Introduction

Sathish H Bowatta
B.Sc. in Computer Science - UCSC
CEH, CCNP, MCSA, MCTS

Essential Duties of the System/Network Administrator

- Account Provisioning
- Adding and Removing Hardware
- Performing Backups
- Installing and upgrading Software
- Monitoring the System
- Troubleshooting
- Maintaining local Documentation
- Vigilantly monitoring Security
- Fire Fighting

History of LINUX

MULTICS

Multiplexed **I**nformation and **C**omputing **S**ervice
[MIT/GE/Bell Labs]

Development initiated in **1964**

Time-sharing Operating System

Multics introduced any innovations, but had many problems

- Complex Architecture
- High resource demands

Bell Labs ended its participation of Multics in **1969**

MULTICS to UNIX

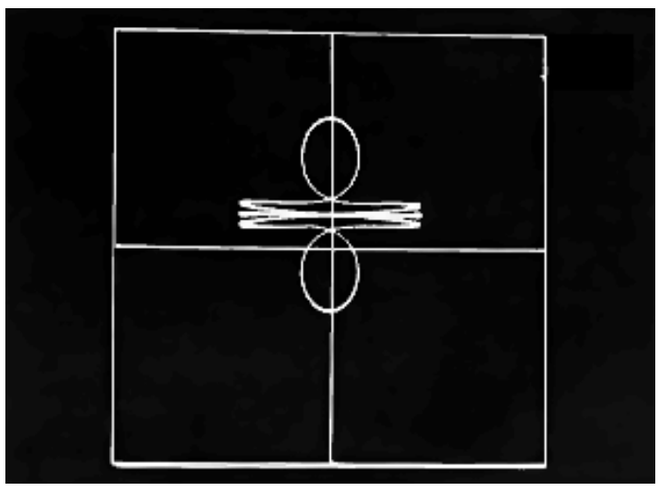
“… over-designed and overbuilt and over everything. It was close to unusable. They (i.e., Massachusetts Institute of Technology) still claim it’s a monstrous success, but it just clearly wasn’t."

“the things that I liked enough (about Multics) to actually take were the hierarchical file system and the shell — a separate process that you can replace with some other process.”

: Ken Thompson

SPACE TRAVEL

During his work with Multics Ken Thompson developed **Space Travel** on a GE 635 computer

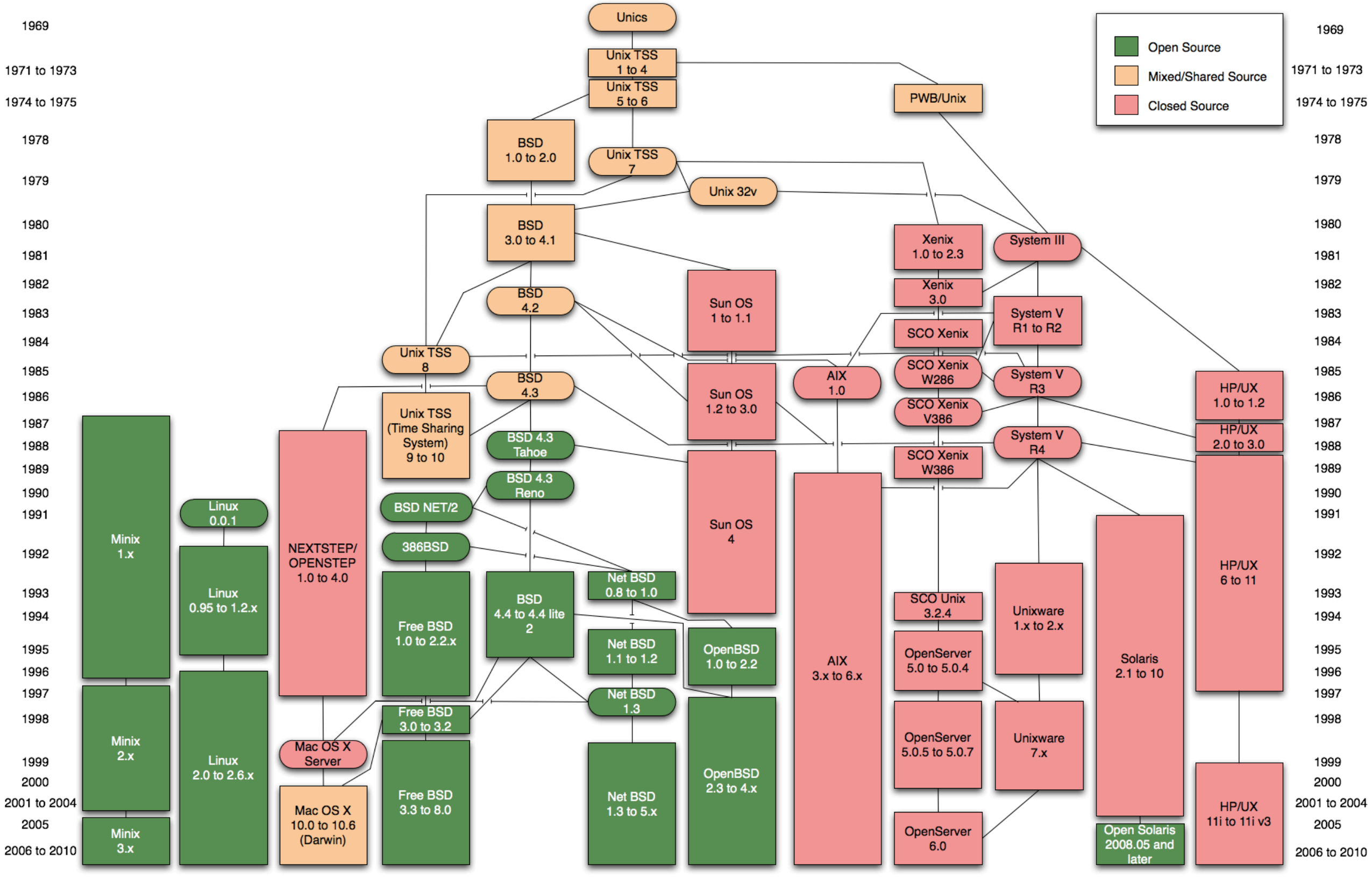


Need for a more efficient and less expensive machine to run on: **PDP -7**

As Thompson began porting the game to the new system, he decided not to base the code on any of the existing software for the computer, and instead write his own

By the time Space Travel was fully ported to the PDP-7, Thompson and his colleagues had expanded his software suite to a full, basic operating system

Origins of LINUX


Open Source
Mixed/Shared Source
Closed Source
1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2010
Unics
Unix TSS 1 to 4
Unix TSS 5 to 6
Unix TSS 7
PWB/Unix
BSD 1.0 to 2.0
Unix 32v
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
BSD 4.2
Sun OS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix TSS 8
SCO Xenix W286
AIX 1.0
BSD 4.3
Sun OS 1.2 to 3.0
System V R3
HP/UX 1.0 to 1.2
SCO Xenix V386
Unix TSS (Time Sharing System) 9 to 10
BSD 4.3 Tahoe
System V R4
HP/UX 2.0 to 3.0
SCO Xenix W386
BSD 4.3 Reno
BSD NET/2
Sun OS 4
Minix 1.x
Linux 0.0.1
NEXTSTEP/ OPENSTEP 1.0 to 4.0
386BSD
HP/UX 6 to 11
Linux 0.95 to 1.2.x
Free BSD 1.0 to 2.2.x
BSD 4.4 to 4.4 lite 2
Net BSD 0.8 to 1.0
SCO Unix 3.2.4
Unixware 1.x to 2.x
Solaris 2.1 to 10
Net BSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.0.4
Net BSD 1.3
Free BSD 3.0 to 3.2
Minix 2.x
Linux 2.0 to 2.6.x
Mac OS X Server
OpenServer 5.0.5 to 5.0.7
Unixware 7.x
OpenBSD 2.3 to 4.x
AIX 3.x to 6.x
Free BSD 3.3 to 8.0
Net BSD 1.3 to 5.x
Mac OS X 10.0 to 10.6 (Darwin)
OpenServer 6.0
Open Solaris 2008.05 and later
HP/UX 11i to 11i v3
Minix 3.x

UNICS to UNIX

In 1970, Peter G. Neumann coined the project name **Unics** as a pun on Multics

Thompson and Ritchie added text processing capabilities to Unix and received funding for a PDP- 11/20.

For the first time in 1970, the Unix operating system was officially named and ran on the PDP-11/20

UNIX

UNIX was originally written in assembler and B

Dennis Ritchie improves B and named it C

In 1972, Unix was rewritten in the C programming language to make it portable

Bell Labs produced several versions of Unix that are collectively referred to as Research Unix

The availability and portability of Unix caused it to be widely adopted, copied and modified by academic institutions and businesses. (BSD and System V)

UNIX Wars

BSD

In 70s AT&T was under a courts order not

to sell software

AT&T gave away UNIX to Universities charging only for media

In 1977, the Berkeley Software Distribution (BSD) was developed by the Computer Systems Research Group (CSRG) from UC Berkeley, based on the 6th edition of Unix

BSD too went through many releases until BSD 4.4 was released.

SYSTEM V

In 1984, AT&T was divested, and was allowed to sell UNIX

AT&T developed more versions, until it released a commercial version called System 3 and this was followed by System V Release 4

Since BSD contained Unix code that AT&T owned, AT&T filed a lawsuit (USL v. BSDi) in the early 1990s against the University of California. This strongly limited the development and adoption of BSD

UNIX became commercial, source code restricted

The Philosophy of Open-Source and LINUX

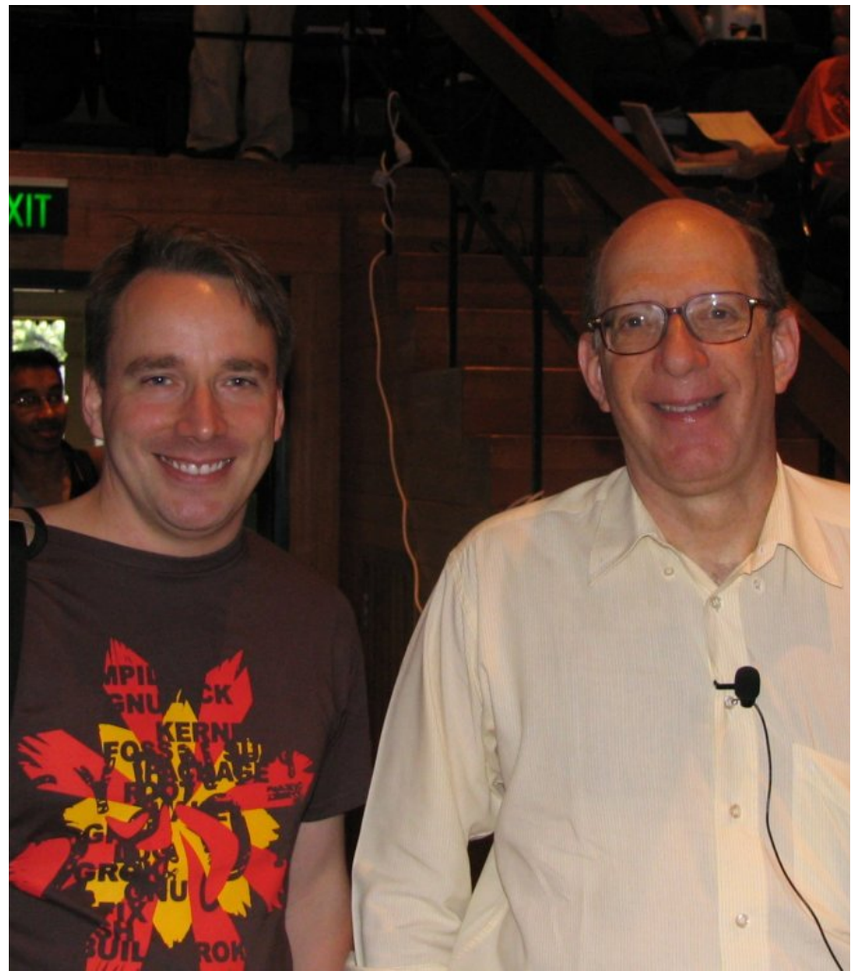
- Free Software Foundation
- Richard Stallman (RMS)
- Open Sourced Software
- GNU GPL

The GNU Project

- Richard M Stallman (RMS) left MIT AI Labs to found the GNU Project under Free Software Foundation
- GNU : GNU is Not Unix
- The goal of the GNU was to create a free UNIX like operating system
- As part of this work, he wrote the GNU General Public License (GPL)
- Users are free to run the software, share it (copy, distribute), study it and modify it
- By the early 1990s, there was almost enough available software to create a full operating system
- However, the GNU kernel, called Hurd, failed to attract enough development effort, leaving GNU incomplete

MINIX

- A famous professor Andrew Tanenbaum developed **Minix**, a simplified version of **UNIX** that runs on PC
- Minix was for class teaching only. No intention for commercial use
- While source code for the system was available, modification and redistribution were restricted



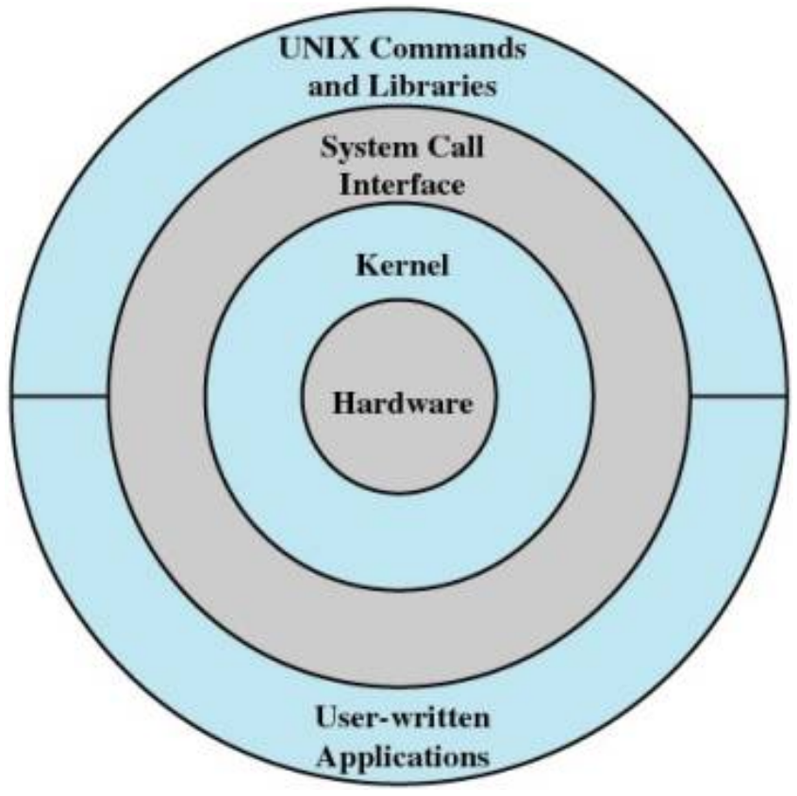
LINUX

- In Sept 1991, Linus Torvalds, a second year student of Computer Science at the University of Helsinki, developed the preliminary kernel of Linux, known as Linux version 0.0.1
- Since its source code was available, Linus decided to take Minix as a model. In his own words,
  'I wanted to write a better Minix than Minix'

LINUX

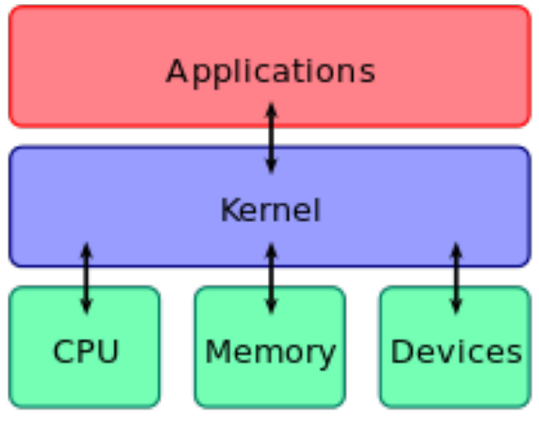
Linux itself is just **kernel**

The heart of the system, takes care of memory management, interrupt handling (i.e. a common interface between user process and hardware)



The kernel is only useful when used in conjunction with other software

- GNU Project
- XFree86
- Others



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 | video_fops | snd_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 | cdev_map | sys_epoll_create | cdev | sys_reboot | sys_init_module | kvm_dev_ioctl

processes

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

memory access

mm/ | sys_brk | sys_mmap | shm_vm_ops | sys_shmctl | sys_shmat | sys_sysinfo | 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_pipe | sys_poll | sys_select | sys_flock | do_path_lookup | sys_readv | iovec | sys_splice | sys_sendfile | sys_getdents | sys_swapon | alloc_file | sys_chown | sys_chmod | notify_change | fd | sys_newfstat | sys_fsync | fsnotify_change | 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

virtual

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

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_read | vfs_write | vfs_fstat | vfs_getattr | vfs_create | inode | fsnotify | 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

load_module | module | module_param | kernel_param | kobject_uevent_init | kobject_uevent

synchronization

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

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

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 | netif_rx | dev_queue_xmit | 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 | 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

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 ports | ACPI | USB controller | PCI controller

CPU

registers | APIC | interrupt controller

memory

RAM | DMA | MMU

disk controllers

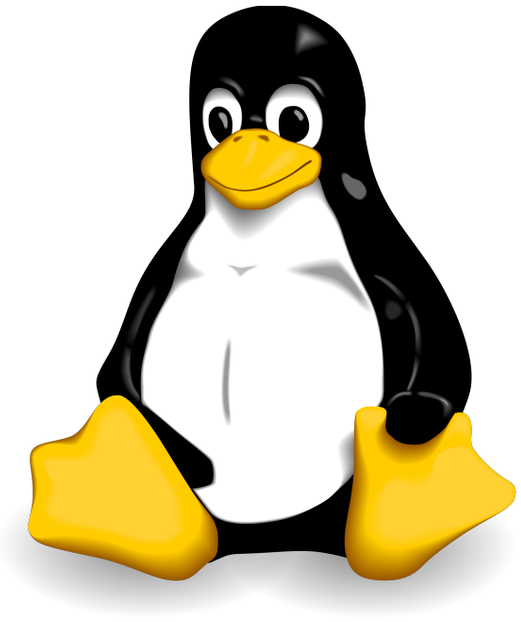
SCSI | SATA

network controllers

Ethernet | WiFi

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

The **LINUX** System



Linux utilizes tools developed as part of:

- Berkeley’s BSD operating system
- MIT’s X Window System
- the Free Software Foundation's GNU project.

The system libraries were started by the GNU project, with improvements provided by the Linux community.

Linux is an independent **POSIX** implementation and includes true multitasking, virtual memory, shared libraries, demand loading, proper memory management, TCP/IP networking, and other features consistent with Unix-type systems.

Licensed under the GNU GPL license (more on this later)
Freedom from viruses (Well almost..!!)

- Strong architecture
- Security measures

Dozens of excellent, free, general-interest desktop applications. Linux cannot possibly be put out-of-business

LINUX Distributions

A particular assortment of application and utility software (various GNU tools and libraries, for example), packaged together with the Linux kernel in such a way that its capabilities meet the needs of many users

Windows: Single package

Linux :

- Linux kernel
- GNU tools and libraries
- A window system (X Window System)
- A window manager
- The desktop environment (which runs on the X server to provide a graphical desktop)
- Additional Software
- Desktop applications (web browsers, email programs, word processors)

Installation

Minimum System Requirements

- 700 MHz processor (about Intel Celeron or better)
- 512 MiB RAM (system memory)
- 5 GB of hard-drive space (or USB stick, memory card or external drive but see LiveCD for an alternative approach)
- VGA capable of 1024x768 screen resolution
- Either a CD/DVD drive or a USB port for the installer media
- Internet access is helpful

Determining Partition Sizes

Root files system:

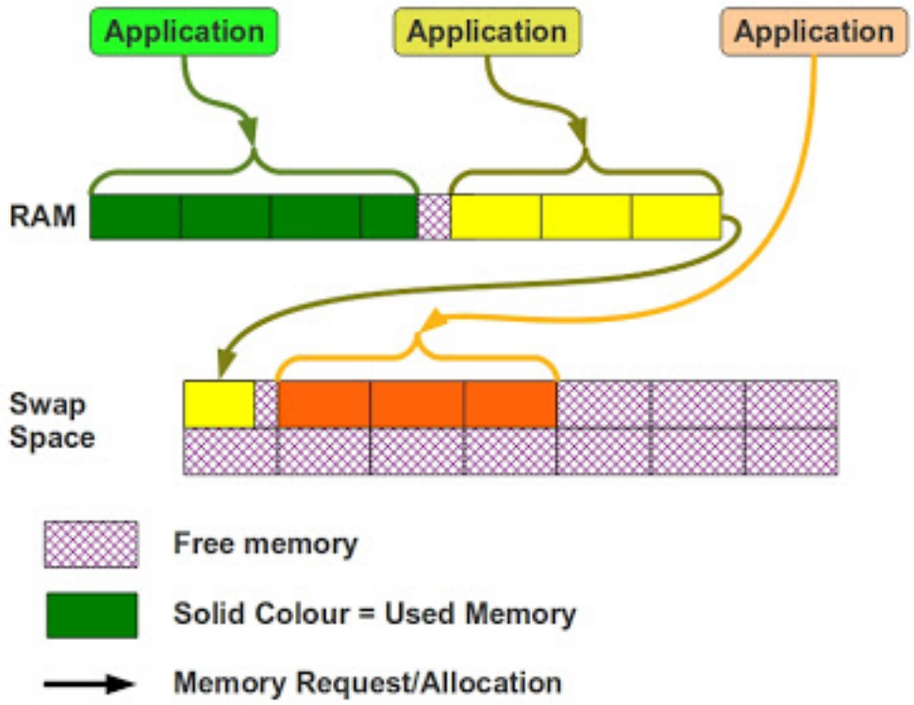
- Distribution type (60MB-2GB)
- User directories
- Softwares

Swap partition

Rule of thumb: Twice as the space in your RAM

SWAP Space

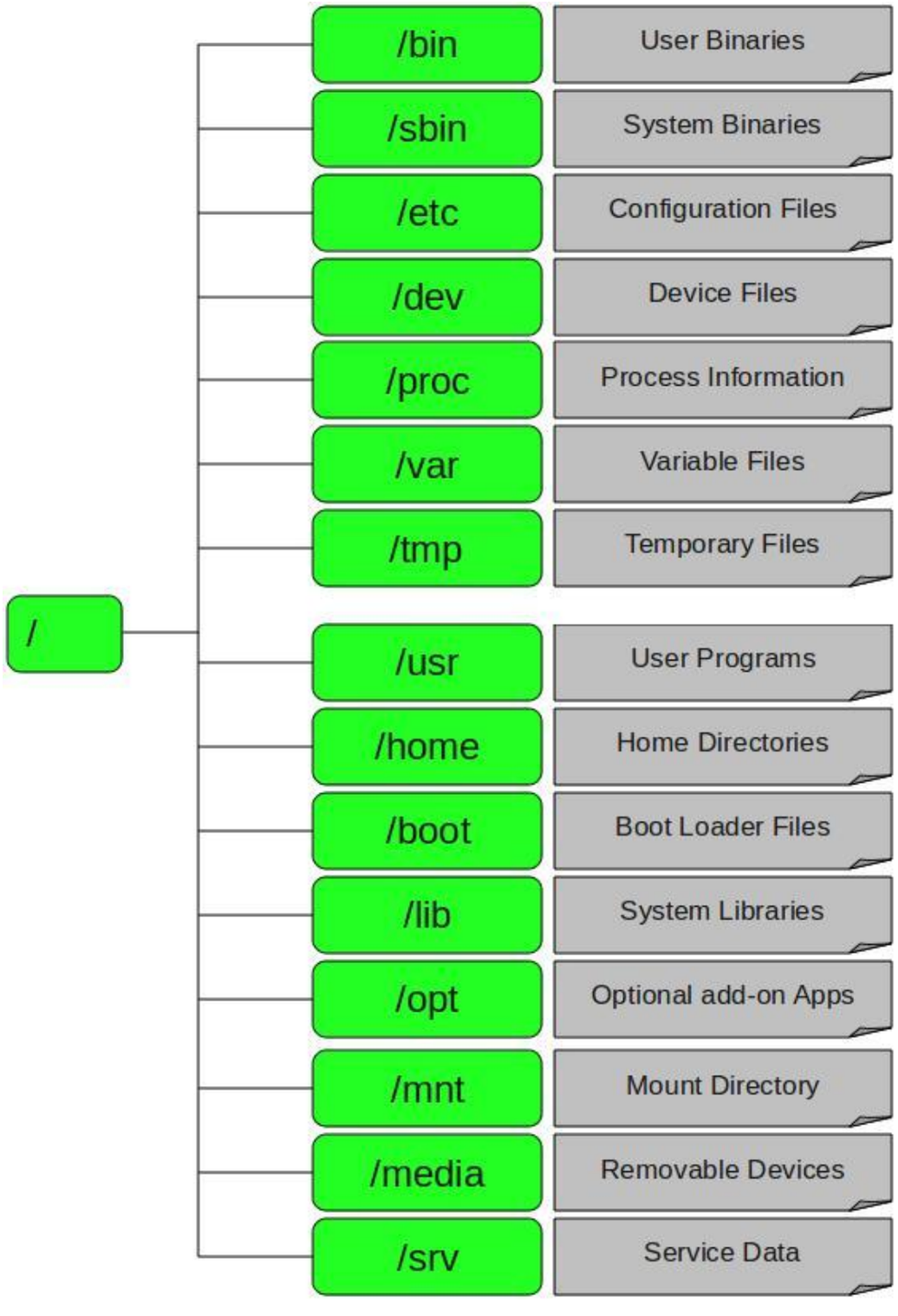
Swap space is a portion of the disk used by an operating system to temporarily store parts of programs that were loaded by the user but aren't currently in use.



You are not required to use swap space with Linux, but if you have less than 256 MB of physical RAM, it is strongly suggested that you do.

Linux: At least two partitions

LINUX Directory Structure



LINUX Boot Process

Boot Loader

Booting

Initialization of a computerized system

OS from secondary storage to primary storage

A boot loader is a small program that places the operating system (OS) of a computer into memory.

BIOS

- Basic Input Output System
- Executes the MBR

MBR

- Master Boot Record
- Executes the GRUB

GRUB

- Grand Unified Boot Loader
- Executes the Kernel

Kernel

- Central Core of the OS
- Executes /sbin/init

Init

- Initialization
- Executes the Run Level Programs

Run Level

- Run Level programs are executed
- From /etc/rc.d/rc*.d/

Run Levels

At level 0, the system is completely shut down.

Levels 1 and S represent single-user mode.

Levels 2 through 5 include support for networking.

Level 6 is a “reboot” level.

Scripting and The Shell

UNIX Shells

Bourne shell – sh

Korn shell – ksh

C shell – csh

Bash – bash

Enhanced C shell (a freeware shell derived from the C shell) – tcsh

Z shell (a freeware shell derived from the Korn shell) – zsh

- echo, cat, less, head, tail, less, view
- mkdir, touch, rmdir, rm,
- chmod, chown
- ls, cd, ln, cp, mv, tee
- wget, curl
- cut, wc, uniq, diff, sort
- ps, top, netstat
- grep, find, locate
- 2> /dev/null

{ return, exit }

Variables

- a=1
- b=‘String’
- c=$(())
- d=(a,b,$c)

Control Flow

Elementary **BASH** comparison operators

String	Numeric	True if
x = y	x -eq y	x is equal to y
x != y	x -ne y	x is not equal to y
x < y	x -lt y	x is less than y
x <= y	x -le y	x is less than or equal to y
x > y	x -gt y	x is greater than y
x >= y	x -ge y	x is greater than or equal to y
-n x		x is not null
-z x		x is null

If..Elif..Else..FI

```
if [ $a -eq 10 ]; then
## do something
elif [ $a -gt 10 ]; then
## do something
else
## do something
fi
```

* spaces

BASH file evaluation operators

-d file	file exists and is a directory
-e file	file exists
-f file	file exists and is a regular file
-r file	you have read permission on file
-s file	file exists and is not empty
-w file	you have write permission on file
file1 -nt file2	file1 is newer than file2
file1 -ot file2	file1 is older than file2

Case..Esac

```
case $a in

pattern1)
## do something
        ;;
        pattern2)
        ## do something
        ;;
        esac
```

* semicolons {;;}

Functions

```
function foo ( ) {
## do something, return, exit
        }

        foo
        foo “$var”
```

* spaces

Arithmetics

```
a=1

b=$((2))

c=$a+$b
```

```
d=$(($a+$b))

echo “$a + $b = $c”

echo “$a + $b = $d”
```

* parentheses {(())}

Arrays

```
array=(1 , ‘2 3’, 4)

array[3]=5

        echo “array[@] = ${array[@]}”

        echo “${#array[@]}”
```

* no spaces

Regular Expressions

.	Matches any character
[chars]	Matches any character from a given set
[^chars]	Matches any character not in a given set
^	Matches the beginning of a line
$	Matches the end of a line
\w	Matches any “word” character (same as [A-Za-z0-9_])
\s	Matches any whitespace character (same as [f\t\n\t])a
\d	Matches any digit (same as [0-9])
|	Matches either the element to its left or the one to its right
(expr)	Limits scope, groups elements, allows matches to be captured
?	Allows zero or one match of the preceding element
*	Allows zero, one, or many matches of the preceding element
+	Allows one or more matches of the preceding element
{n}	Matches exactly n instances of the preceding element
{min}	Matches at least min instances (note the comma)
{min, max}	Matches any number of instances from min to max

I/O Redirection

Standard Input

```
ls > filename
ls >> filename
```

Standard Output

```
sort < filename
sort << filename
```

Pipes

```
ls -l | less
```

Filters

```
sort, uniq, grep, fmt, pr, head, tail, tr, sed, awk
```

Named Pipes (FIFO)

- Named pipes exist as a device special file in the file system.
- Processes of different ancestry can share data through a named pipe.
- When all I/O is done by sharing processes, the named pipe remains in the file system for later use.

Create a FIFO

```
mkfifo -m 0666 /tmp/namedfifo
```

FIFO Operations

```
tail -f /tmp/namedfifo
```

```
echo “Something” >>  /tmp/namedfifo
```