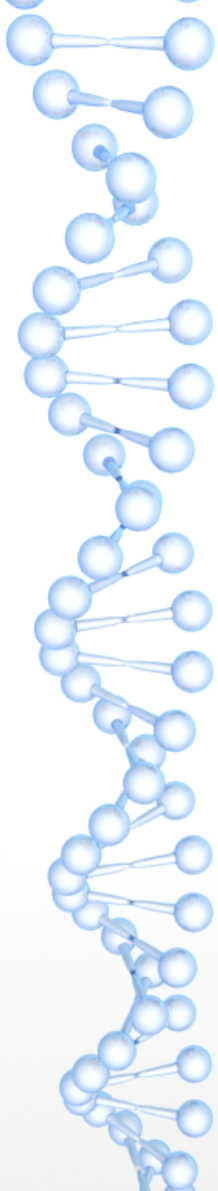
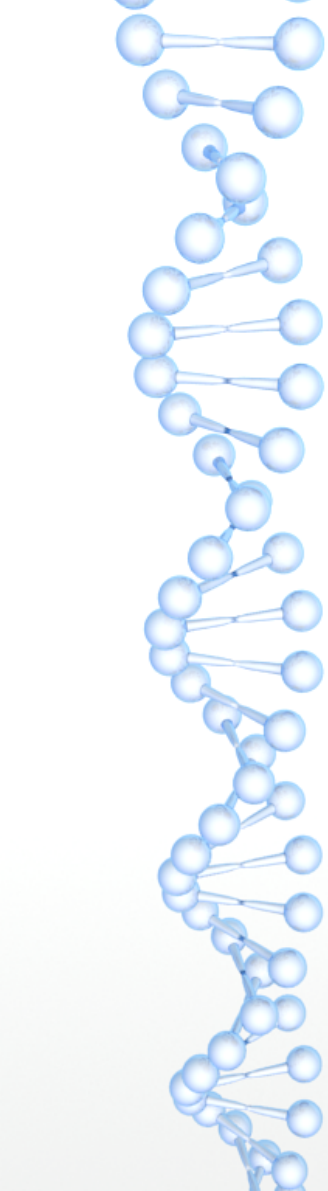


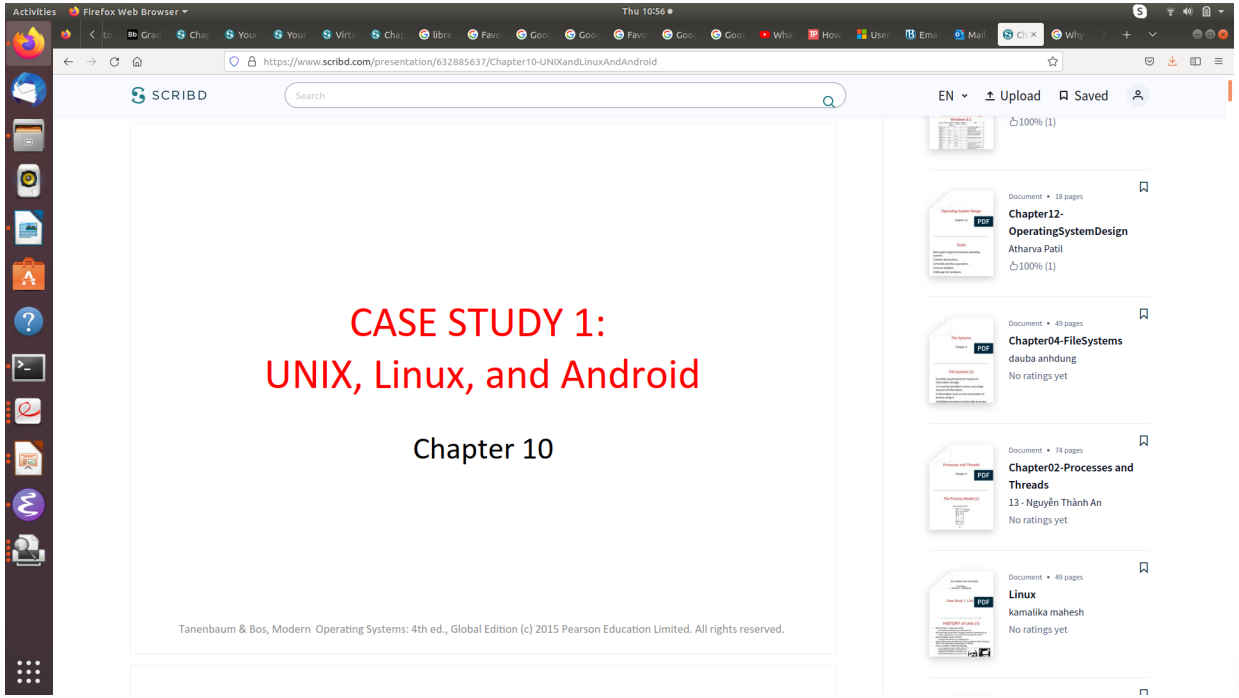
# Chapter 10 Linux/Unix/Android and Ch11/5.6 Windows11

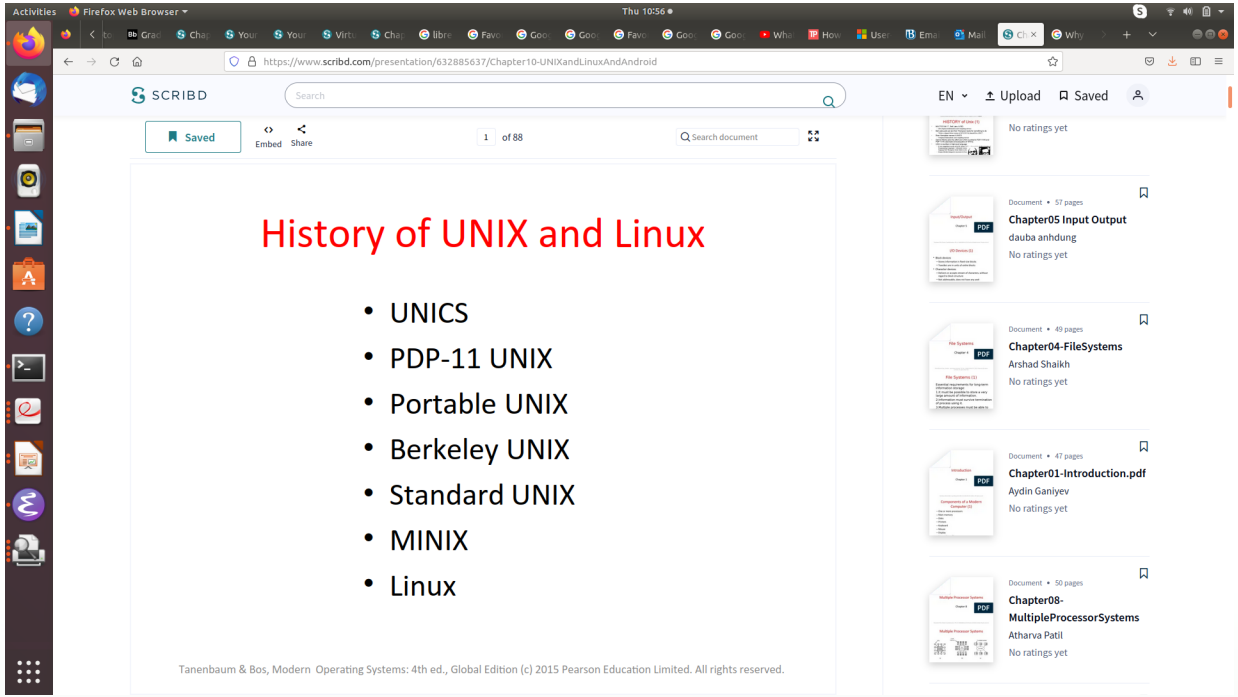
We want to look at specifics for Linux, Android and Windows.

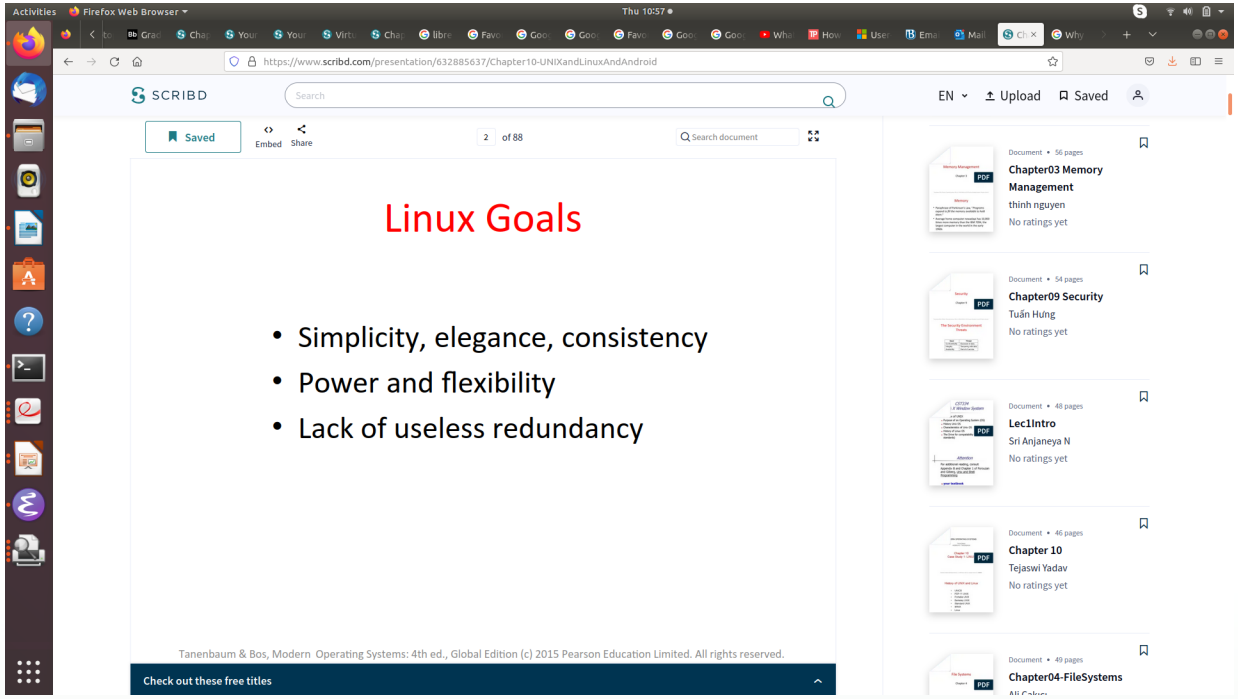


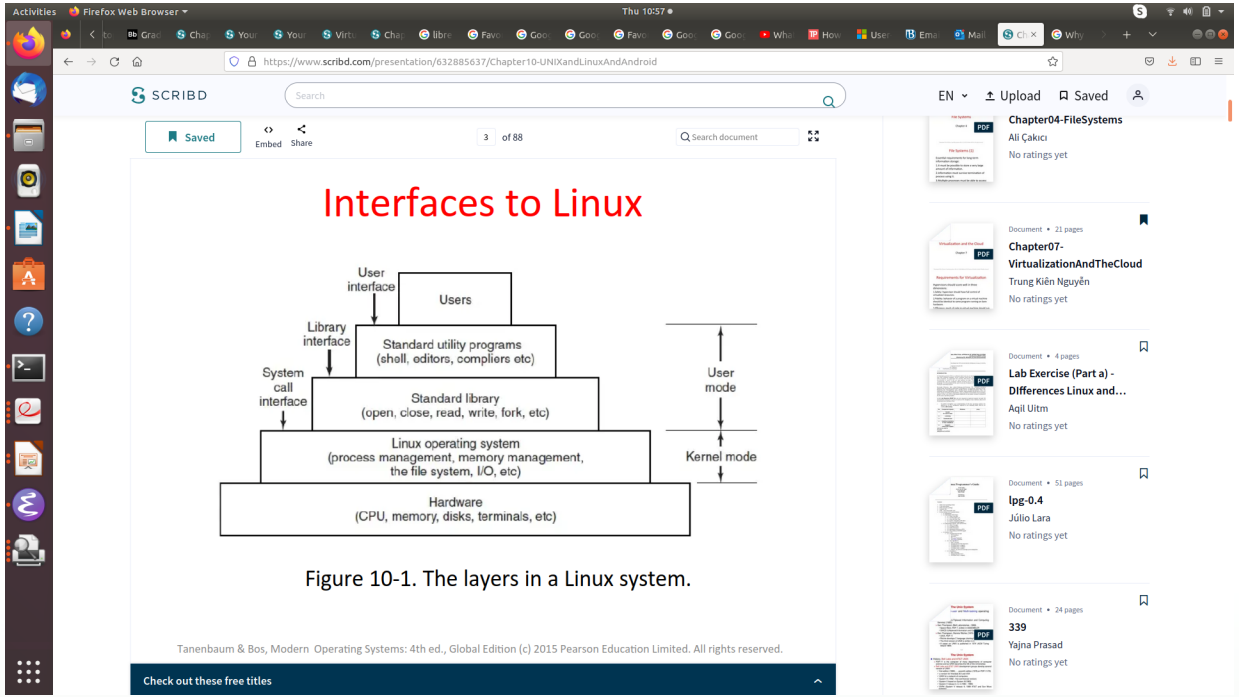
# Linux introduction

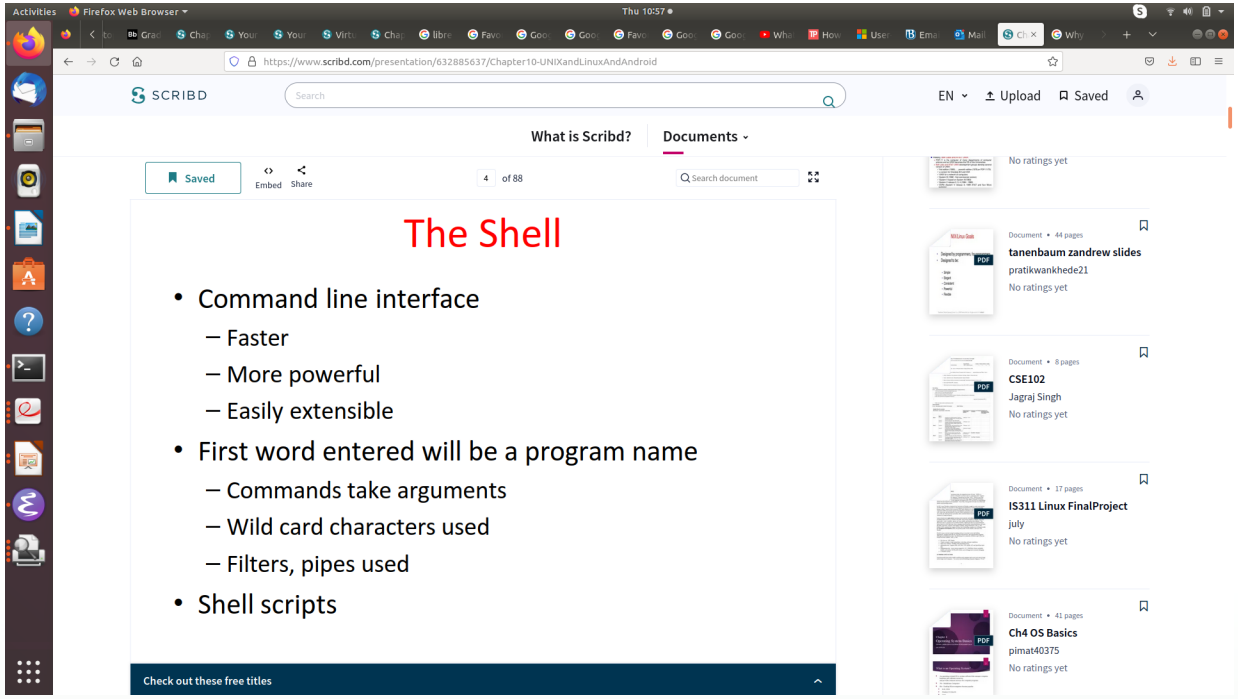
- 10.1 History of UNIX and Linux
- 10.2 Overview of Linux: goals, interfaces, shell, kernel
- 10.3 Processes in Linux

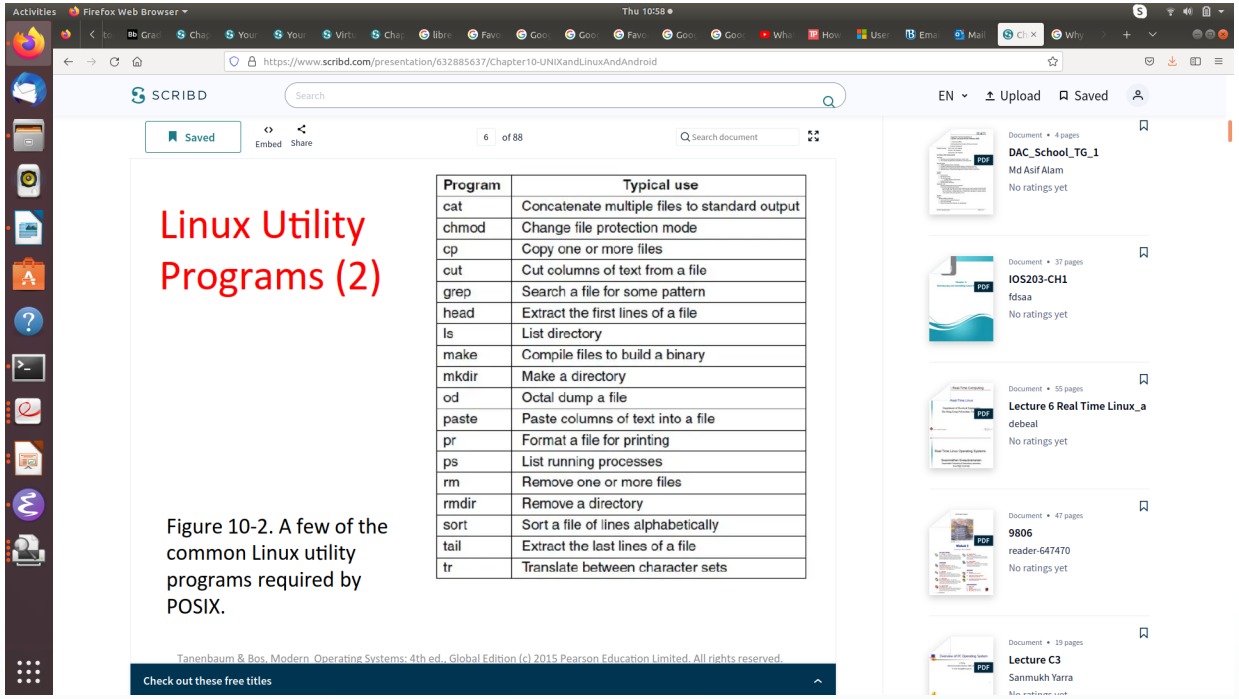

Activities
Firefox Web Browser
Thu 10:56
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
CASE STUDY 1:
UNIX, Linux, and Android
Chapter 10
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Document • 18 pages
Chapter12-OperatingSystemDesign
Atharva Patil
100% (1)
Document • 49 pages
Chapter04-FileSystems
dauba anhdung
No ratings yet
Document • 74 pages
Chapter02-Processes and Threads
13 - Nguyễn Thành An
No ratings yet
Document • 49 pages
Linux
kamalika mahesh
No ratings yet

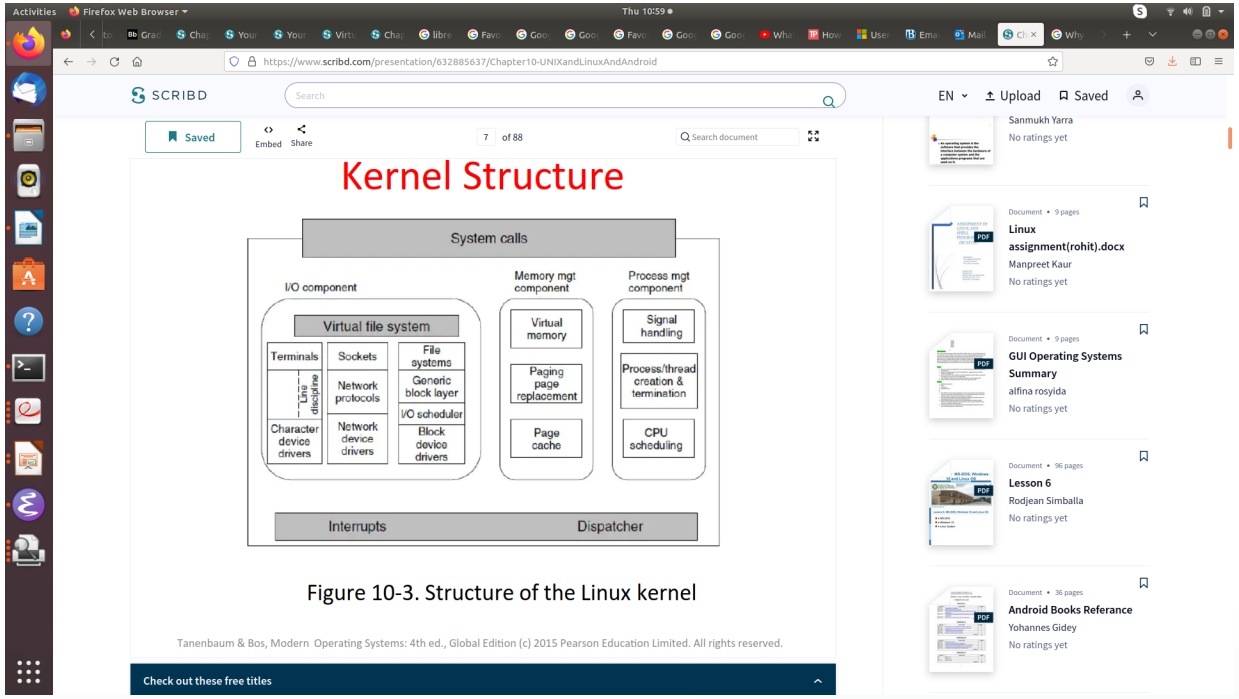

Activities
Firefox Web Browser
Thu 10:56
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
1 of 88
Search document
History of UNIX and Linux
UNICS
PDP-11 UNIX
Portable UNIX
Berkeley UNIX
Standard UNIX
MINIX
Linux
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
No ratings yet
Document • 57 pages
Chapter05 Input Output
dauba anhdung
No ratings yet
Document • 49 pages
Chapter04-FileSystems
Arshad Shaikh
No ratings yet
Document • 47 pages
Chapter01-Introduction.pdf
Aydin Ganiyev
No ratings yet
Document • 50 pages
Chapter08-MultipleProcessorSystems
Atharva Patil
No ratings yet

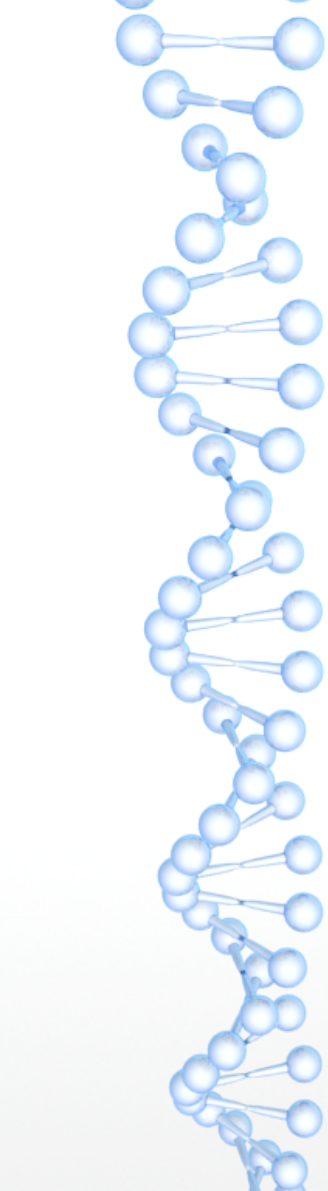

Activities
Firefox Web Browser
Thu 10:57
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
2
of 88
Search document
Linux Goals
• Simplicity, elegance, consistency
• Power and flexibility
• Lack of useless redundancy
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles
Document • 56 pages
Chapter03 Memory Management
thinh nguyen
No ratings yet
Document • 54 pages
Chapter09 Security
Tuấn Hưng
No ratings yet
Document • 48 pages
Lec1Intro
Sri Anjaneya N
No ratings yet
Document • 46 pages
Chapter 10
Tejaswi Yadav
No ratings yet
Document • 49 pages
Chapter04-FileSystems


Activities
Firefox Web Browser
Thu 10:57
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
3
of 88
Search document
Interfaces to Linux
User interface
Users
Library interface
Standard utility programs (shell, editors, compliers etc)
System call interface
Standard library (open, close, read, write, fork, etc)
Linux operating system (process management, memory management, the file system, I/O, etc)
Hardware (CPU, memory, disks, terminals, etc)
User mode
Kernel mode
Figure 10-1. The layers in a Linux system.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles
Chapter04-FileSystems
Ali Çakıcı
No ratings yet
Document • 21 pages
Chapter07-VirtualizationAndTheCloud
Trung Kiên Nguyễn
No ratings yet
Document • 4 pages
Lab Exercise (Part a) - Differences Linux and...
Aqil Uitm
No ratings yet
Document • 51 pages
lpg-0.4
Júlio Lara
No ratings yet
Document • 24 pages
339
Yajna Prasad
No ratings yet


Activities
Firefox Web Browser
Thu 10:57
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
What is Scribd?
Documents
Saved
Embed
Share
4
of 88
Search document
The Shell
• Command line interface
– Faster
– More powerful
– Easily extensible
• First word entered will be a program name
– Commands take arguments
– Wild card characters used
– Filters, pipes used
• Shell scripts
Check out these free titles
No ratings yet
Document • 44 pages
tanenbaum zandrew slides
pratikwankhede21
No ratings yet
Document • 8 pages
CSE102
Jagraj Singh
No ratings yet
Document • 17 pages
IS311 Linux FinalProject
july
No ratings yet
Document • 41 pages
Ch4 OS Basics
pimat40375
No ratings yet

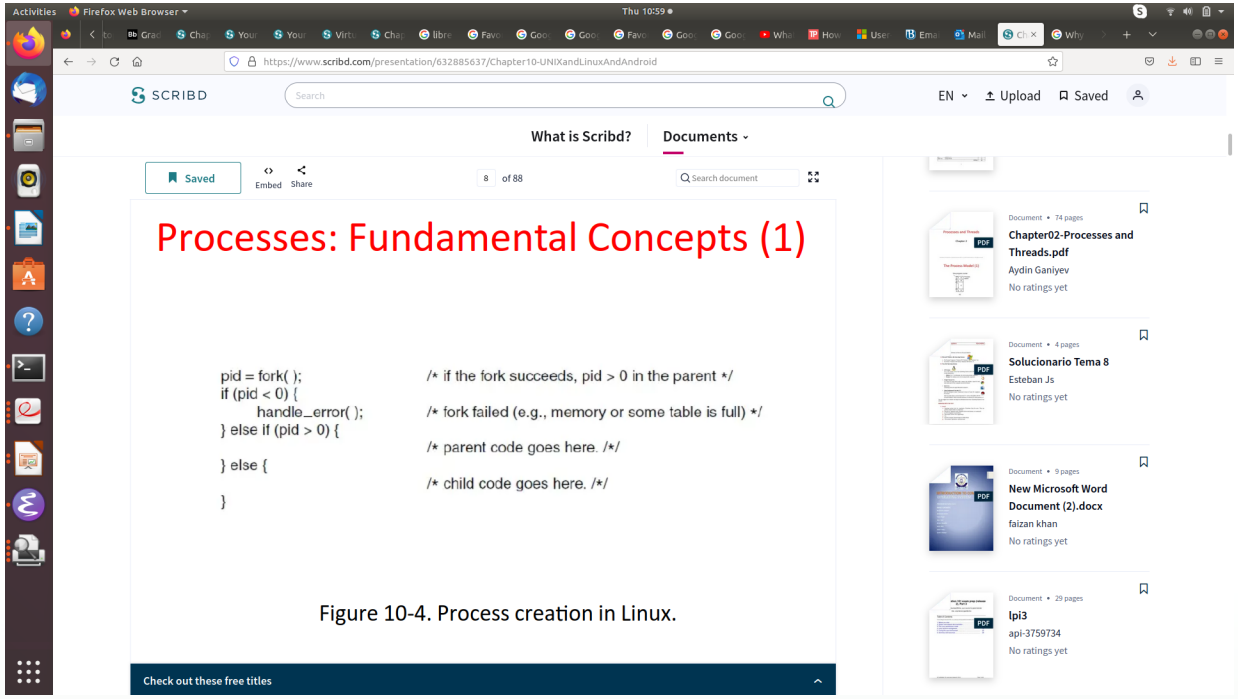

Activities
Firefox Web Browser
Thu 10:58
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
6 of 88
Search document
Linux Utility Programs (2)
Program | Typical use
cat | Concatenate multiple files to standard output
chmod | Change file protection mode
cp | Copy one or more files
cut | Cut columns of text from a file
grep | Search a file for some pattern
head | Extract the first lines of a file
ls | List directory
make | Compile files to build a binary
mkdir | Make a directory
od | Octal dump a file
paste | Paste columns of text into a file
pr | Format a file for printing
ps | List running processes
rm | Remove one or more files
rmdir | Remove a directory
sort | Sort a file of lines alphabetically
tail | Extract the last lines of a file
tr | Translate between character sets
Figure 10-2. A few of the common Linux utility programs required by POSIX.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles
Document • 4 pages
DAC_School_TG_1
Md Asif Alam
No ratings yet
Document • 37 pages
IOS203-CH1
fdsaa
No ratings yet
Document • 55 pages
Lecture 6 Real Time Linux_a
debeal
No ratings yet
Document • 47 pages
9806
reader-647470
No ratings yet
Document • 19 pages
Lecture C3
Sanmukh Yarra

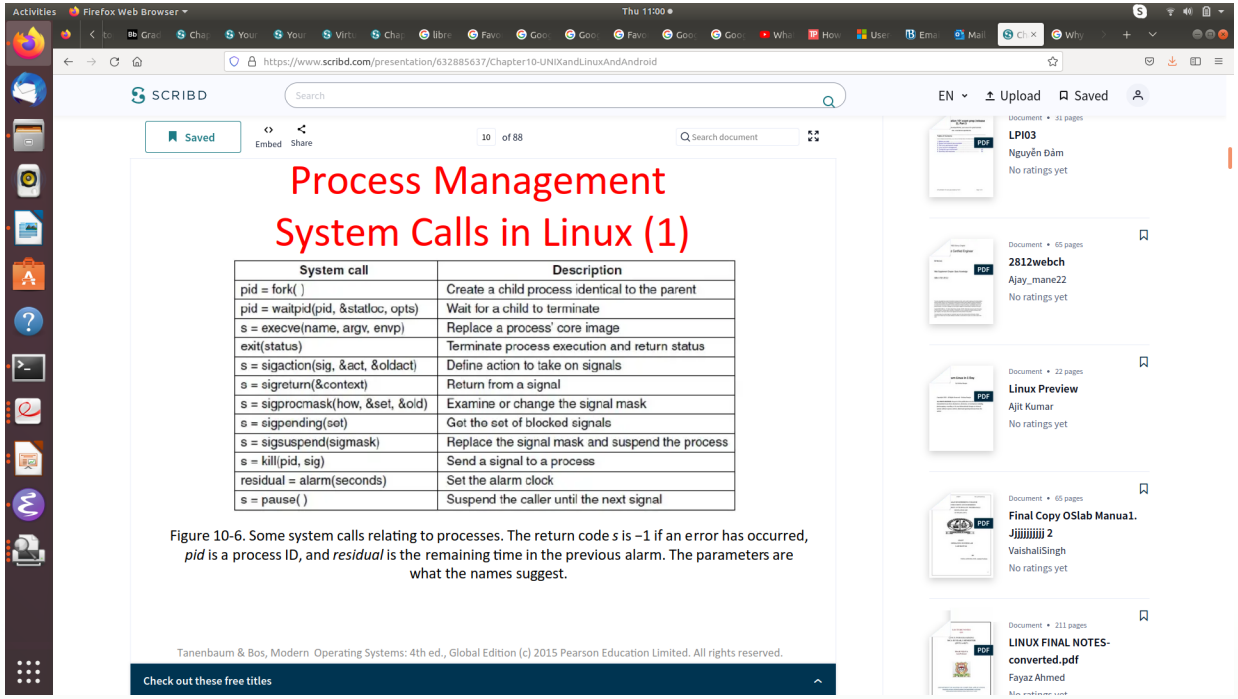

Activities
Firefox Web Browser
Thu 10:59
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
7 of 88
Search document
Kernel Structure
System calls
I/O component
Memory mgt component
Process mgt component
Virtual file system
Terminals
Sockets
File systems
Line discipline
Network protocols
Generic block layer
I/O scheduler
Character device drivers
Network device drivers
Block device drivers
Virtual memory
Paging page replacement
Page cache
Signal handling
Process/thread creation & termination
CPU scheduling
Interrupts
Dispatcher
Figure 10-3. Structure of the Linux kernel
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles
Sanmukh Yarra
No ratings yet
Document • 9 pages
Linux assignment(rohit).docx
Manpreet Kaur
No ratings yet
Document • 9 pages
GUI Operating Systems Summary
alfina rosyida
No ratings yet
Document • 96 pages
Lesson 6
Rodjean Simballa
No ratings yet
Document • 36 pages
Android Books Referance
Yohannes Gidey
No ratings yet

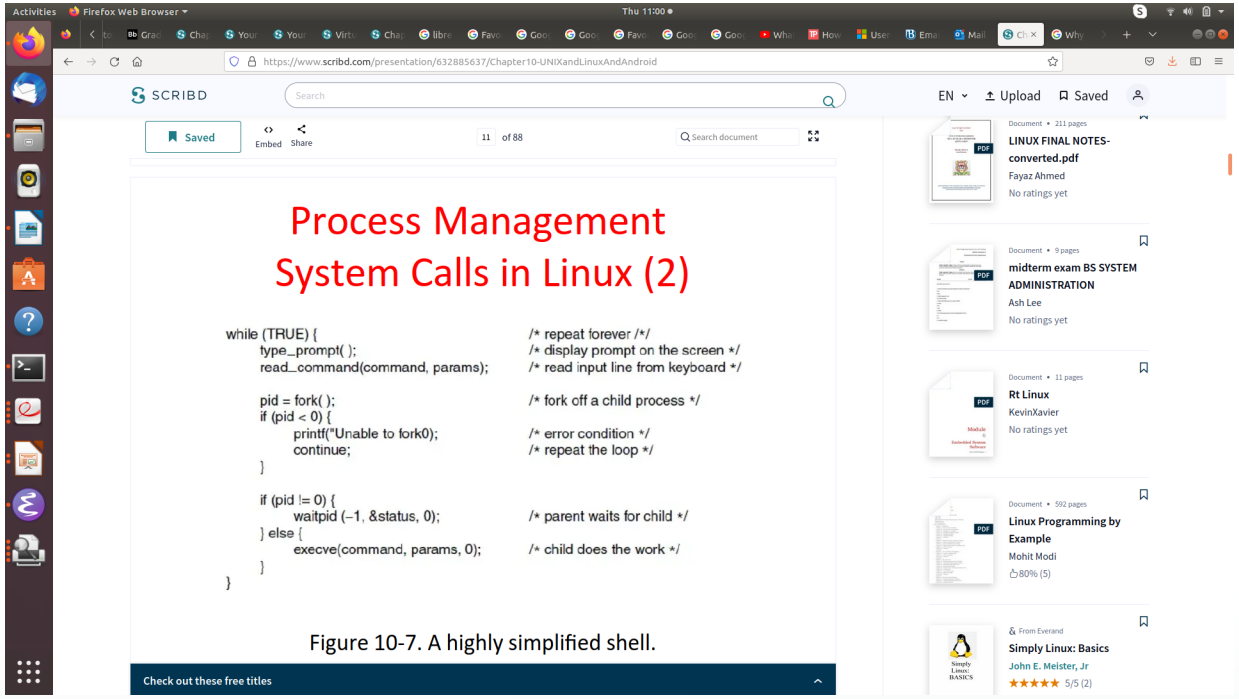


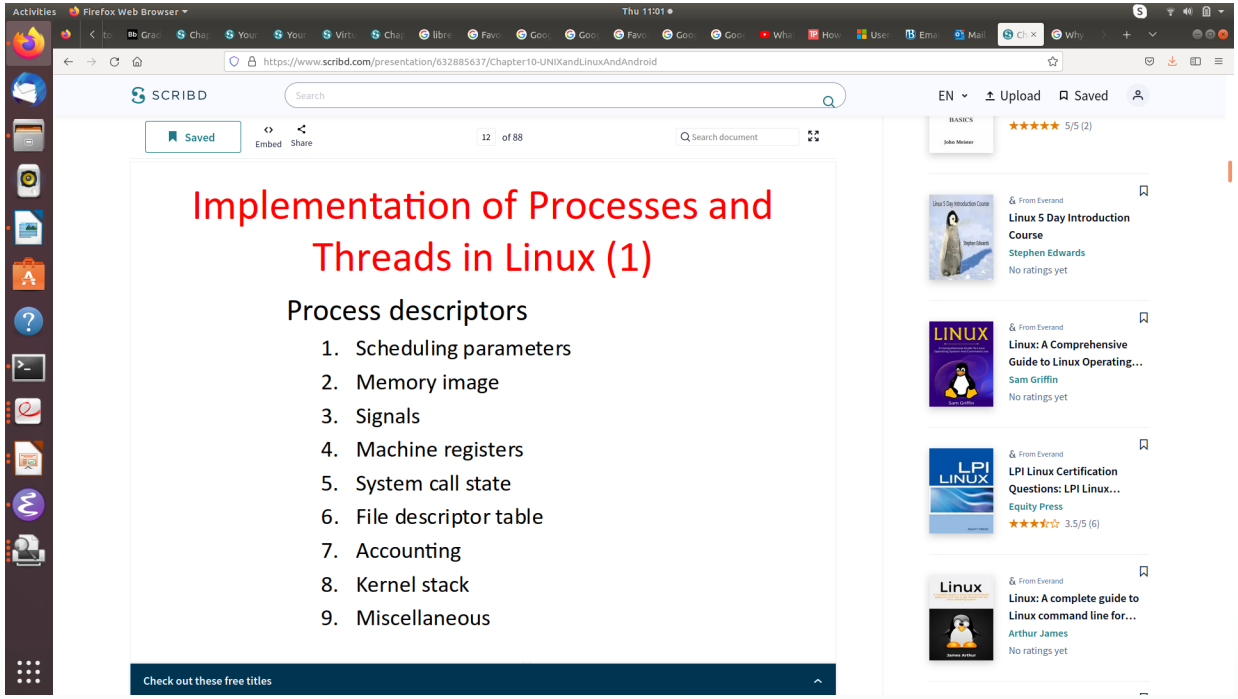
# 10.3 Processes in Linux

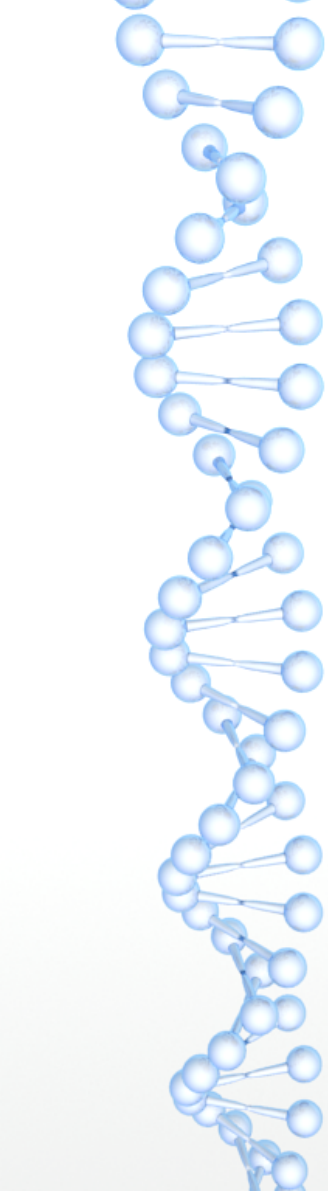
- Main activities in a Linux system are the processes.
- They are similar to classical sequential processes studied in Ch 2.
- Linux is multiprocessing so daemon processes run in background mode.


Activities
Firefox Web Browser
Thu 10:59
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
What is Scribd?
Documents
Saved
Embed
Share
8 of 88
Search document
Processes: Fundamental Concepts (1)
pid = fork( );
if (pid < 0) {
handle_error( );
} else if (pid > 0) {
} else {
}
/* if the fork succeeds, pid > 0 in the parent */
/* fork failed (e.g., memory or some table is full) */
/* parent code goes here. /*/
/* child code goes here. /*/
Figure 10-4. Process creation in Linux.
Check out these free titles
Document • 74 pages
Chapter02-Processes and Threads.pdf
Aydin Ganiyev
No ratings yet
Document • 4 pages
Solucionario Tema 8
Esteban Js
No ratings yet
Document • 9 pages
New Microsoft Word Document (2).docx
faizan khan
No ratings yet
Document • 29 pages
lpi3
api-3759734
No ratings yet


Activities
Firefox Web Browser
Thu 11:00
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
10 of 88
Search document
Process Management
System Calls in Linux (1)
System call | Description
pid = fork( ) | Create a child process identical to the parent
pid = waitpid(pid, &statloc, opts) | Wait for a child to terminate
s = execve(name, argv, envp) | Replace a process' core image
exit(status) | Terminate process execution and return status
s = sigaction(sig, &act, &oldact) | Define action to take on signals
s = sigreturn(&context) | Return from a signal
s = sigprocmask(how, &set, &old) | Examine or change the signal mask
s = sigpending(set) | Get the set of blocked signals
s = sigsuspend(sigmask) | Replace the signal mask and suspend the process
s = kill(pid, sig) | Send a signal to a process
residual = alarm(seconds) | Set the alarm clock
s = pause( ) | Suspend the caller until the next signal
Figure 10-6. Some system calls relating to processes. The return code s is −1 if an error has occurred, pid is a process ID, and residual is the remaining time in the previous alarm. The parameters are what the names suggest.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles
LPI03
Nguyễn Đảm
No ratings yet
2812webch
Ajay_mane22
No ratings yet
Linux Preview
Ajit Kumar
No ratings yet
Final Copy OSlab Manua1. Jjjjjjjjjj 2
VaishaliSingh
No ratings yet
LINUX FINAL NOTES-converted.pdf
Fayaz Ahmed

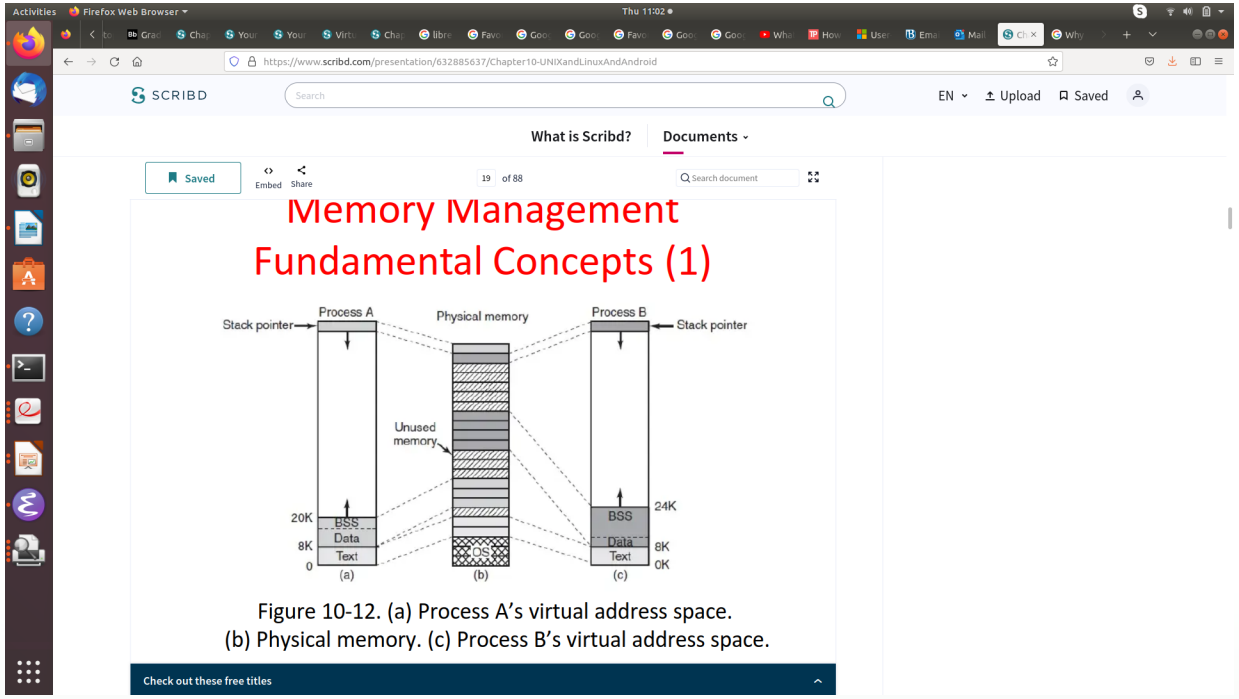

Activities
Firefox Web Browser
Thu 11:00
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
11 of 88
Search document
Process Management
System Calls in Linux (2)
while (TRUE) {
type_prompt( );
read_command(command, params);
pid = fork( );
if (pid < 0) {
printf("Unable to fork0);
continue;
}
if (pid != 0) {
waitpid (–1, &status, 0);
} else {
execve(command, params, 0);
}
}
/* repeat forever */
/* display prompt on the screen */
/* read input line from keyboard */
/* fork off a child process */
/* error condition */
/* repeat the loop */
/* parent waits for child */
/* child does the work */
Figure 10-7. A highly simplified shell.
Check out these free titles
LINUX FINAL NOTES-converted.pdf
Fayaz Ahmed
No ratings yet
midterm exam BS SYSTEM ADMINISTRATION
Ash Lee
No ratings yet
Rt Linux
KevinXavier
No ratings yet
Linux Programming by Example
Mohit Modi
Simply Linux: Basics
John E. Meister, Jr

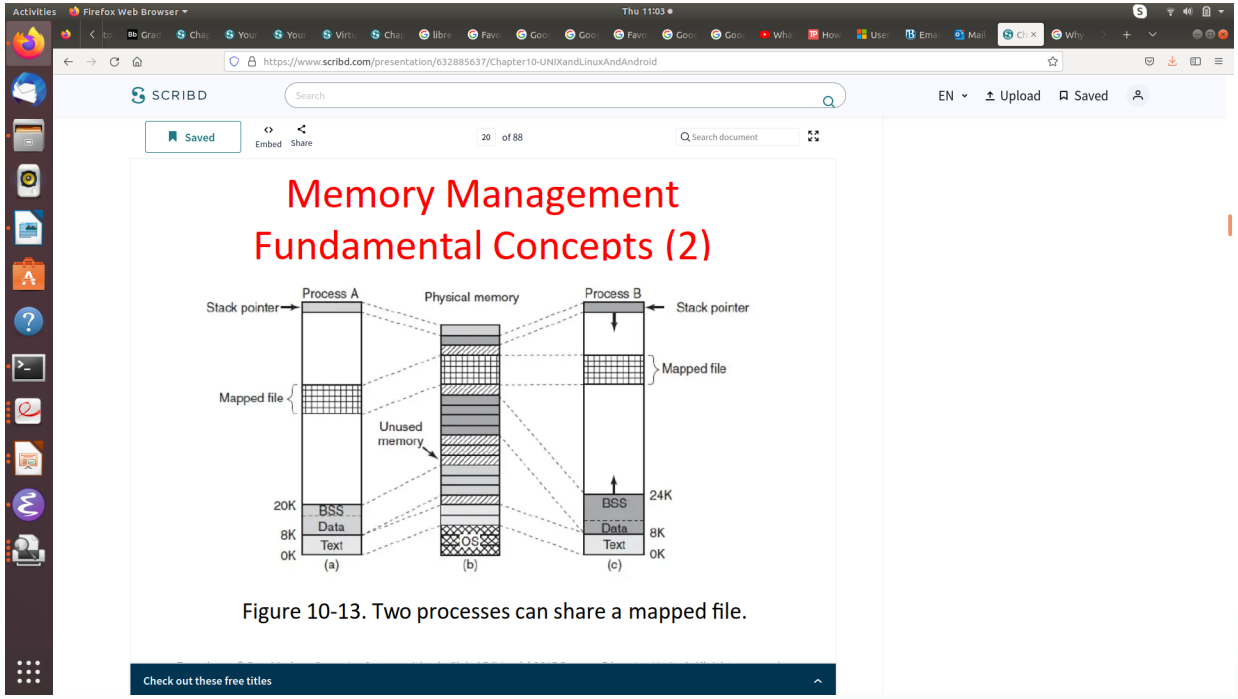

Activities
Firefox Web Browser
Thu 11:01
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
12
of 88
Search document
Implementation of Processes and Threads in Linux (1)
Process descriptors
1. Scheduling parameters
2. Memory image
3. Signals
4. Machine registers
5. System call state
6. File descriptor table
7. Accounting
8. Kernel stack
9. Miscellaneous
Check out these free titles
Linux 5 Day Introduction Course
Stephen Edwards
No ratings yet
Linux: A Comprehensive Guide to Linux Operating…
Sam Griffin
No ratings yet
LPI Linux Certification Questions: LPI Linux…
Equity Press
3.5/5 (6)
Linux: A complete guide to Linux command line for…
Arthur James
No ratings yet

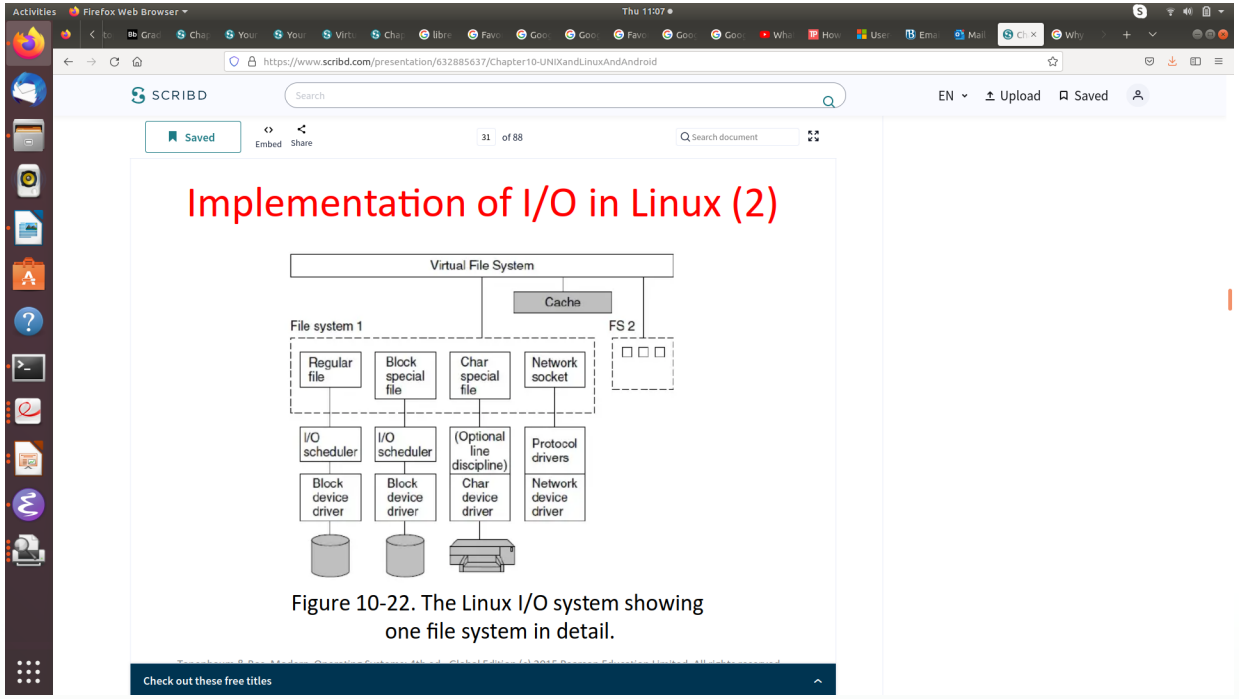


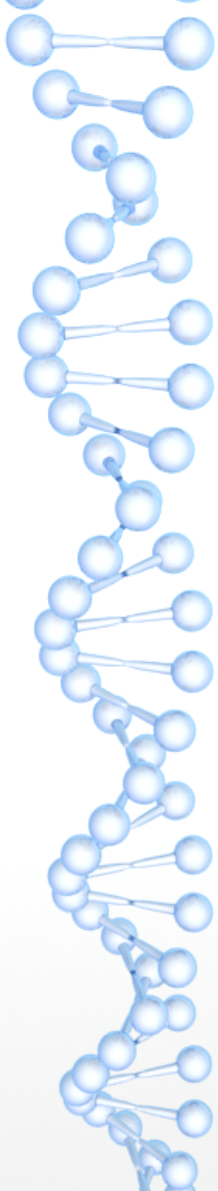
# A few slides from intervening sections

- 10.4 Memory Management in Linux
- 10.5 Input/Output in Linux


Activities
Firefox Web Browser
Thu 11:02
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
What is Scribd?
Documents
Saved
Embed
Share
19 of 88
Search document
Memory Management
Fundamental Concepts (1)
Stack pointer
Process A
Physical memory
Process B
Stack pointer
Unused memory
20K
BSS
Data
Text
8K
0
(a)
OS
(b)
BSS
Data
Text
24K
8K
0K
(c)
Figure 10-12. (a) Process A's virtual address space.
(b) Physical memory. (c) Process B's virtual address space.
Check out these free titles

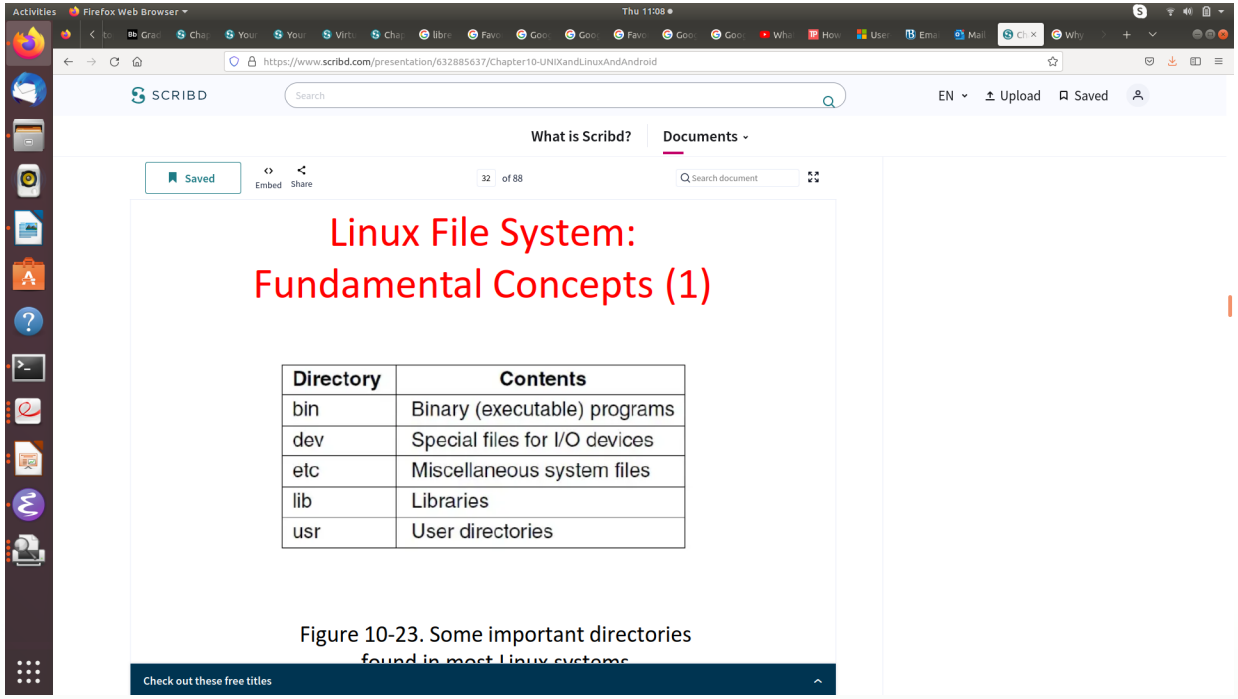

Activities
Firefox Web Browser
Thu 11:03
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
20 of 88
Search document
Memory Management
Fundamental Concepts (2)
Process A
Physical memory
Process B
Stack pointer
Stack pointer
Mapped file
Mapped file
Unused memory
20K
8K
0K
BSS
Data
Text
OS
24K
BSS
Data
8K
Text
0K
(a)
(b)
(c)
Figure 10-13. Two processes can share a mapped file.
Check out these free titles

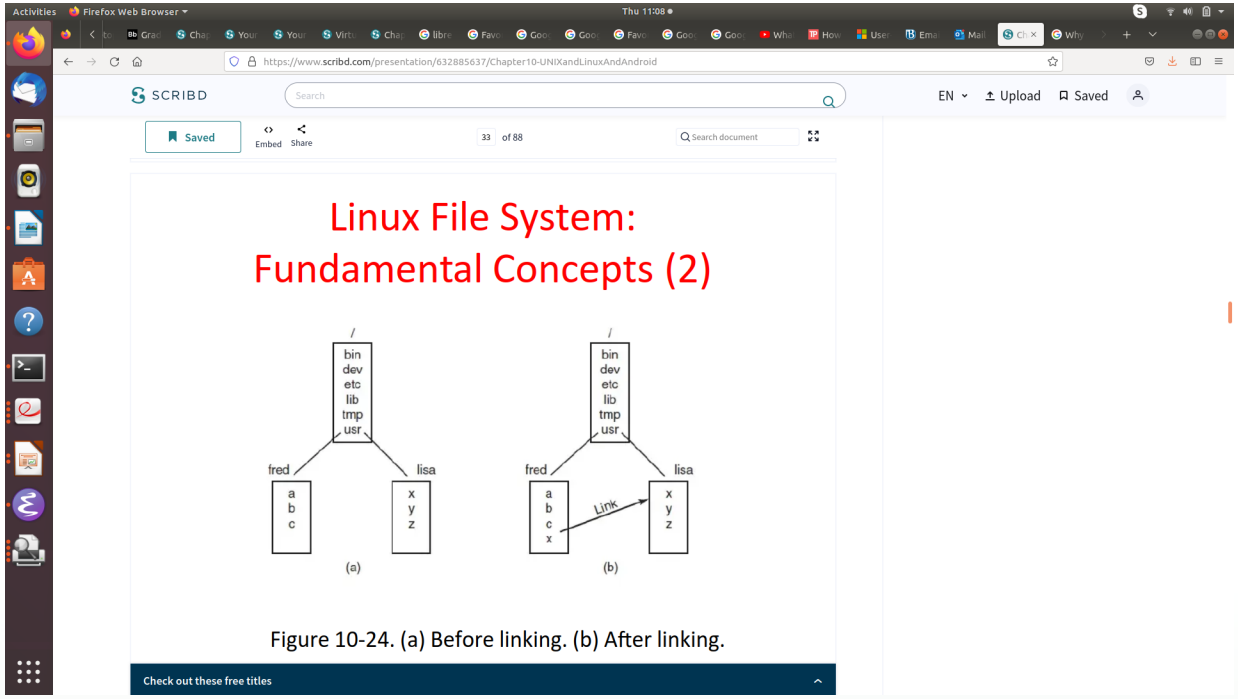

Activities
Firefox Web Browser
Thu 11:07
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
31 of 88
Search document
Implementation of I/O in Linux (2)
Virtual File System
Cache
File system 1
FS 2
Regular file
Block special file
Char special file
Network socket
I/O scheduler
I/O scheduler
(Optional line discipline)
Protocol drivers
Block device driver
Block device driver
Char device driver
Network device driver
Figure 10-22. The Linux I/O system showing one file system in detail.
Check out these free titles

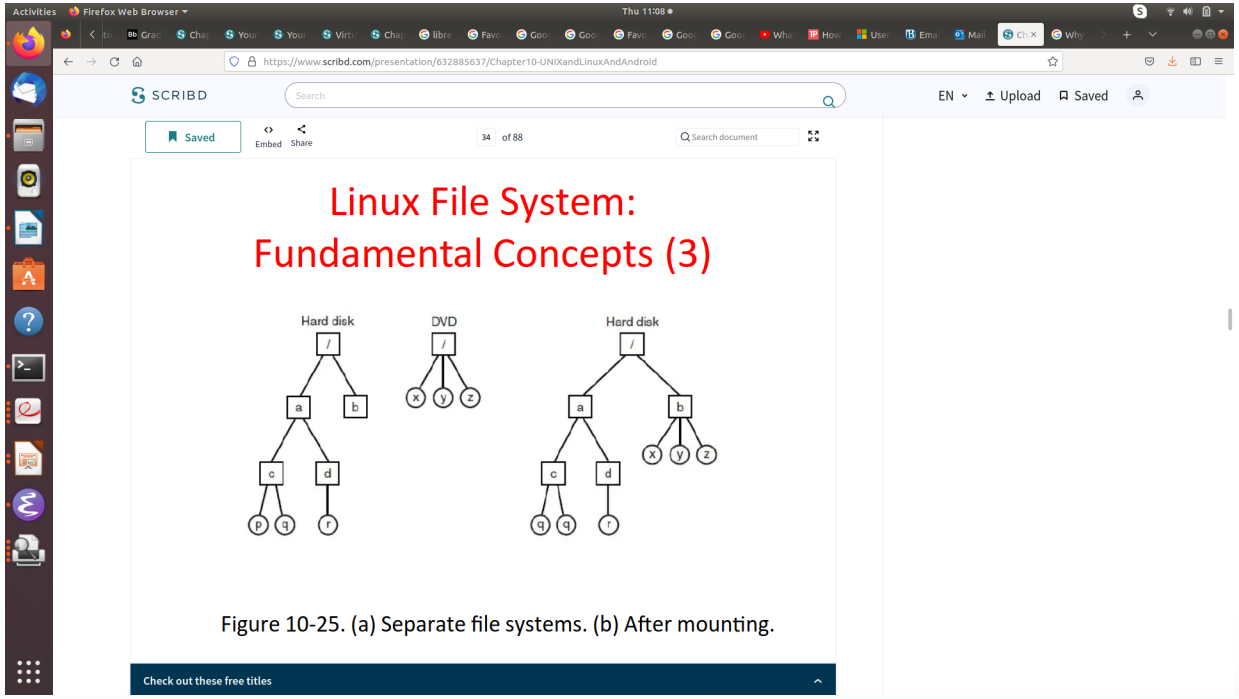


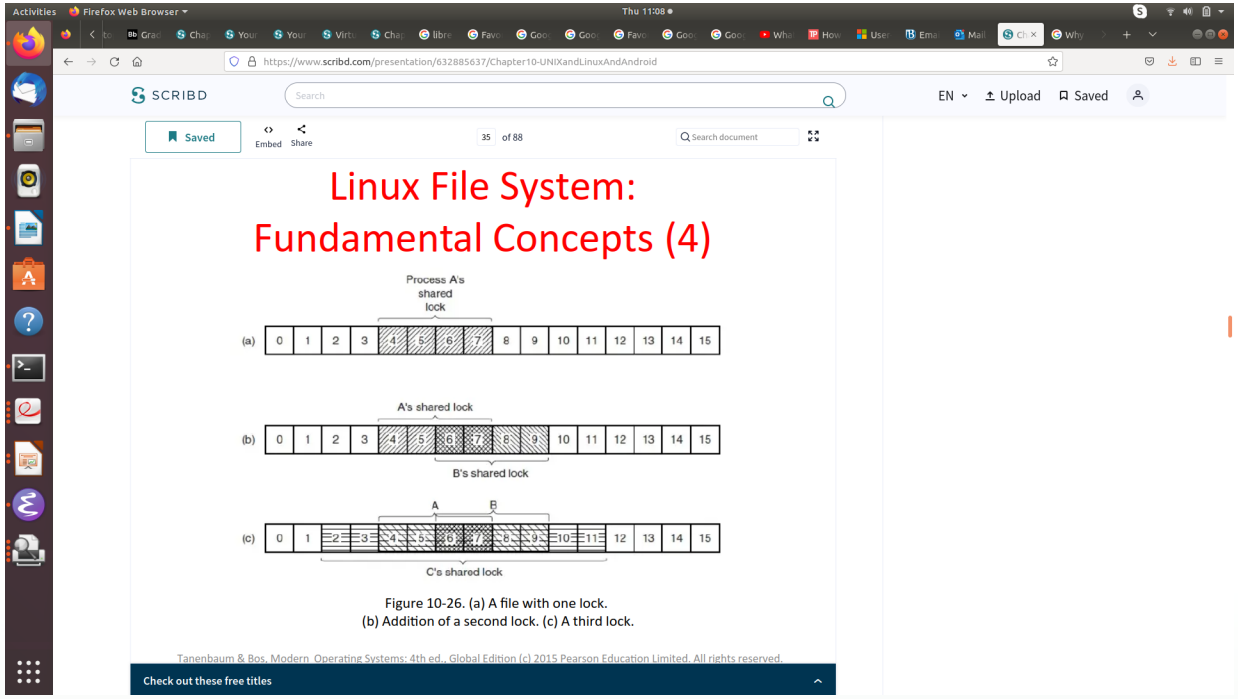
# 10.6 The Linux File System

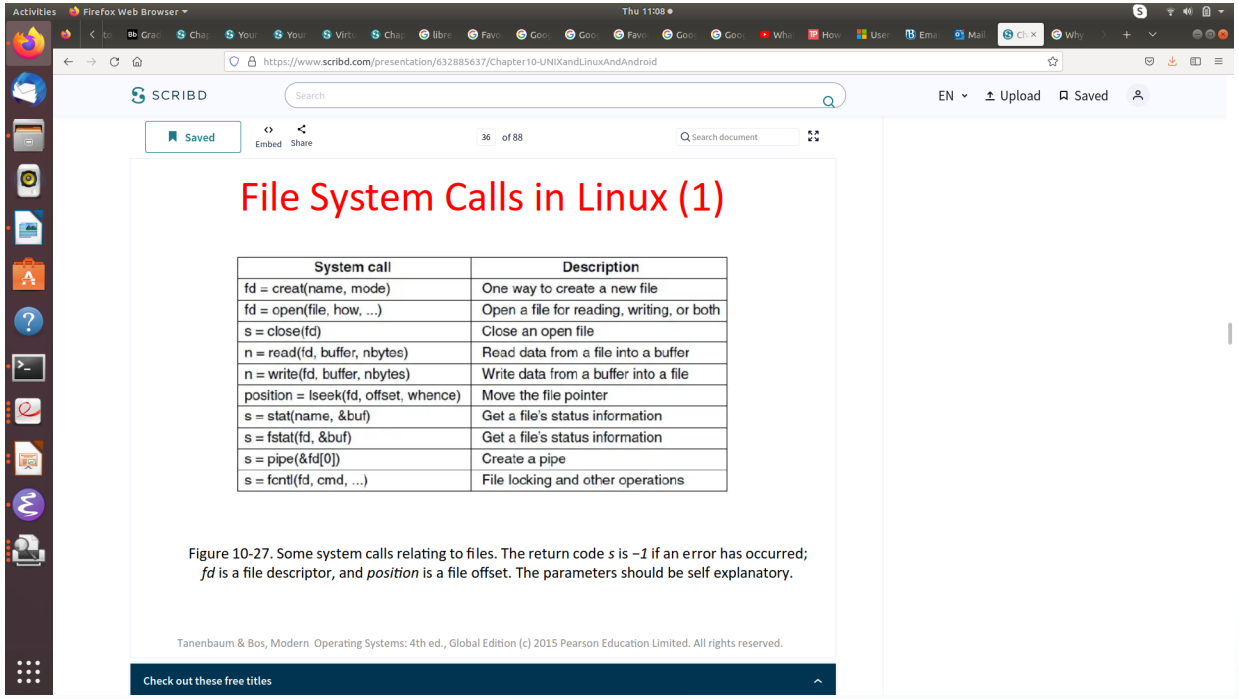
- Fundamental Concepts
- File System Calls
- Implementation

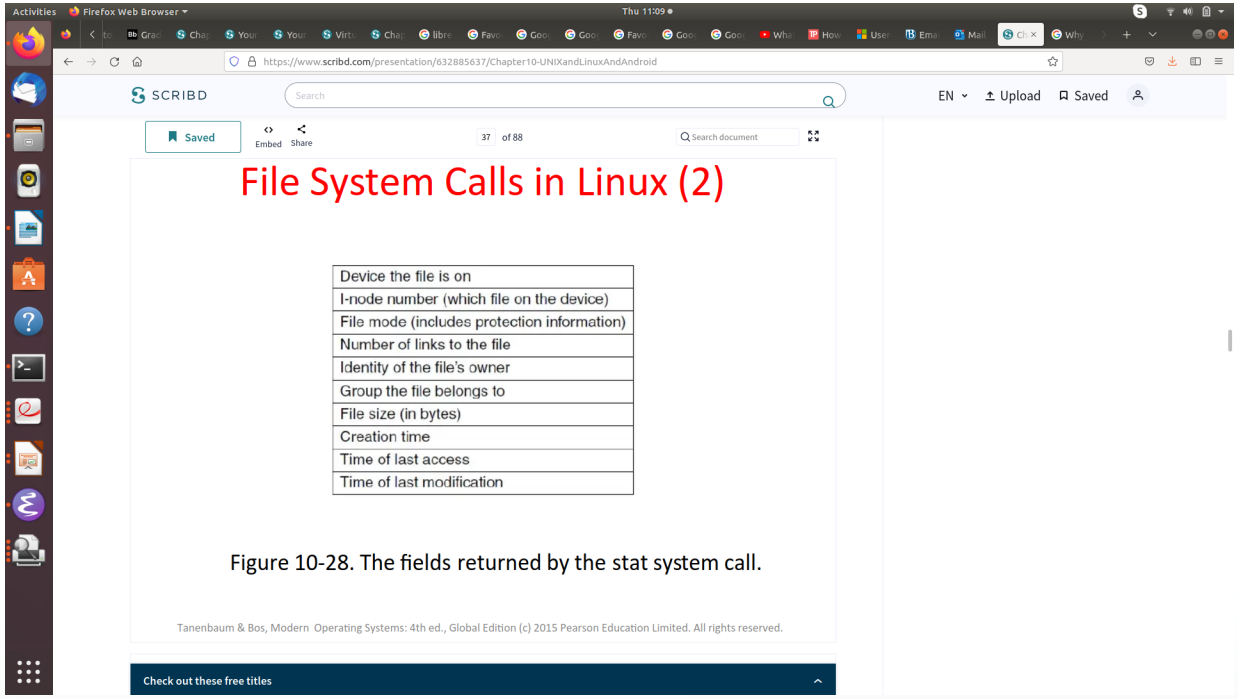

Activities
Firefox Web Browser
Thu 11:08
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
What is Scribd?
Documents
Saved
Embed
Share
32 of 88
Search document
Linux File System:
Fundamental Concepts (1)
Directory
Contents
bin
Binary (executable) programs
dev
Special files for I/O devices
etc
Miscellaneous system files
lib
Libraries
usr
User directories
Figure 10-23. Some important directories
found in most Linux systems
Check out these free titles

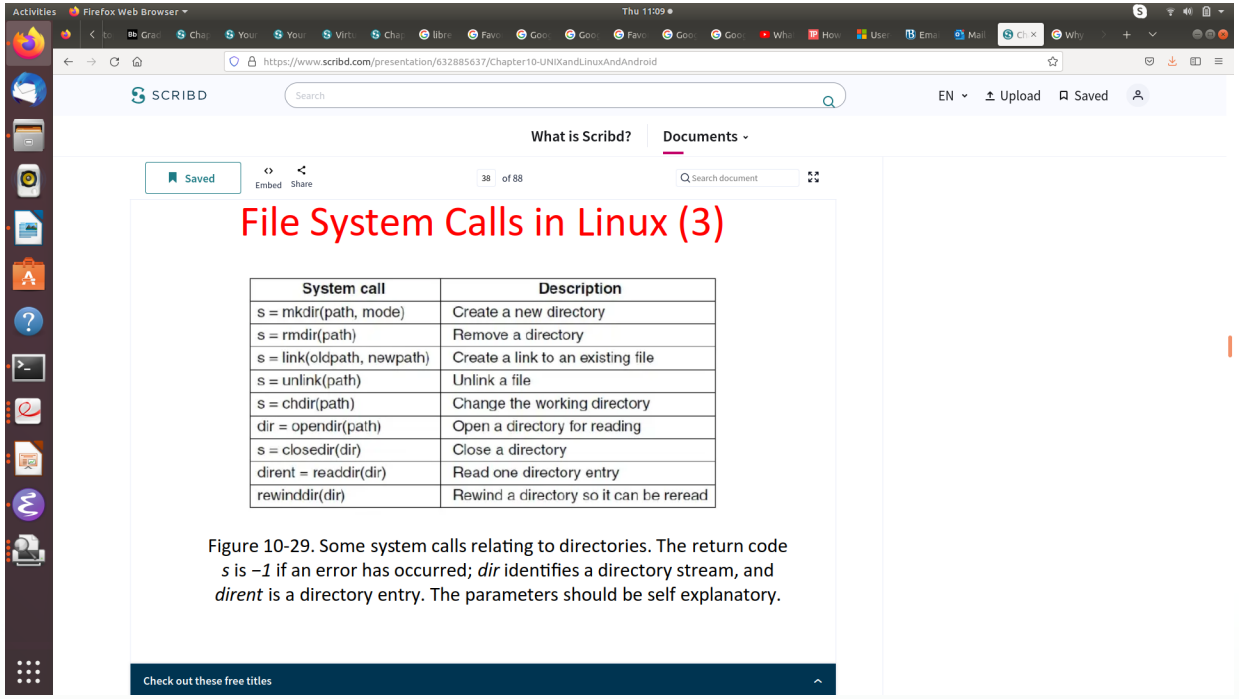

Activities
Firefox Web Browser
Thu 11:08
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
33 of 88
Search document
Linux File System:
Fundamental Concepts (2)
/
bin
dev
etc
lib
tmp
usr
fred
lisa
a
b
c
x
y
z
(a)
Link
(b)
Figure 10-24. (a) Before linking. (b) After linking.
Check out these free titles

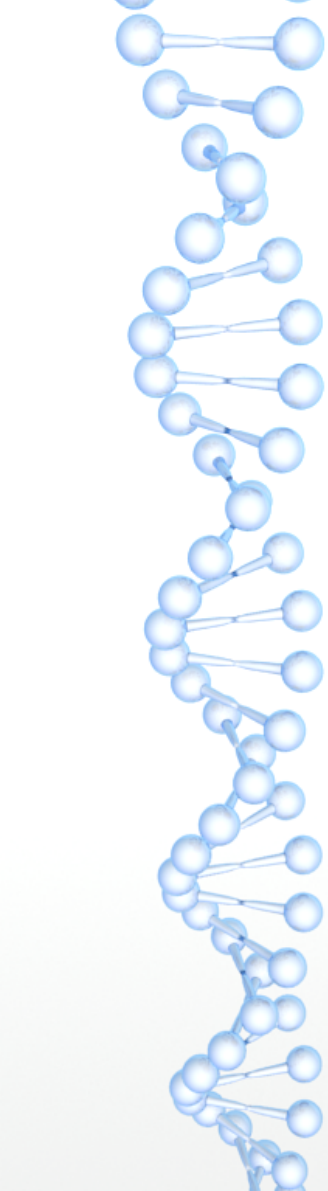

Activities
Firefox Web Browser
Thu 11:08
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
34 of 88
Search document
Linux File System:
Fundamental Concepts (3)
Hard disk
DVD
Hard disk
Figure 10-25. (a) Separate file systems. (b) After mounting.
Check out these free titles


Activities
Firefox Web Browser
Thu 11:08
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
35 of 88
Search document
Linux File System:
Fundamental Concepts (4)
Process A's shared lock
(a)
A's shared lock
(b)
B's shared lock
A
B
(c)
C's shared lock
Figure 10-26. (a) A file with one lock.
(b) Addition of a second lock. (c) A third lock.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles


Activities
Firefox Web Browser
Thu 11:08
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
36 of 88
Search document
File System Calls in Linux (1)
System call | Description
fd = creat(name, mode) | One way to create a new file
fd = open(file, how, ...) | Open a file for reading, writing, or both
s = close(fd) | Close an open file
n = read(fd, buffer, nbytes) | Read data from a file into a buffer
n = write(fd, buffer, nbytes) | Write data from a buffer into a file
position = lseek(fd, offset, whence) | Move the file pointer
s = stat(name, &buf) | Get a file's status information
s = fstat(fd, &buf) | Get a file's status information
s = pipe(&fd[0]) | Create a pipe
s = fcntl(fd, cmd, ...) | File locking and other operations
Figure 10-27. Some system calls relating to files. The return code s is −1 if an error has occurred; fd is a file descriptor, and position is a file offset. The parameters should be self explanatory.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

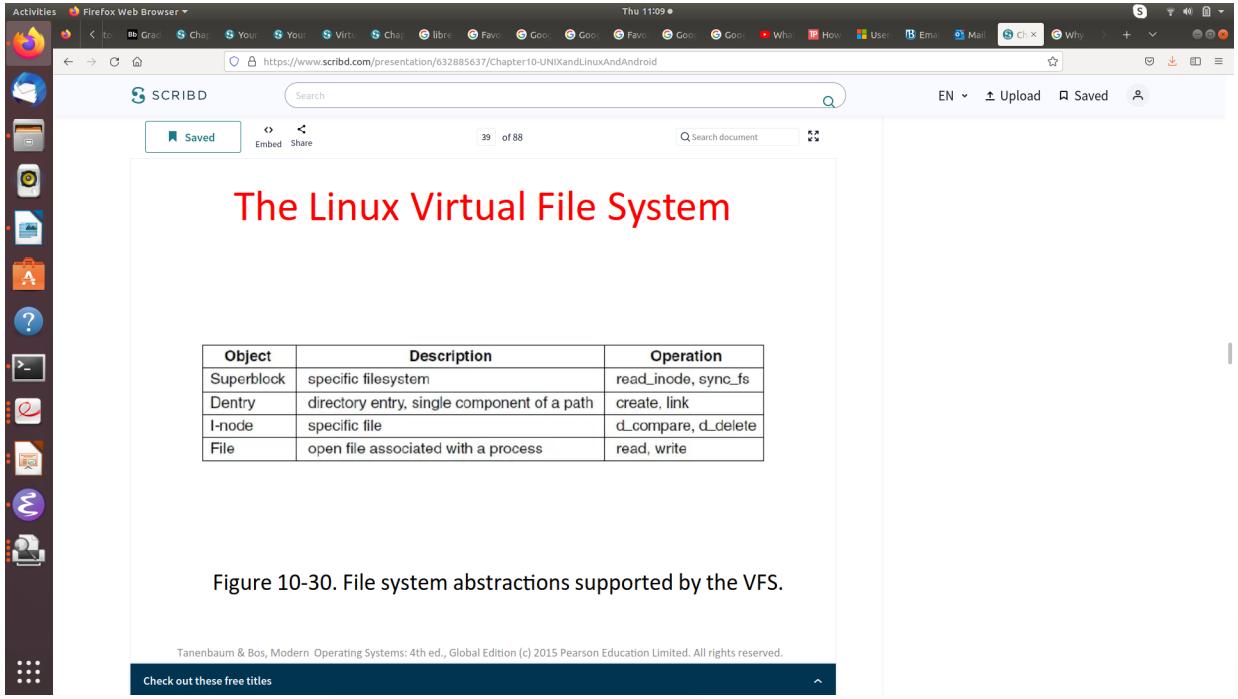

Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
37 of 88
Search document
File System Calls in Linux (2)
Device the file is on
I-node number (which file on the device)
File mode (includes protection information)
Number of links to the file
Identity of the file's owner
Group the file belongs to
File size (in bytes)
Creation time
Time of last access
Time of last modification
Figure 10-28. The fields returned by the stat system call.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

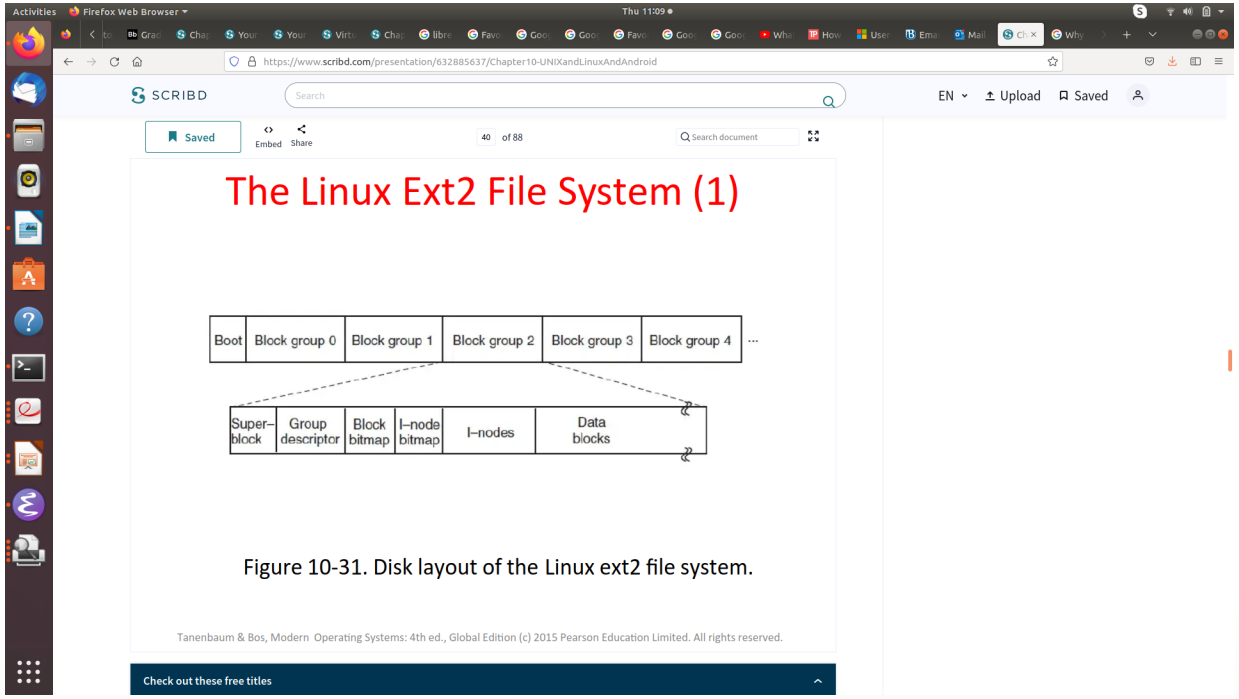

Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
What is Scribd?
Documents
Saved
Embed
Share
38 of 88
Search document
File System Calls in Linux (3)
System call | Description
s = mkdir(path, mode) | Create a new directory
s = rmdir(path) | Remove a directory
s = link(oldpath, newpath) | Create a link to an existing file
s = unlink(path) | Unlink a file
s = chdir(path) | Change the working directory
dir = opendir(path) | Open a directory for reading
s = closedir(dir) | Close a directory
dirent = readdir(dir) | Read one directory entry
rewinddir(dir) | Rewind a directory so it can be reread
Figure 10-29. Some system calls relating to directories. The return code s is –1 if an error has occurred; dir identifies a directory stream, and dirent is a directory entry. The parameters should be self explanatory.
Check out these free titles

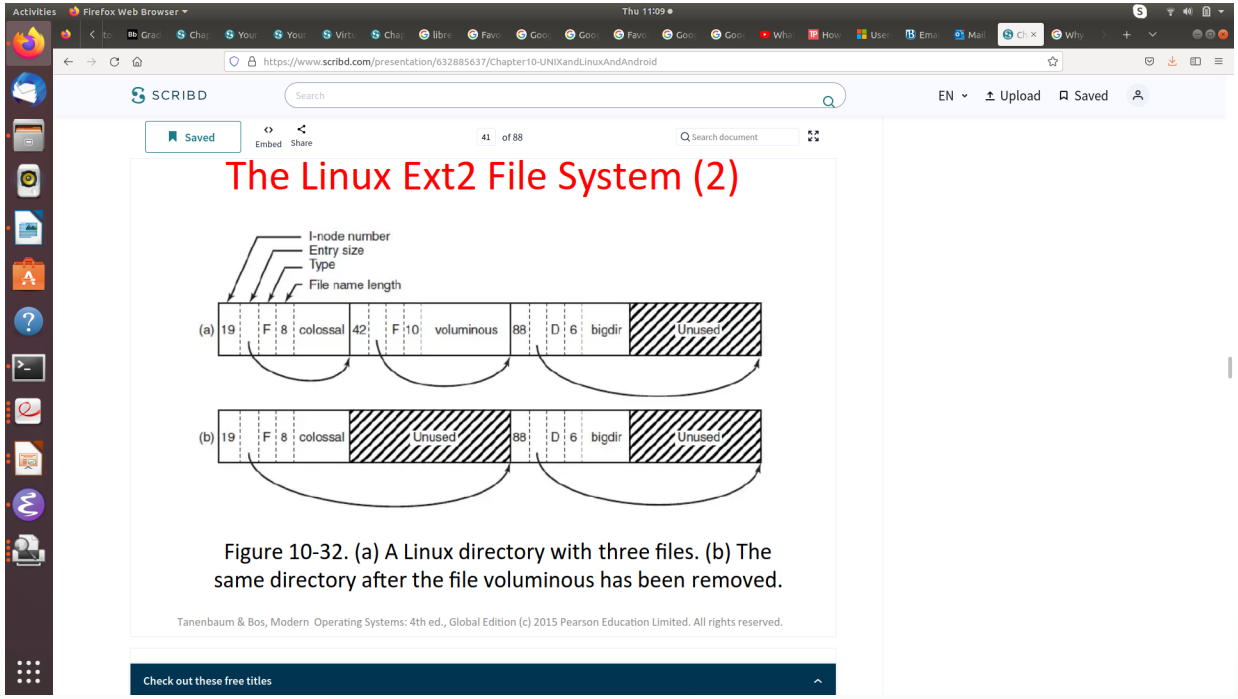


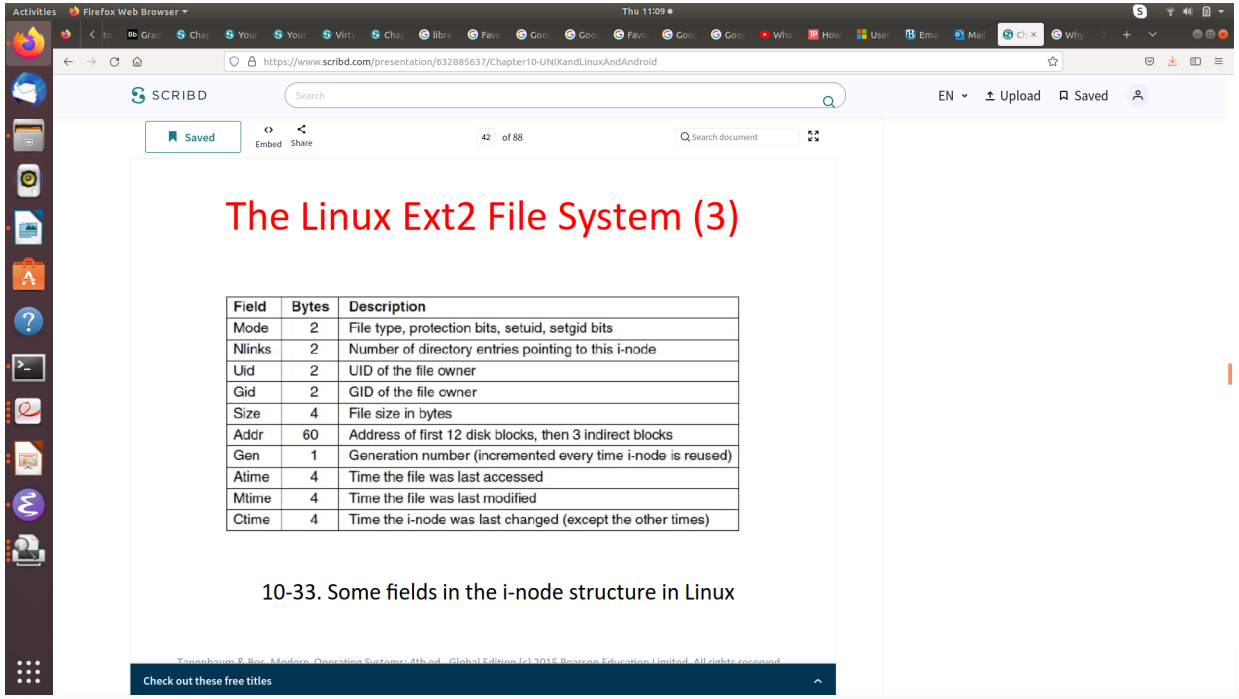
# 10.6.3 Implementation

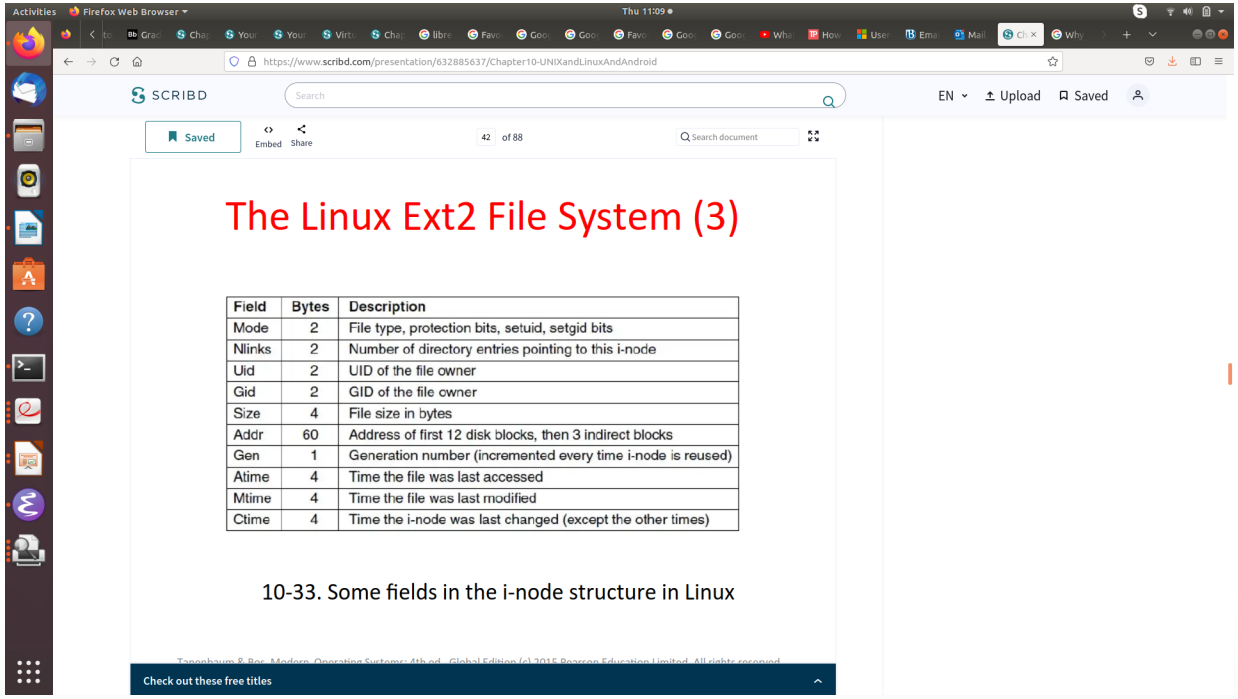
- Linux Virtual File System
- Earlier Ext2
- Ext3 and Ext4 are journaling file system

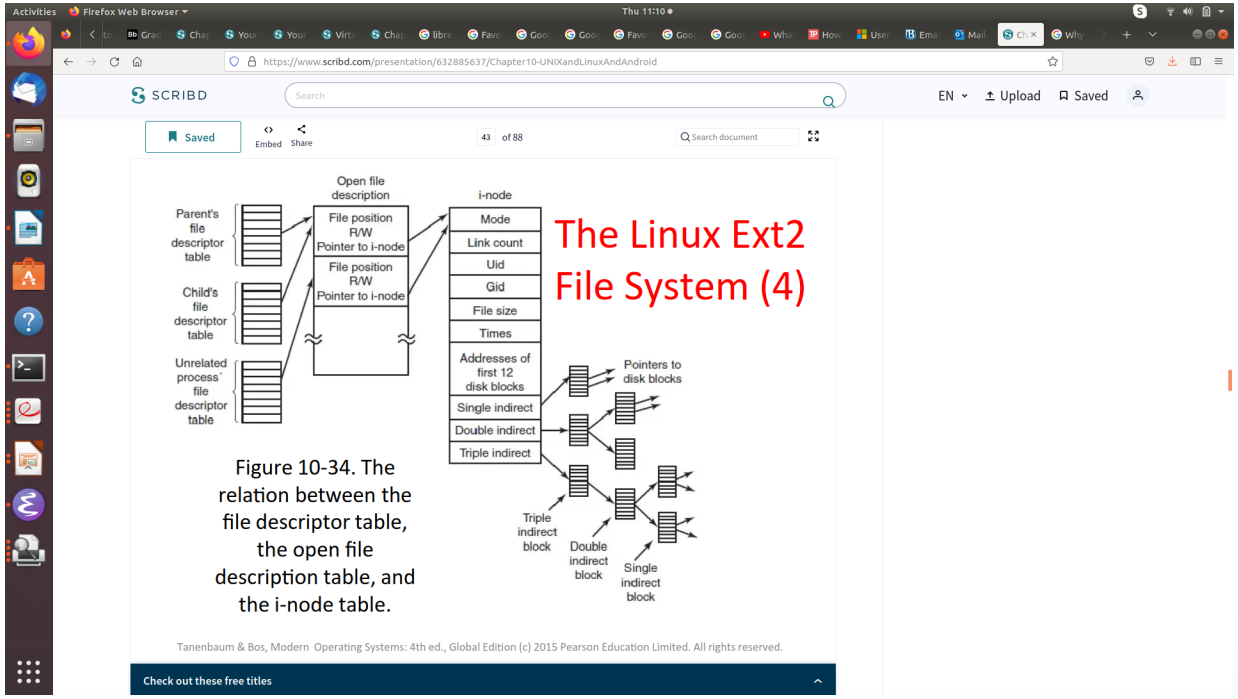

Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
39 of 88
Search document
The Linux Virtual File System
Object | Description | Operation
Superblock | specific filesystem | read_inode, sync_fs
Dentry | directory entry, single component of a path | create, link
I-node | specific file | d_compare, d_delete
File | open file associated with a process | read, write
Figure 10-30. File system abstractions supported by the VFS.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles


Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
40
of 88
Search document
The Linux Ext2 File System (1)
Boot
Block group 0
Block group 1
Block group 2
Block group 3
Block group 4
Super-block
Group descriptor
Block bitmap
I-node bitmap
I-nodes
Data blocks
Figure 10-31. Disk layout of the Linux ext2 file system.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles


Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
41 of 88
Search document
The Linux Ext2 File System (2)
I-node number
Entry size
Type
File name length
(a) 19 F 8 colossal 42 F 10 voluminous 88 D 6 bigdir Unused
(b) 19 F 8 colossal Unused 88 D 6 bigdir Unused
Figure 10-32. (a) A Linux directory with three files. (b) The same directory after the file voluminous has been removed.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

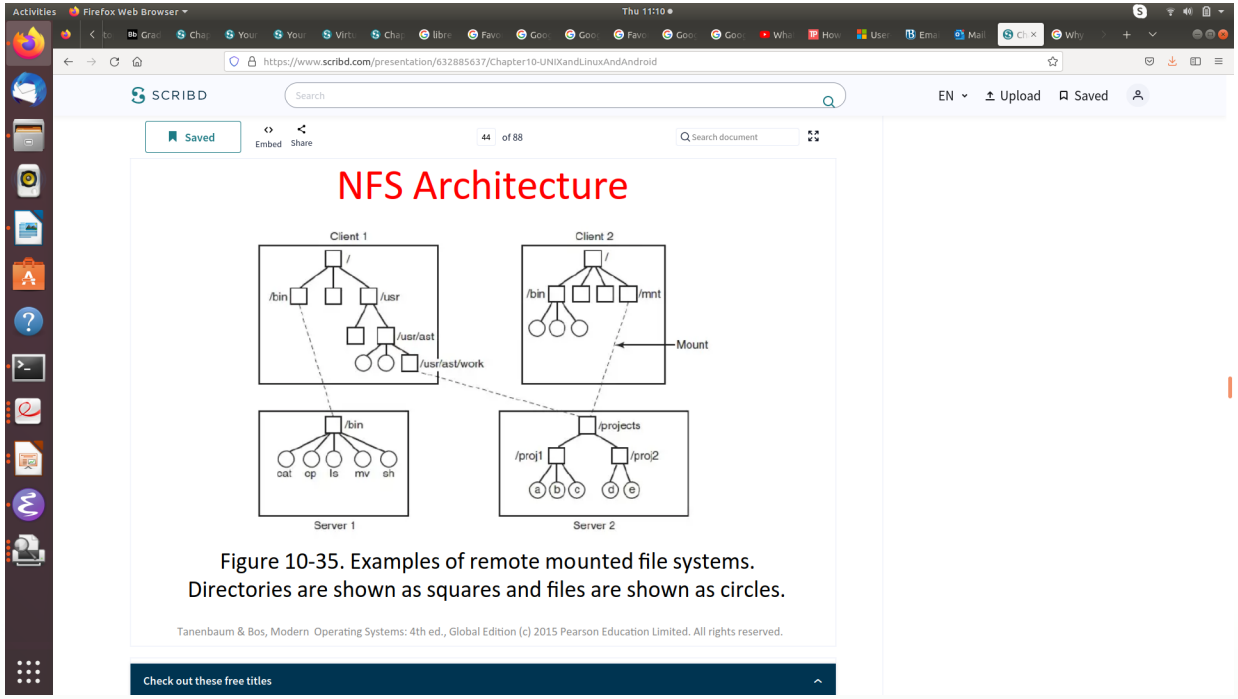

Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
42 of 88
Search document
The Linux Ext2 File System (3)
Field | Bytes | Description
Mode | 2 | File type, protection bits, setuid, setgid bits
Nlinks | 2 | Number of directory entries pointing to this i-node
Uid | 2 | UID of the file owner
Gid | 2 | GID of the file owner
Size | 4 | File size in bytes
Addr | 60 | Address of first 12 disk blocks, then 3 indirect blocks
Gen | 1 | Generation number (incremented every time i-node is reused)
Atime | 4 | Time the file was last accessed
Mtime | 4 | Time the file was last modified
Ctime | 4 | Time the i-node was last changed (except the other times)
10-33. Some fields in the i-node structure in Linux
Check out these free titles

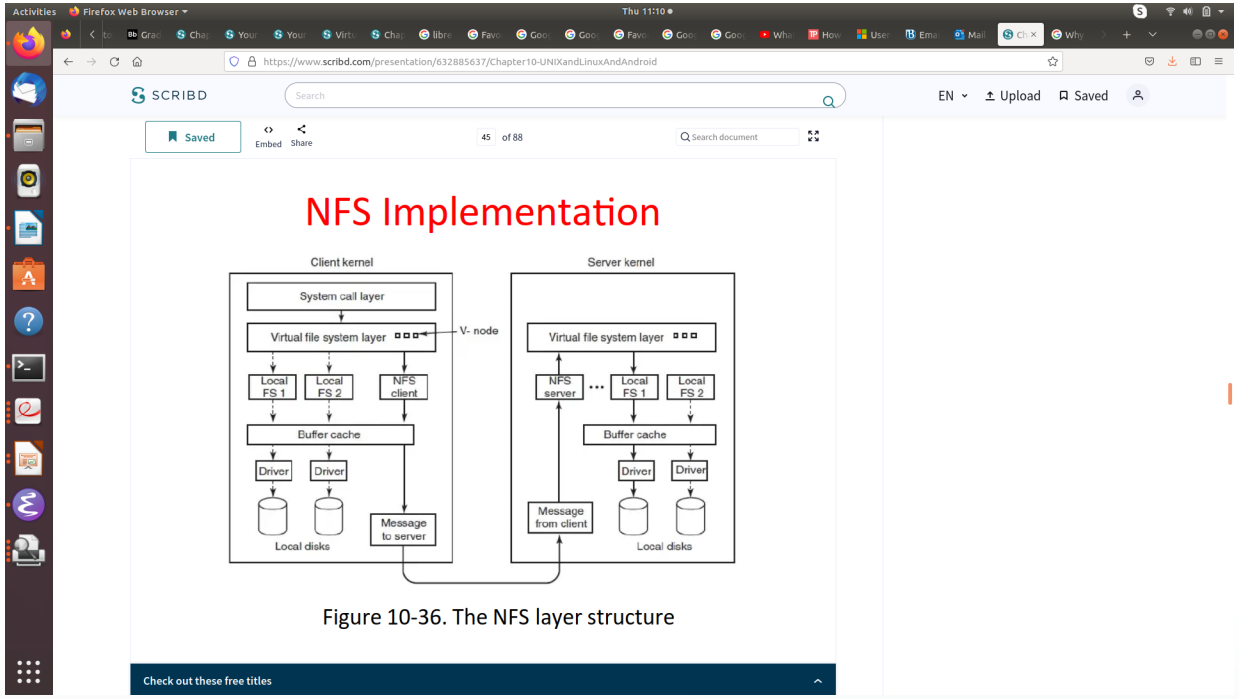

Activities
Firefox Web Browser
Thu 11:09
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
42 of 88
Search document
The Linux Ext2 File System (3)
Field | Bytes | Description
Mode | 2 | File type, protection bits, setuid, setgid bits
Nlinks | 2 | Number of directory entries pointing to this i-node
Uid | 2 | UID of the file owner
Gid | 2 | GID of the file owner
Size | 4 | File size in bytes
Addr | 60 | Address of first 12 disk blocks, then 3 indirect blocks
Gen | 1 | Generation number (incremented every time i-node is reused)
Atime | 4 | Time the file was last accessed
Mtime | 4 | Time the file was last modified
Ctime | 4 | Time the i-node was last changed (except the other times)
10-33. Some fields in the i-node structure in Linux
Check out these free titles

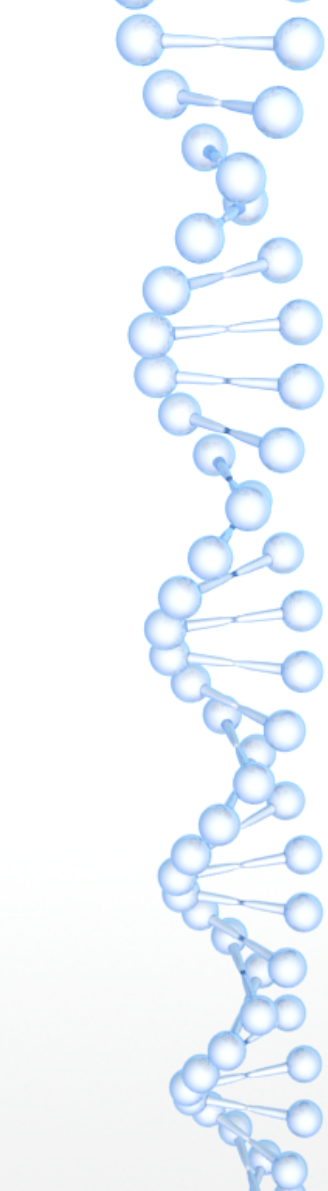

Activities
Firefox Web Browser
Thu 11:10
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
43 of 88
Search document
Open file description
i-node
Parent's file descriptor table
Child's file descriptor table
Unrelated process' file descriptor table
File position R/W Pointer to i-node
File position R/W Pointer to i-node
Mode
Link count
Uid
Gid
File size
Times
Addresses of first 12 disk blocks
Single indirect
Double indirect
Triple indirect
Pointers to disk blocks
Triple indirect block
Double indirect block
Single indirect block
The Linux Ext2 File System (4)
Figure 10-34. The relation between the file descriptor table, the open file description table, and the i-node table.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

# 10.6.4 NFS: The Network File System

- Architecture
- Protocols
- Implementation
- NFS Version 4 is stateful – permitting better integration of remote files and works better with Linux and Windows file system semantics

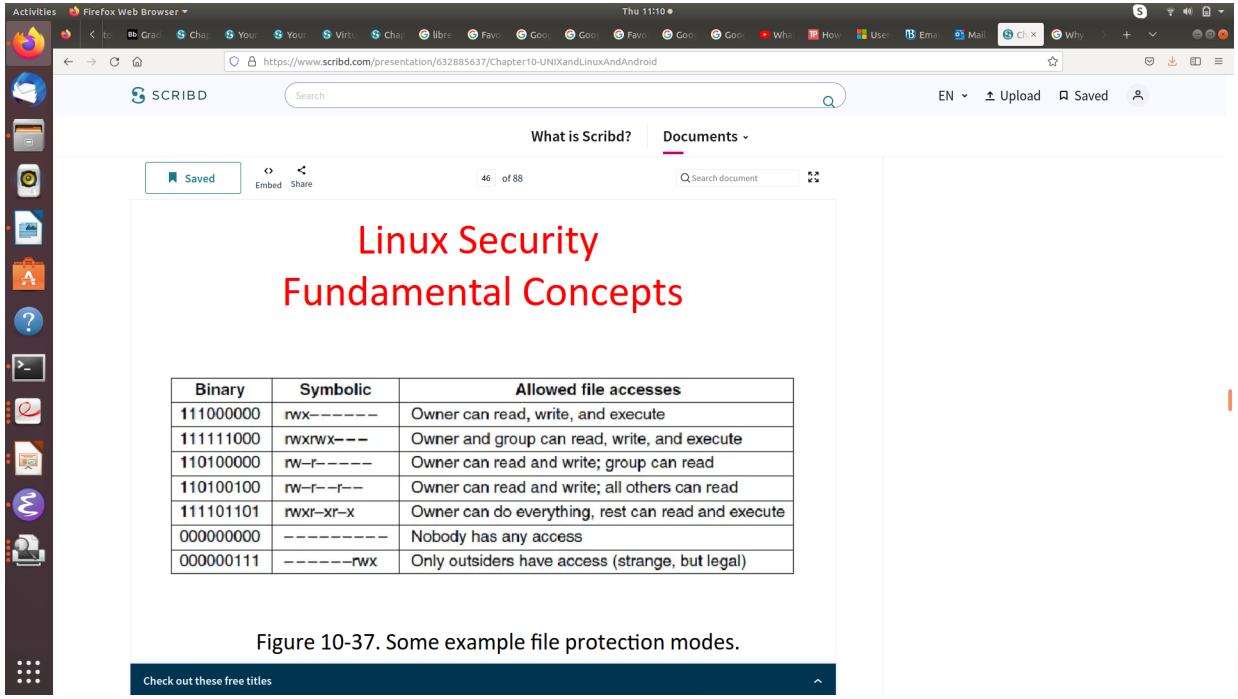

Activities
Firefox Web Browser
Thu 11:10
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
44 of 88
Search document
NFS Architecture
Client 1
Client 2
/
/bin
/usr
/usr/ast
/usr/ast/work
/mnt
Mount
/bin
cat
cp
ls
mv
sh
/projects
/proj1
/proj2
a
b
c
d
e
Server 1
Server 2
Figure 10-35. Examples of remote mounted file systems.
Directories are shown as squares and files are shown as circles.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

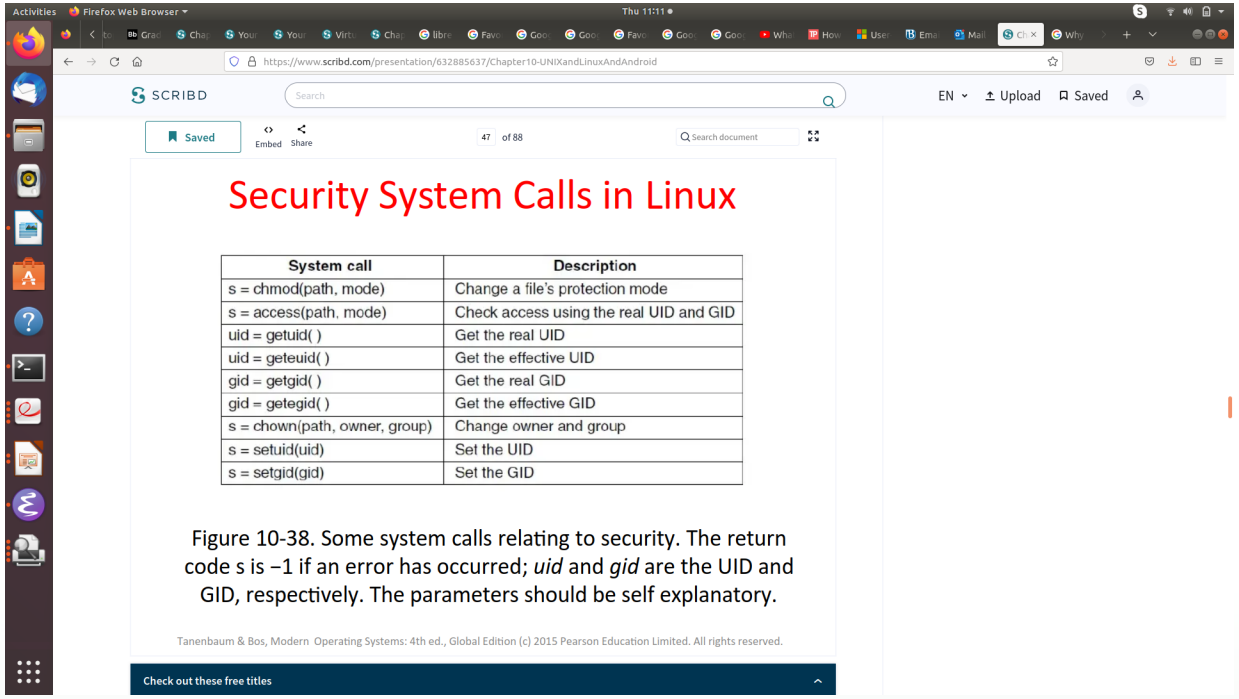

Activities
Firefox Web Browser
Thu 11:10
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
45 of 88
Search document
NFS Implementation
Client kernel
Server kernel
System call layer
Virtual file system layer
V- node
Local FS 1
Local FS 2
NFS client
Buffer cache
Driver
Driver
Local disks
Message to server
Virtual file system layer
NFS server
Local FS 1
Local FS 2
Buffer cache
Driver
Driver
Message from client
Local disks
Figure 10-36. The NFS layer structure
Check out these free titles

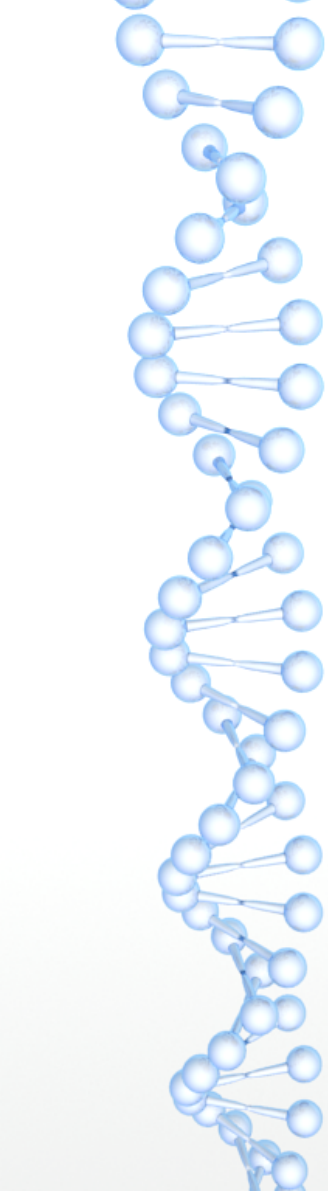


# 10.7 Security in Linux

- Fundamental Concepts
- System calls


Activities
Firefox Web Browser
Thu 11:10
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
What is Scribd?
Documents
Saved
Embed
Share
46 of 88
Search document
Linux Security
Fundamental Concepts
Binary | Symbolic | Allowed file accesses
111000000 | rwx------ | Owner can read, write, and execute
111111000 | rwxrwx--- | Owner and group can read, write, and execute
110100000 | rw-r----- | Owner can read and write; group can read
110100100 | rw-r--r-- | Owner can read and write; all others can read
111101101 | rwxr-xr-x | Owner can do everything, rest can read and execute
000000000 | ---------- | Nobody has any access
000000111 | ------rwx | Only outsiders have access (strange, but legal)
Figure 10-37. Some example file protection modes.
Check out these free titles

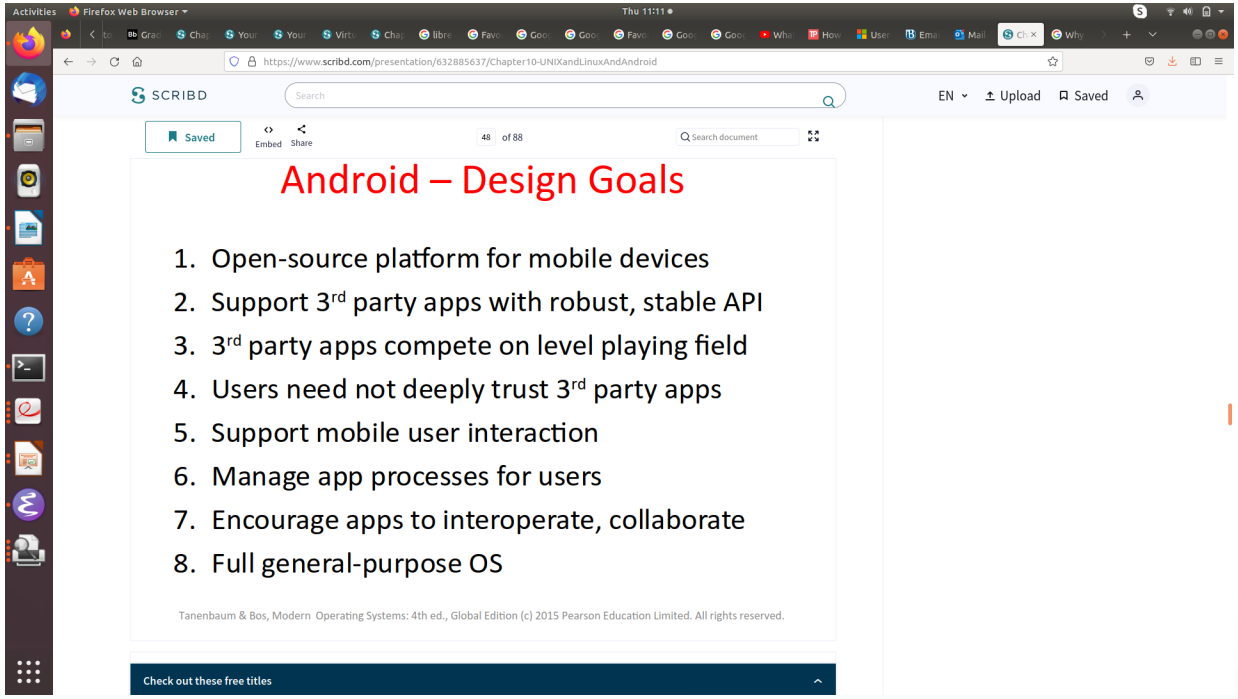

Activities
Firefox Web Browser
Thu 11:11
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
47 of 88
Search document
Security System Calls in Linux
System call | Description
s = chmod(path, mode) | Change a file's protection mode
s = access(path, mode) | Check access using the real UID and GID
uid = getuid( ) | Get the real UID
uid = geteuid( ) | Get the effective UID
gid = getgid( ) | Get the real GID
gid = getegid( ) | Get the effective GID
s = chown(path, owner, group) | Change owner and group
s = setuid(uid) | Set the UID
s = setgid(gid) | Set the GID
Figure 10-38. Some system calls relating to security. The return code s is −1 if an error has occurred; uid and gid are the UID and GID, respectively. The parameters should be self explanatory.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles



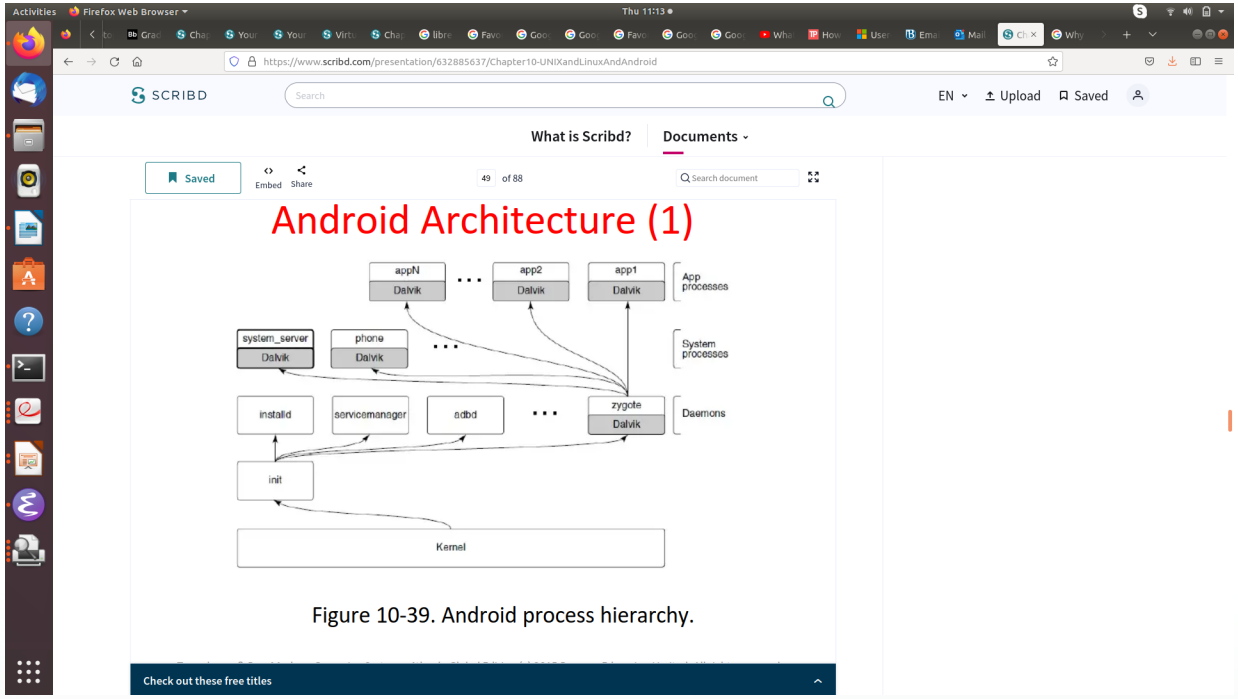
# 10.8 Android

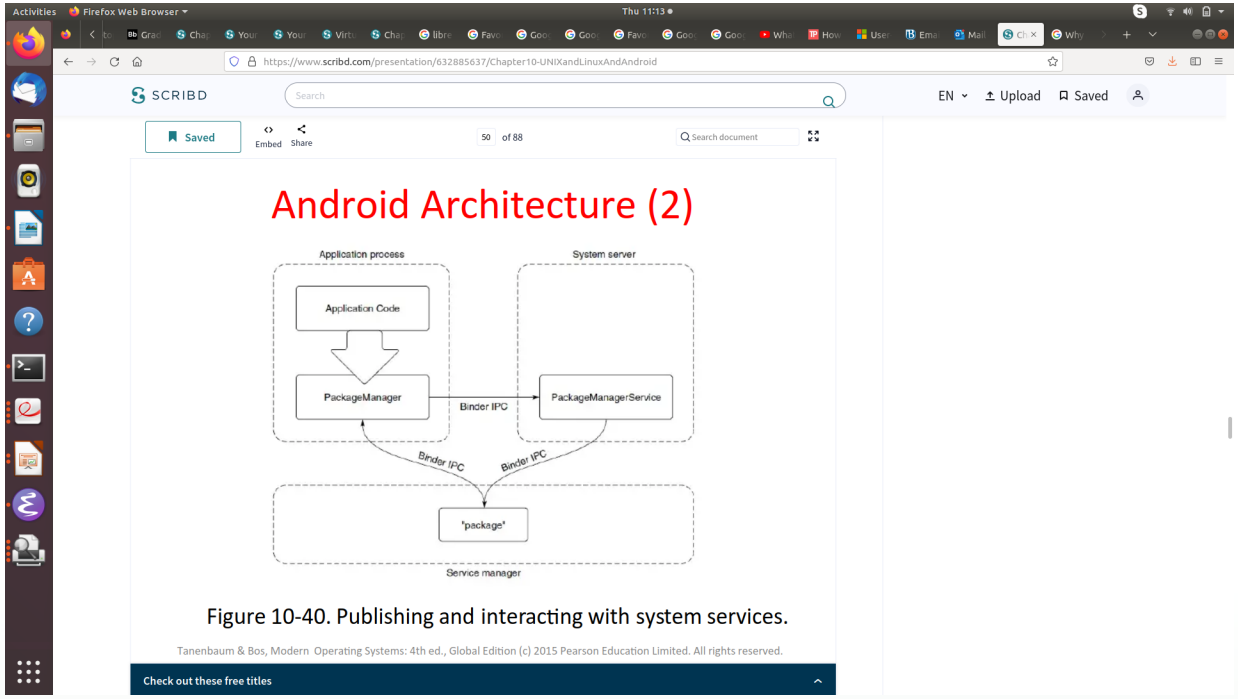
- Mobile Android
- Sits on Linux
- Manages power well
- Boots up fast

# 10.8.3 Design Goals

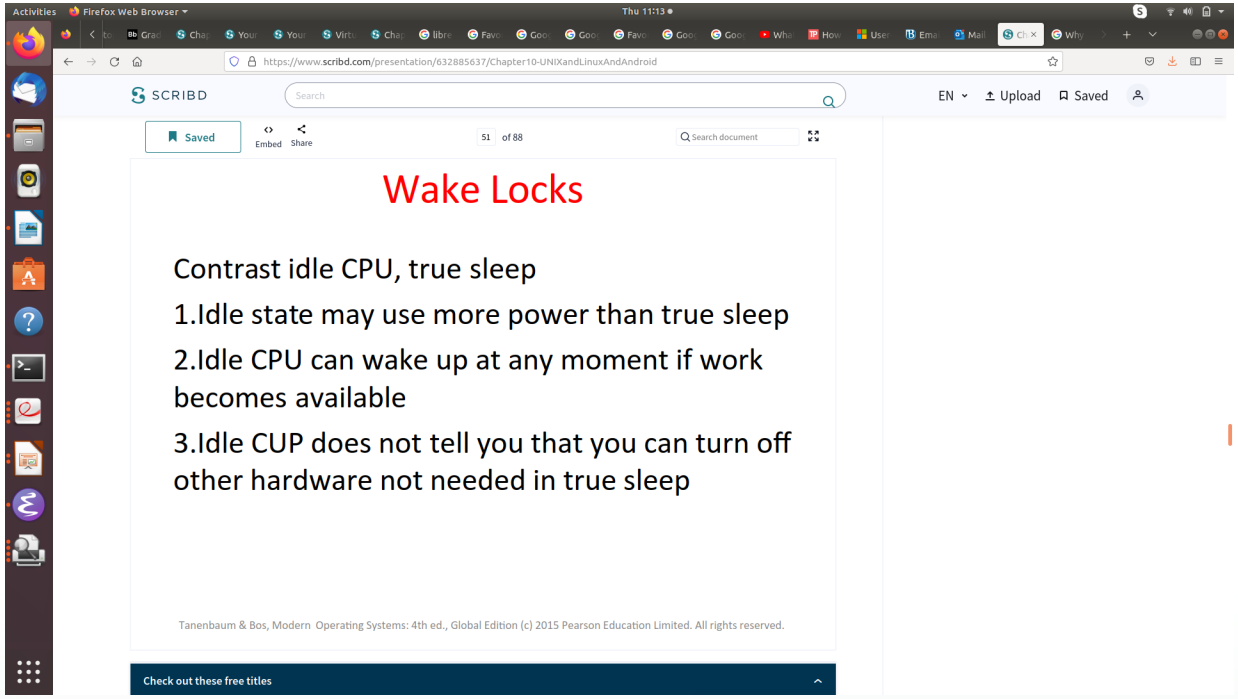


# 10.8.4 Android Architecture

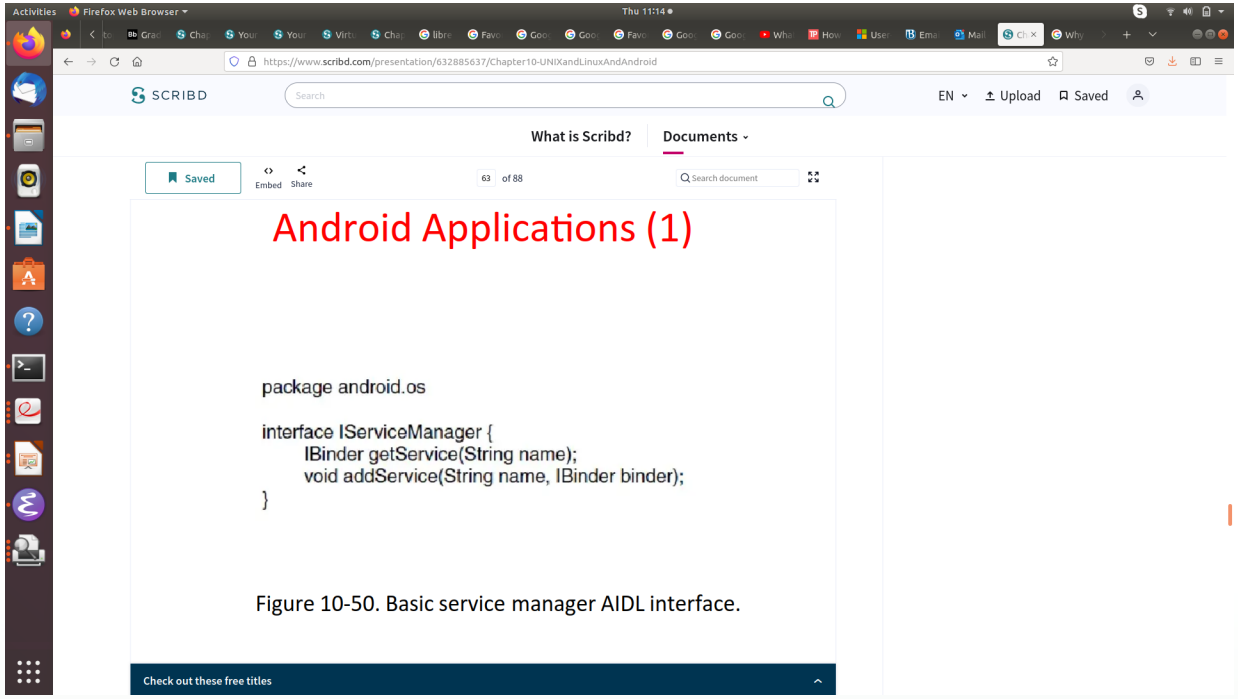


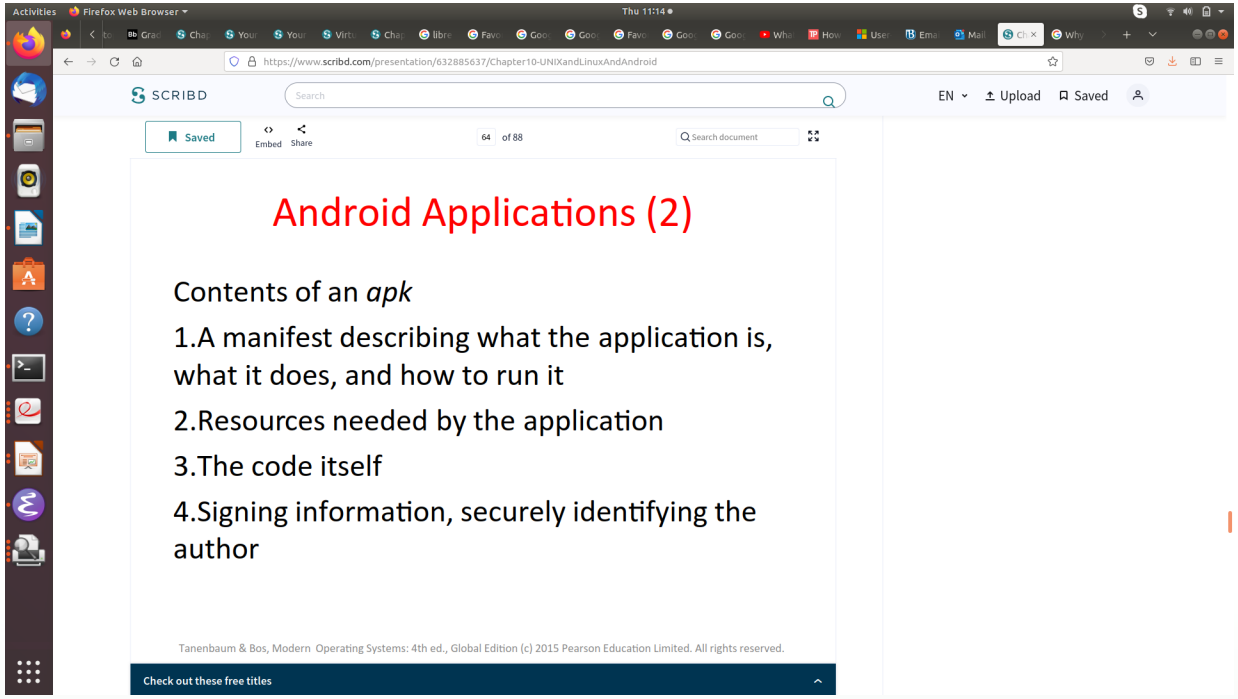

Activities
Firefox Web Browser
Thu 11:13
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
50 of 88
Search document
Android Architecture (2)
Application process
System server
Application Code
PackageManager
Binder IPC
PackageManagerService
Binder IPC
Binder IPC
'package'
Service manager
Figure 10-40. Publishing and interacting with system services.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

# 10.8.5 Linux extensions to stock Linux kernel Wake Locks manage how system sleeps

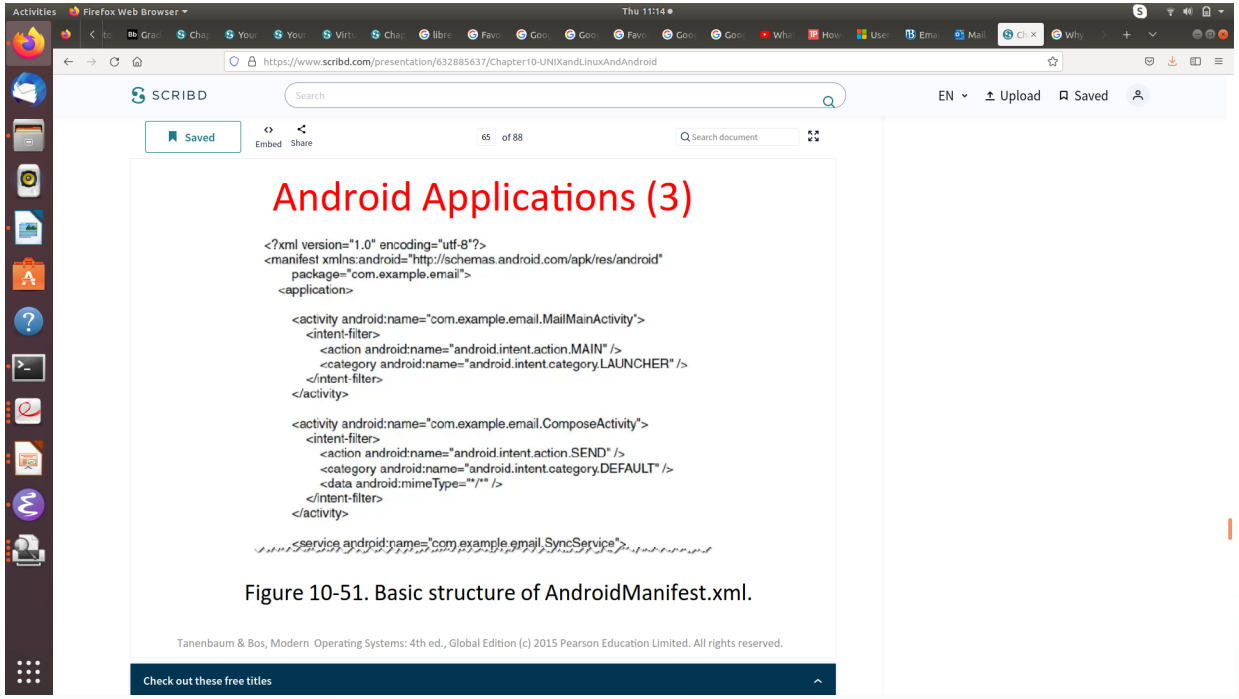


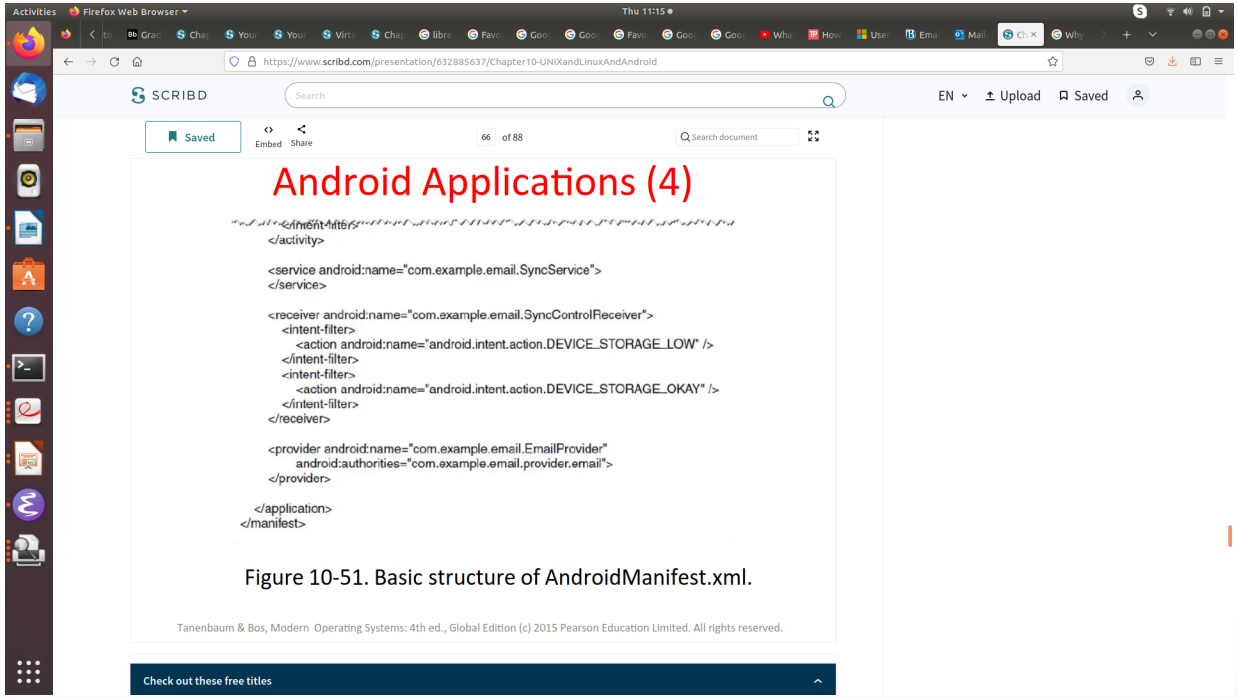
# 10.8.8 Android Applications

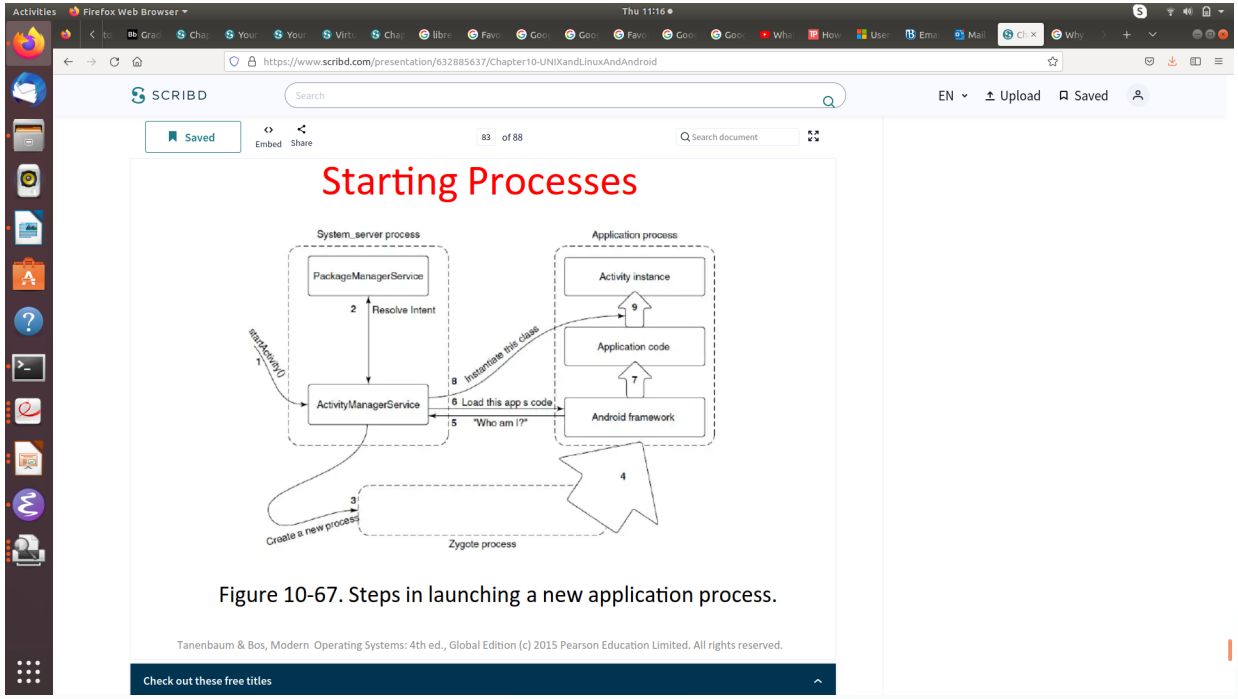


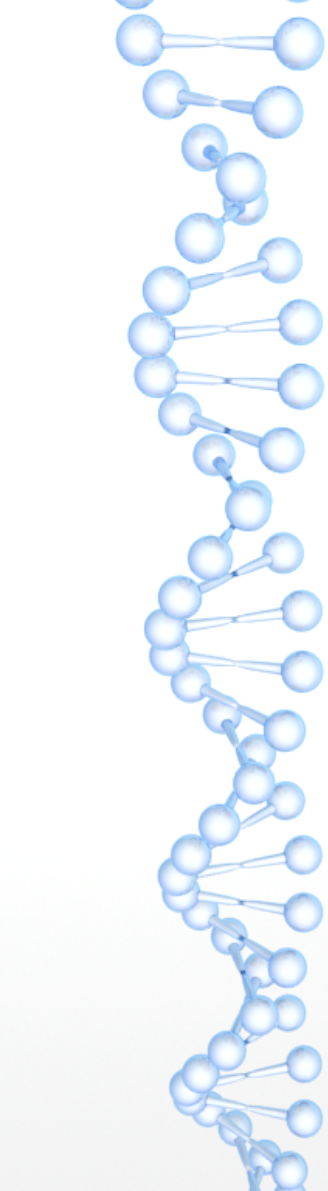

Activities
Firefox Web Browser
Thu 11:14
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
64 of 88
Search document
Android Applications (2)
Contents of an apk
1.A manifest describing what the application is, what it does, and how to run it
2.Resources needed by the application
3.The code itself
4.Signing information, securely identifying the author
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

# XML is used, more lightweight than html




Activities
Firefox Web Browser
Thu 11:15
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
66 of 88
Search document
Android Applications (4)
</intent-filter>
</activity>
<service android:name="com.example.email.SyncService">
</service>
<receiver android:name="com.example.email.SyncControlReceiver">
<intent-filter>
<action android:name="android.intent.action.DEVICE_STORAGE_LOW" />
</intent-filter>
<intent-filter>
<action android:name="android.intent.action.DEVICE_STORAGE_OKAY" />
</intent-filter>
</receiver>
<provider android:name="com.example.email.EmailProvider"
android:authorities="com.example.email.provider.email">
</provider>
</application>
</manifest>
Figure 10-51. Basic structure of AndroidManifest.xml.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles

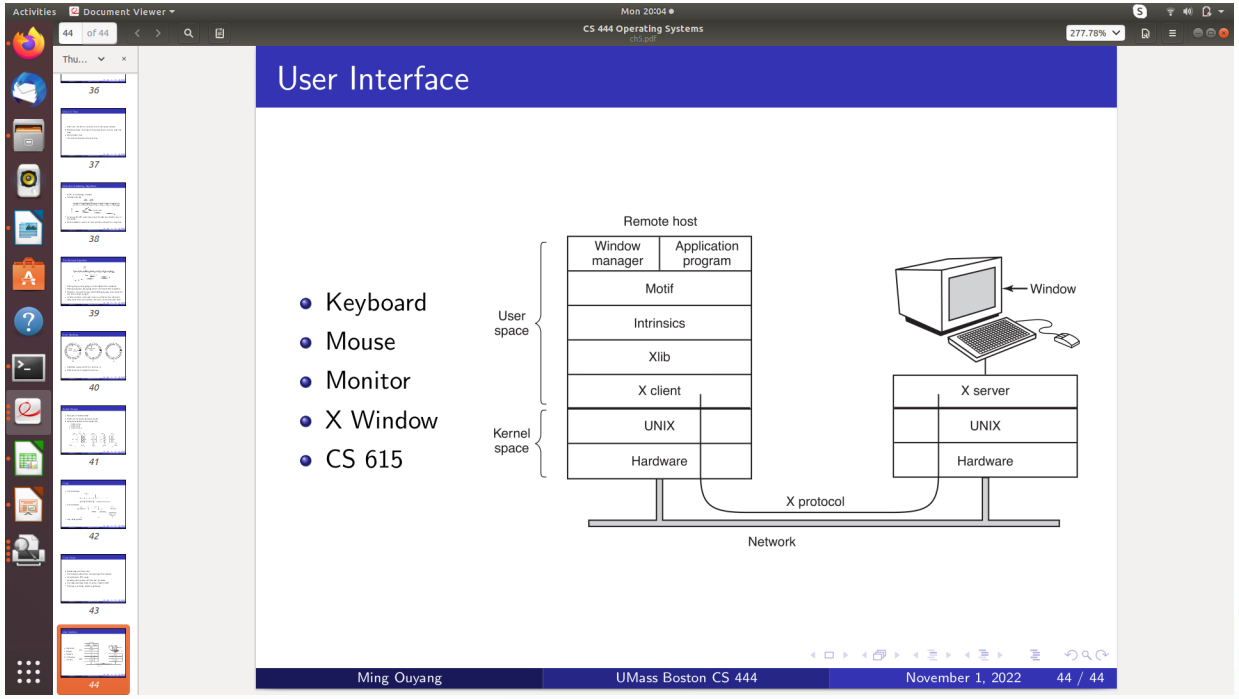

Activities
Firefox Web Browser
Thu 11:16
https://www.scribd.com/presentation/632885637/Chapter10-UNIXandLinuxAndAndroid
SCRIBD
Search
EN
Upload
Saved
Saved
Embed
Share
83 of 88
Search document
Starting Processes
System_server process
Application process
PackageManagerService
Activity instance
2 Resolve Intent
9
Application code
7
startActivity()
1
8 Instantiate this class
ActivityManagerService
6 Load this app s code
5 "Who am I?"
Android framework
4
3
Create a new process
Zygote process
Figure 10-67. Steps in launching a new application process.
Tanenbaum & Bos, Modern Operating Systems: 4th ed., Global Edition (c) 2015 Pearson Education Limited. All rights reserved.
Check out these free titles



# Chapter 11: Case Study: Windows 11

- Look back at Ch 5.6 Windowing
- Compare windowing in both Linux and Windows

# Windowing


Activities
Document Viewer
CS 444 Operating Systems
User Interface
Keyboard
Mouse
Monitor
X Window
CS 615
Remote host
Window manager
Application program
Motif
Intrinsics
Xlib
X client
UNIX
Hardware
User space
Kernel space
X protocol
Network
Window
X server
UNIX
Hardware
Ming Ouyang
UMass Boston CS 444
November 1, 2022
44 / 44

# Consider Windows

- Very sensitive to running legacy systems
- Not open source, so we have limited information
- Nice diagram of Windows 365, Cloud

# An overview of Windows in the Cloud