd_fops
video_fops
console_fops
sys_capset
fb_fops
sys_syslog

interfaces core

kernel/
System Call Interface
system files
linux/syscalls.h
linux/uaccess.h
copy_from_user
/proc /sysfs /dev
sysfs_ops
register_chrdev
sys_ioctl
cdev_add
/sys/class/
cdev_map
sys_epoll_create
cdev
sys_reboot
sys_init_module
kvm_dev_ioctl

processes

kernel/
fs/exec.c
kernel/signal.c
sys_fork
sys_kill
sys_vfork
sys_execve
sys_signal
sys_clone
do_sigaction
linux_binfmt
sys_gettimeofday
sys_futex
sys_time
sys_times
sys_nanosleep

memory access

mm/
sys_brk
sys_mmap
shm_vm_ops
sys_sysinfo
sys_shmctl
sys_shmat
si_meminfo
sys_mprotect
sys_mincore
/dev/mem
mem_fops
mmap_mem
/proc/meminfo
/proc/self/maps
sys_msync

files & directories access

fs/
sys_chroot
sys_pivot_root
sys_mkdir
sys_chdir
sys_open
sys_read
sys_write
sys_tee
sys_poll
sys_pipe
sys_select
do_path_lookup
sys_getdents
sys_flock
sys_readv
sys_splice
sys_sendfile
sys_swapon
alloc_file
sys_chown
sys_chmod
fd
sys_newfstat
notify_change
sys_fsync
fsnotify_change
fget
sys_mount
inode_setattr
sys_sync
sys_sysfs

sockets access

net/
sys_socketcall
sys_connect
sys_accept
sys_bind
sys_listen
sys_sendmsg
sys_recvmsg
sys_setsockopt
sys_socket
/proc/net/
tcp4_seq_show
sg_proc_seq_show_dev
rt_cache_seq_show
sock_ioctl

virtual

security

security/
security
linux/security.h
security_capset
may_open
security_socket_create
inode_permission
security_inode_create
security_ops
selinux_ops

Device Model

drivers/base/
driver_init
kobject
linux/kobject.h
kset
bus_register
linux/device.h
bus_type
device_create
device
class
device_type
driver_register
device_driver
probe
kvm

threads

schedule_work
create_workqueue
INIT_WORK
queue_work
work_struct
workqueue_struct
kthread_create
kernel_thread
current
thread_info
do_fork

virtual memory

vmalloc_init
find_vma_prepare
vmalloc
vfree
vmlist
vm_struct
virt_to_page

Virtual File System

file
vfs_fsync
file_systems
vfs_fstat
vfs_read
vfs_getattr
vfs_write
vfs_create
fsnotify
inode
inode_operations
file_operations
file_system_type
get_sb
super_block
do_splice_direct
generic_file_aio_read
ramfs_fs_type

protocol families

inet_init
__sock_create
socket
inet_family_ops
inet_create
unix_family_ops
proto_ops
inet_dgram_ops
inet_stream_ops
socket_file_ops

bridges

cross-functional modules

debugging

sys_ptrace
log_buf
register_kprobe
printk
handle_sysrq
oprofile_start
kgdb_breakpoint
oprofile_init

module

load_module
module
module_param
kernel_param
kobject_uevent_init
kobject_uevent

synchronization

lock_kernel
mutex_lock_interruptible
kernel_flag
mutex_unlock
wait_for_completion
mutex
complete
owner
add_timer
timer_list
down_interruptible
run_timer_softirq
up
semaphore
wait_event
wake_up
spin_lock_irqsave
msleep
spin_unlock_irqrestore

memory mapping

mm/mmap.c
do_mmap
do_mmap_pgoff
kmem_cache_alloc
vma_link
mm_struct
vm_area_struct

page cache

address_space
bdi_writeback_thread
do_writepages

swap

si_swapinfo
swap_info
kswapd
do_swap_page
wakeup_kswapd

network storage

nfs_file_operations
smb_fs_type
cifs_file_ops
iscsi_tcp_transport

socket splice

sock_sendpage
tcp_sendpage
udp_sendpage
sock_splice_read
tcp_splice_read

logical

functions implementations

HI subsystems

oss
alsa
video_device
mousedev_handler
tty

system run

init/
kernel/
boot, shutdown
power management
init/main.c
start_kernel
mount_root
do_initcalls
run_init_process
kernel_restart
kernel_power_off
hibernate
machine_ops

Scheduler

kernel/sched.c
task_struct
schedule_timeout
schedule
setup_timer
process_timeout
activate_task
rq
context_switch

logical memory

physically mapped memory
mm_init
kmalloc
kfree
pgd_t
pmd_t
pte_t

logical file systems

ext4_file_operations
ext4_get_sb
ext4_readdir

protocols

proto
/proc/net/protocols
udp_prot
tcp_prot
udp_recvmsg
udp_sendmsg
tcp_recvmsg
tcp_sendmsg
tcp_v4_rcv
udp_rcv
tcp_transmit_skb
NF_HOOK
nf_hooks
ip_queue_xmit
ip_local_deliver
ip_push_pending_frames
ip_route_input
alloc_skb
ip_rcv
sk_buff
ip_output

device control

drivers/

abstract devices and HID class drivers

drivers/input/
drivers/media/
sound/
console
kbd
fb_ops
mousedev
drm_driver

generic HW access

request_region
pci_driver
request_mem_region
pci_register_driver
pci_request_regions
usb_driver
usb_submit_urb
usb_hcd_giveback_urb
ioremap
usb_hcd

interrupts core

request_irq
tasklet_struct
jiffies_64++
tasklet_action
do_timer
setup_irq
tick_periodic
timer_interrupt
do_softirq
do_IRQ
irq_desc
softirq_init

Page Allocator

/proc/slabinfo
mm/slob.c
mm/slub.c
mm/slab.c
kmem_cache
__free_pages
kmem_cache_init
kmem_cache_alloc
__free_one_page
__get_free_pages
page
__alloc_pages
vm_stat
totalram_pages
try_to_free_pages

block devices

block/
gendisk
block_device_operations
request_queue
init_scsi
scsi_device
scsi_driver
sd_fops
usb_storage_driver
usb_stor_host_template

network interface

linux/netdevice.h
netif_receive_skb
dev_queue_xmit
netif_rx
register_netdev
net_device
dev_ioctl
alloc_netdev_mq
ieee80211_alloc_hw
ether_setup
ieee80211_rx
netif_carrier_on
ieee80211_xmit

hardware interfaces

drivers, registers and interrupts

HI peripherals device drivers

uvc_driver
vga_con
ac97_driver
atkbd_drv
i8042_driver
psmouse
drivers/media/video/

device access and bus drivers

writew
ehci_irq
readw
ehci_urb_enqueue
outw
inw
usb_hcd_irq
pci_read
pci_write

CPU specific

include/asm/
arch/x86/
setup_arch
x86_init
trap_init
early_trap_init
start_thread
native_init_IRQ
switch_to
/proc/interrupts
set_intr_gate
system_call
show_regs
interrupt
pt_regs
atomic_t
cli
sti

physical memory operations

arch/x86/mm/
mem_init
get_page_from_freelist
zonelist
zone
free_area
free_list
out_of_memory
die
num_physpages
do_page_fault

disk controller drivers

scsi_host_alloc
libata
Scsi_Host
ahci_pci_driver
aic94xx_init

network device drivers

drivers/net/
usbnet_probe
ipw2100_pci_init_one
zd1201_probe
e1000_xmit_frame
e1000_intr

electronics

user peripherals

keyboard
camera
mouse
graphics card
audio

I/O

I/O mem
I/O
PCI controller
I/O ports
ACPI
USB controller

CPU

registers
APIC
interrupt controller

memory

RAM
DMA
MMU

disk controllers

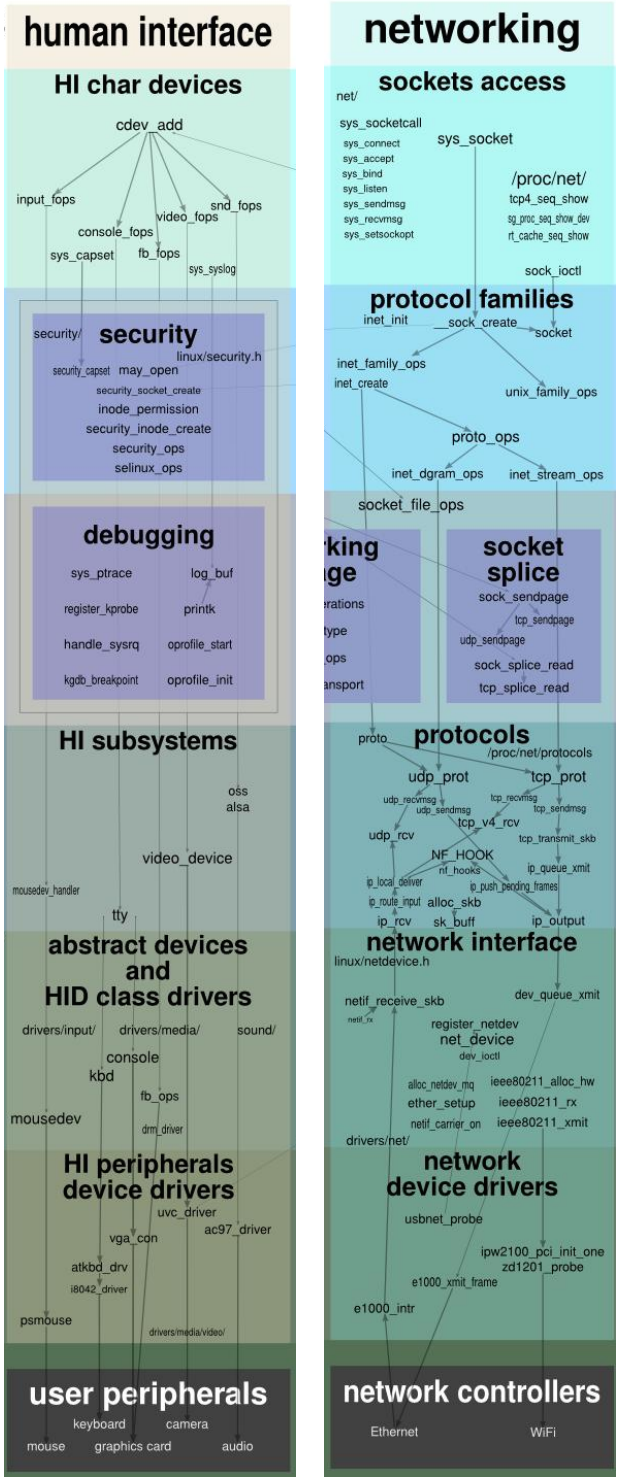
SCSI
SATA

network controllers

Ethernet
WiFi

© 2007, 2010 Constantine Shulyupin www.MakeLinux.net/kernel_map

- As time allows...



Other Topics

- Energy considerations/management
  - how to extend battery life, reduce cost
- Security
  - how systems get compromised
  - how can systems be protected
- Android case study
- Windows case study

Resources

- Class
  - https://classes.cs.siue.edu/
  - http://www.cs.siue.edu/~icrk
- Linux/bash
  - http://www.makelinux.com/home/
  - www.makelinux.com/books/Bash-Beginners-Guide/
  - http://www.makelinux.com/books/abs-guide/
- machine
  - os.cs.siue.edu
    - From unix-like systems, open a terminal and and type ssh <your SIUE username>@os.cs.siue.edu
    - From windows, use PuTTY, cygwin, or ...