Do Now: UNIX Navigation Screenshot Safari

- Go to http:\\www.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
- Download today's activity files
- Read and complete the file named UNIX_Navigation_Screenshot_Safari.md

UNIX and “UNIX -like” Operating Systems

The Linux/UNIX Operating System

MR. SABAUGH 12.01.22

Today you'll learn...

- How to navigate the UNIX/Linux environment
- How UNIX differs from other OS
- A deep dive into the Linux Operating System

3 components of the Linux Operating System

1. The Kernel

- The kernel is responsible for maintaining all the important abstractions of the operating system including such things as virtual memory and processes
  - The kernel is the core part of Linux.
  - It is responsible for all major activities of this operating system.
  - It consists of various modules and it interacts directly with the underlying hardware.
  - The kernel provides the required abstraction to hide low-level hardware details in system or application programs.

3 components of the Linux Operating System

2. System Library

- System Libraries define a standard set of functions through which applications can interact with the kernel and that implement much of the operating -system functionality that doesn’t need the full privileges of kernel code.
    - System libraries are special functions or programs using which application programs or system utilities that access the Kernel's features.
    - These libraries implement most of the functionalities of the operating system and do not require the kernel module's code access rights.

3 components of the Linux Operating System

3. System Utilities

- System Utility programs are responsible to do specialized, individual-level tasks.
  - System Utilities are system programs that perform individual, specialized management tasks.
  - Some of the System utilities may be invoked just to initialize and configure an aspect of the system and others may run permanently, handling such tasks as responding to incoming network connections, accepting logon requests from terminals, or updating log files.

The Kernel's responsibility in Linux

- The kernel is the essential center of a computer operating system, the core that provides basic services for all other parts of the operating system.
- A synonym is a nucleus. A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands

Linux Kernel modules have three components:

- Module management
  - allows modules to be loaded into memory and to talk to the rest of the kernel
- Driver registration
  - allows modules to tell the rest of the kernel that a new driver has become available
- Conflict resolution mechanism
  - A mechanism that allows different device drivers to reserve hardware resources and to protect those resources from accidental use by another driver.
  - Linux provides a ventral conflict resolution mechanism.

Device drivers include...

- Character devices such as printers, terminals
- Block devices including all disk drives
- Network interface devices

Design principles of Linux systems

- Linux is a multiuser, multitasking system
- multiuser meaning resources are shared by all the users
    - Linux is UNIX compatible
    - its file system adheres to traditional UNIX semantics
    - it fully implements the standard UNIX networking model
    - its API adheres to the SVR4 UNIX semantics
    - it is POSIX-compliant
    - Linux supports a wide variety of architectures
    - The main design goals are speed, efficiency, and standardization

UNIX compatibility

- Linux and UNIX kernels are written differently because UNIX is proprietary and Linux is open-source although the shell commands are almost identical depending on the version
- All of the ‘above-the-hood’ functionalities are identical depending on the version
- In addition, UNIX is a multiuser, multitasking, and a time-sharing system

Process (Linux)

- A Process is the basic context within which all user-requested activity is serviced within the Operating system.

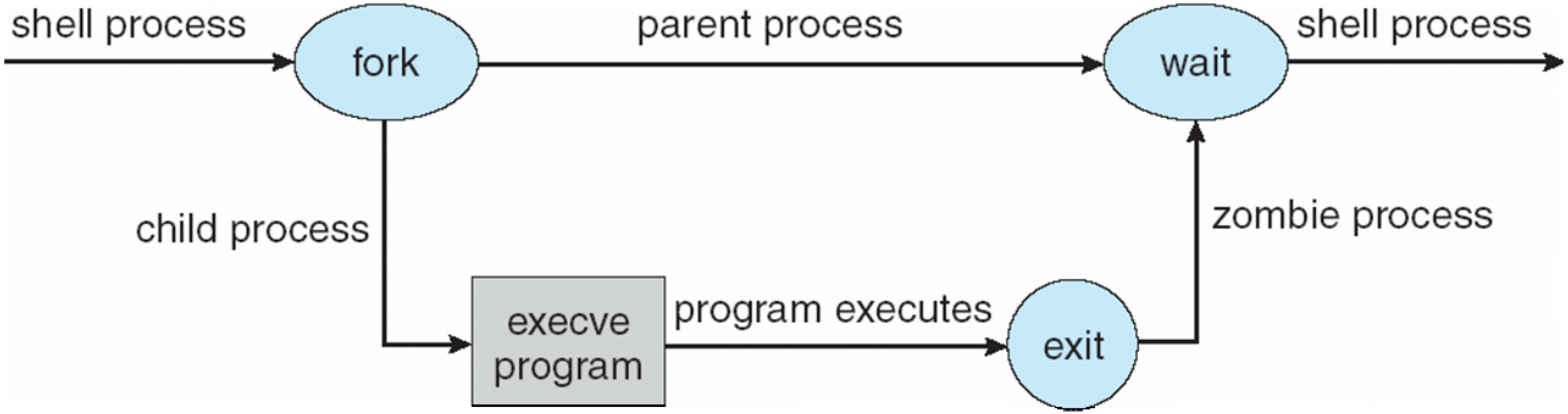
System Calls in UNIX/Linux

- System calls define the programmer interface to UNIX
- The set of systems programs commonly available defines the user interface
- The programmer and user interface define the context that the kernel must support
- Roughly three categories of system calls in UNIX
    1. File manipulation (same system calls also support device manipulation)
    2. Process control
    3. Information manipulation

Fork and exec system calls

- fork() is the name of the system call that the parent process uses to "divide" itself ("fork") into two identical processes.
- After calling fork(), the Creating child process is an exact copy of the parent except for the return value.
- When the child process calls exec(), all data in the original program is lost, and it is replaced with a running copy of the new program. This is known as overlaying.
- Purpose:
  - Fork is a System call by which a new process is created.
  - Exec is also a System call, which is used after a fork by one of the two processes to replace the process memory space with a new program.

Illustration of Process Control Calls



Shells and Commands

- Shell – the user process which executes programs (also called **command interpreter** or CLI)
- Called a shell, because it surrounds the kernel
- The shell indicates its readiness to accept another command by typing a prompt, and the user types a command on a single line
- A typical command is an executable binary object file
- The shell travels through the search path to find the command file, which is then loaded and executed

Shells and Commands

- The directories /bin and /usr/bin are almost always in the search path
    - bin directory contains executable files for most of the UNIX/Linux commands.
- Typical search path on a BSD (UNIX but very similar for Linux) system:

```
(./home/teacher/avi/bin /usr/local/bin /usr/cuny/bin /usr/bin)
```

- The shell usually suspends its own execution until the command completes
- `ps` for example is a command that displays the status of a process (like ctrl + alt + delete in windows)

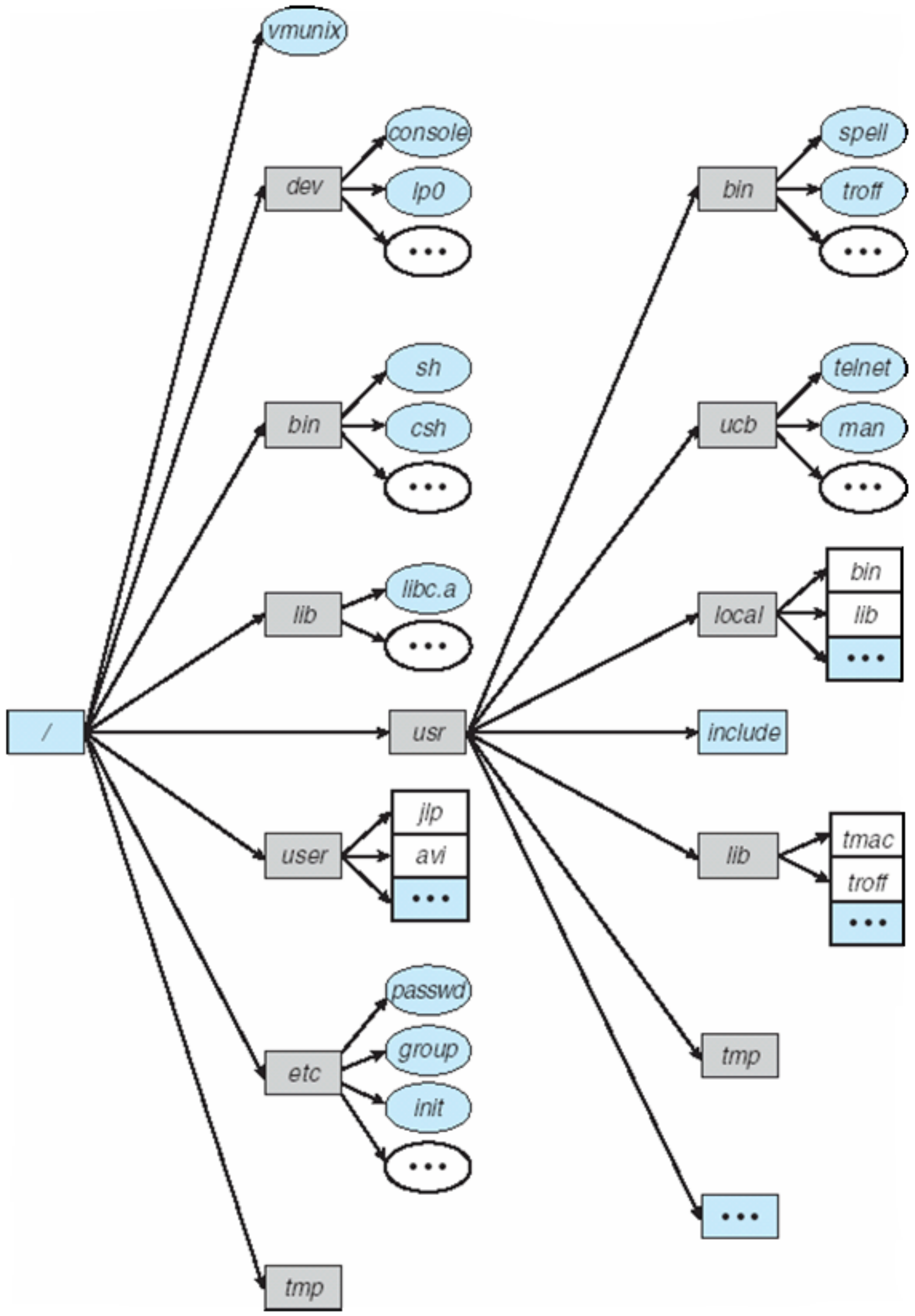
User Interface

- Programmers and users mainly deal with already existing systems programs: the needed system calls are embedded within the program and do not need to be obvious to the user.
- The most common systems programs are file or directory oriented
- Directory: `mkdir, rmdir, cd, pwd`
- File: `ls, cp, mv, rm`
- Other programs relate to editors (e.g., emacs, vi) text formatters (e.g., troff, TEX), and other activities

The File System

- The UNIX file system supports two main objects: files and directories.
- Directories are just files with a special format, so the representation of a file is the basic UNIX concept.

The File System



Files

- A file is a named collection of related information that is recorded on secondary storage.
- A file contains either programs or data.
- A file in Linux/UNIX is typeless, meaning it is just a collection of bytes. This affords much flexibility in interfacing and access.

File Manipulation

- A file is a sequence of bytes; the kernel does not impose a structure on files
- Files are organized in tree-structured directories
- Directories are files that contain information on how to find other files
- Path name: identifies a file by specifying a path through the directory structure to the file
    - Absolute path names start at the root (/) of the file system
    - Relative path names start at the current directory
- System calls for basic file manipulation: `create, open, read, write, close, unlink, trunc`

6 Basic File Operations

1. Creating a file
2. Writing a file
3. Reading a file
4. Repositioning within a file
5. Deleting a file
6. Truncating a file

Directory

- The device directory or simply known as the directory records information-such as name, location, and size for all files on that particular partition.
- The directory can be viewed as a symbol table that translates file names into their directory entries.

Path Name

- A pathname is the path from the root through all subdirectories to a specified file.
- If the first character is “/”, the starting directory is the root directory
- For any other starting character, the starting directory is the current directory