

CHAPTER II: UNIX/LINUX SYSTEM PRESENTATION

I/ Introduction:

The Unics (Uniplexed Information and Computing Service) system was created at AT&T Bell Laboratories in 1969 by Ken Thompson, and modified and named by Brian Kernighan. It was a scaled-down version of Multics (Multiplexed Information and Computing Service). Shortly afterwards, the name Unics was changed to Unix, and from then on, the Unix system began a long road of technical development.

II/ History

The Unix system was a real success when it was rewritten in the C language in 1973 by Denis Ritchie and Thompson. The University of California at Berkeley obtained a copy of version 6 of the Unix system. AT&T and Berkeley separately made numerous modifications and improvements to the Unix system (AT&T's System V and Berkeley's 4.4BSD). A simplified view of the evolution of Unix is shown in next figure 1.

The project standardization of the Unix system Posix (Portable Operating System UNIX) has led to the development of a standard that defines a set of procedures. All Posix-compliant Unix systems provide these standard procedures or system calls. These procedures constitute the standard Unix library. Any software written using only Posix standard procedures should run on all Unix systems conforming to this standard.

A free version of Unix is called Linux (source code available). It was created by Linus Torvalds in 1991. Subsequently, a large number of programmers contributed to its accelerated development. Initially designed to run on machines with the 80x86 processor, Linux has migrated to several other platforms. The Linux operating system can be downloaded from a number of sites, including <http://www.linux.org>.

FreeBSD is another free version of Unix. It is an operating system for x86 compatibles, DEC Alpha, and PC-98 architectures. It is derived from BSD Unix and developed by a large community of individuals.

III/ Characteristics:

- ☐ Unix is an interactive, multi-user (time-sharing) system. Several users can use the system's computing power at the same time.
- ☐ The Unix system controls and manages the use of resources, allocating them in turn to different users.
- ☐ It enables processes to be created, communicated and synchronized.
- ☐ Unix is an open, portable operating system available on a variety of platforms.
- ☐ Linux is also free, and you have the right to study and modify the source code.

- ❑ Linux is available in numerous distributions from a number of major branches, differing in terms of system maintenance tools and software packages, as well as server or consumer orientation: Slackware and its derivatives (Suse, OpenSuse, etc.), Redhat and its derivatives (Mandrake then Mandriva and today Mageia, CentOS, Scientific Linux, Fedora), which use the RPM package management system (Redhat Package Manager), Debian from which knopix and Ubuntu derive, etc. To get an idea of the abundance of products, take a look at the UNIX family tree <http://www.levenez.com/unix/> or the linux family tree http://fr.wikipedia.org/wiki/Distribution_Linux.

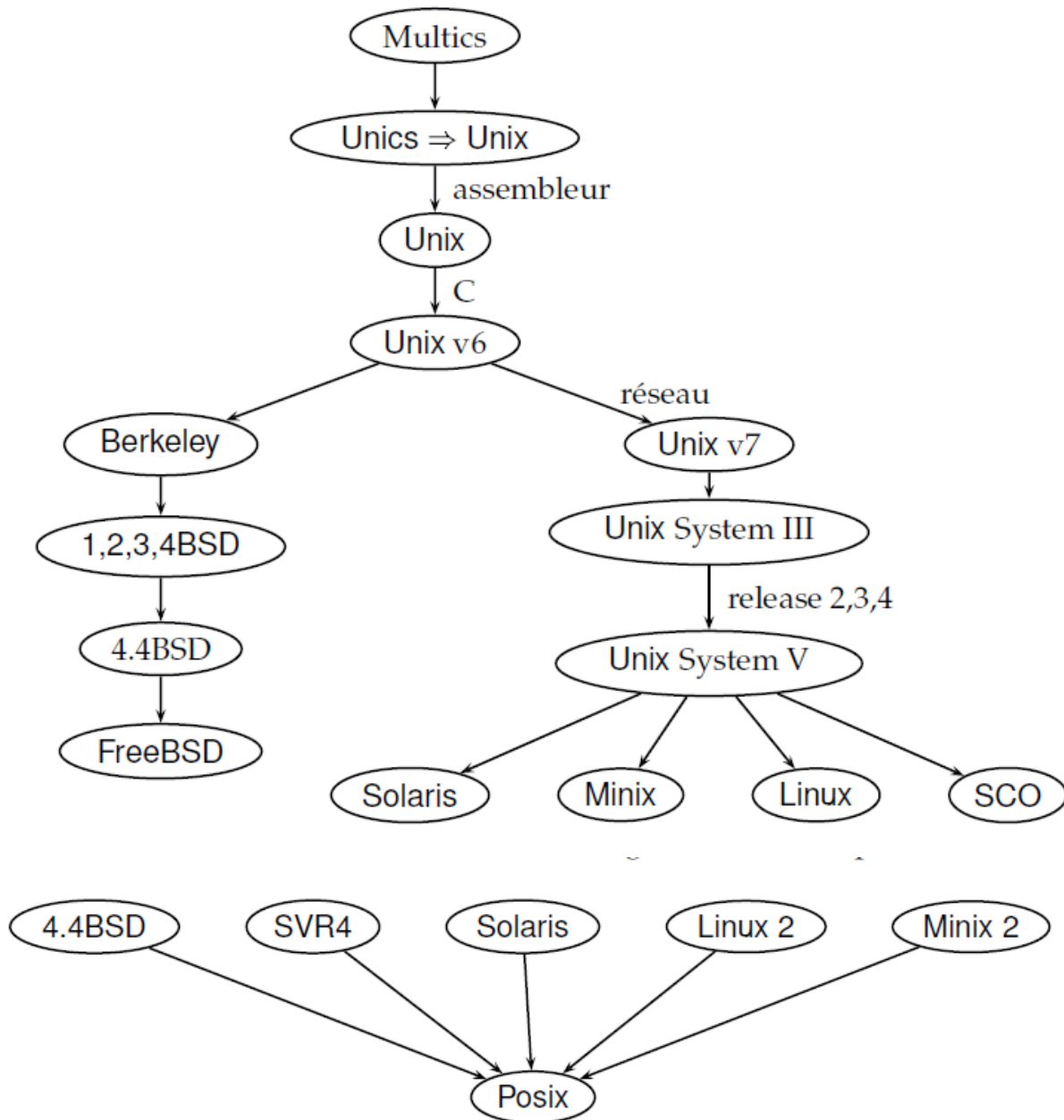


Figure 1: History of Unix system

1/ Unix system structure:

A Unix/Linux computer system is made up of layers of software, as shown in the figure 2.

The operating system, called the kernel, manages the hardware and provides programs with a system call interface. System calls enable programs to create and manage processes and files.

Each system call corresponds to a library procedure that the user can call (standard library). Each procedure places the parameters of the corresponding system call in a predefined location, such as the processor registers, and causes a software interrupt (TRAP instruction) to switch from user mode to kernel mode, thereby activating the operating system.

The purpose of the library procedure is to hide the details of the TRAP instruction and make procedure calls appear as ordinary procedure calls.

For example, **read(f, b, 50)** will be used to read **50 characters** from a temporary storage or buffer **b**, from a file **f** currently open.

When the operating system takes control following TRAP, it checks the validity of the parameters and carries out the requested processing. At the end of processing, it places a status code, indicating whether processing has succeeded or failed, in a register, then returns control to the library procedure and returns to user mode. The library procedure, in turn, gives control back to the caller, passing on the status code. Users can develop programs and invoke system calls, thanks to the standard library and a set of utilities (shells, editors, compilers, etc.) supplied with the Unix/Linux system.

For example, to execute the **READ** system call, a C/C++ program can call the standard library procedure **read()**, which will make the actual **READ** system call.

Note: True system calls run in kernel mode, protecting the operating system from intrusion attempts and errors.

2/ Unix from Windows:

Starting from a personal workstation running the Windows operating system, there are many ways of accessing the Unix world. We'll look at just a few of them, starting with the easiest to implement.

MobaXterm (see <https://mobaxterm.mobatek.net/>) allows you to locally simulate some of the commands of the UNIX system. It also provides tools for connecting to Unix servers.

Cygwin The cygwin software is a fairly complete emulation of the Linux system, but requires a slightly heavier installation.

wubi for Windows-based UBuntu Installer (see <https://doc.ubuntu-fr.org/wubi>) used to install Ubuntu on a Windows machine without modifying partitions. Since 2017, however, it has been discouraged, as several Linux distributions (including Ubuntu and OpenSuse) are available (in the Microsoft App Store) as simple Windows 10 subsystems, without modifying partitions or virtual machines (see <https://blog.ikoula.com/fr/sous-syst%C3%A8me-linux-pour-Windows>).

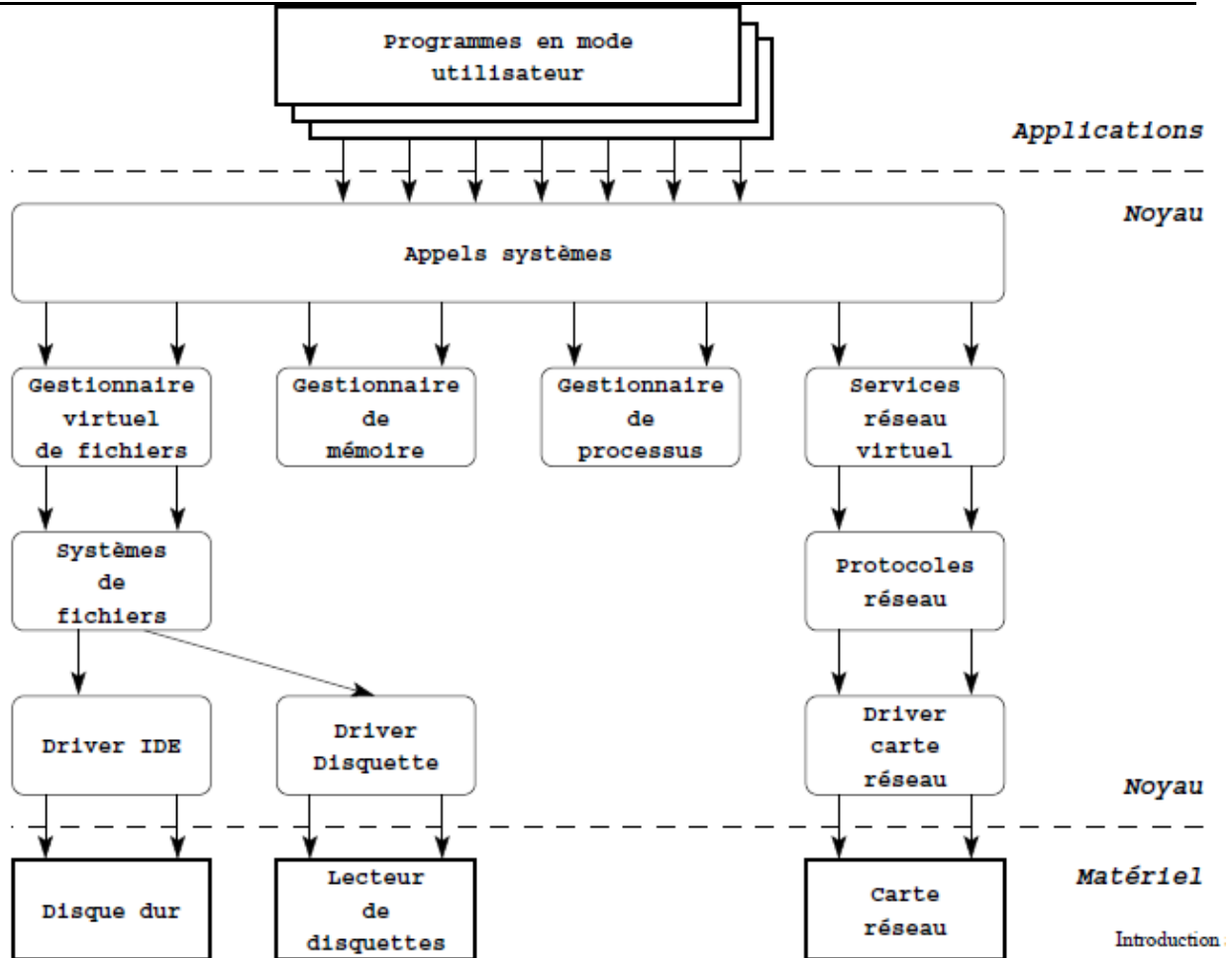


Figure 2: Layers of Unix system

Unix machines on dedicated media If you have a bootable physical medium, such as a disk or USB stick, which you can devote entirely to the Unix system, this can be a prudent solution, without interfering with the existing system. This is what the open-source software Linuxlive USB Creator offers for a wide range of Linux distributions (see <http://www.linuxliveusb.com>).

In virtual machines, you can also install a fairly complete UNIX system as a special Windows application: in this case, we speak of a virtual machine simulated by the physical machine, drawing on its resources. Virtual machines are guest operating systems, while the physical machine is the host system that hosts them. Before installing a virtual machine, you first need to install the virtual machine hosting software, known as virtualization software, such as Virtualbox (<https://www.virtualbox.org>), or VMware (<https://www.vmware.com>). It will then be possible to install several operating systems on the host machine.

Finally, let's mention the dual-boot technique: the Linux system is installed in parallel, possibly on the same disk, alongside the Windows system; the choice of which operating system to activate is made at boot-up. Installing Unix using this method requires administrative skills and entails the risk of information loss, particularly when partitioning the disk.

3/ Unix users:

Two types of people who use Unix

- ❑ The person who is going to use the system in order to produce something, for whom the UNIX system is a means, a tool. This person is the UNIX user, and can include programmers, database users and so on.
- ❑ The second category of people responsible for installing, configuring and running the UNIX system are UNIX system administrators. On a UNIX system, UNIX users have limited rights, i.e., they are not allowed to run certain commands or access certain parts of the system. System administrators, on the other hand, have full rights over the system.

4/ Principal functions of Unix:

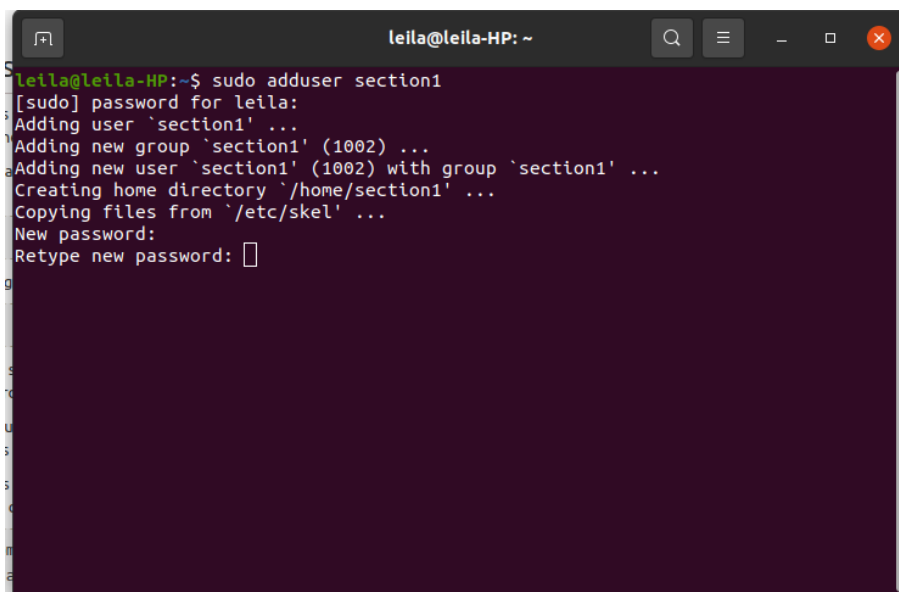
- ❑ Fair resource sharing
 - ✓ UNIX ensures that all computer resources (printer, memory, etc.) are shared equally between all processes.
- ❑ Hardware interface
 - ✓ UNIX ensures that no process has direct access to a hardware resource (hard disk, floppy drive, etc.):
 - ✓ To access these resources, we use special files. A special file is seen by a user as a classic file: to write to a floppy disk in the floppy disk drive, we simply write to the special file in the floppy disk drive. Similarly, to read from a hard disk, you read the special file on the hard disk.
- ❑ Memory management
 - ✓ Unix must therefore ensure that memory is shared between all processes: one process must not monopolize all memory, otherwise other processes would no longer be able to function.
 - ✓ Memory is seen as a hardware resource, so Unix must ensure that no process accesses memory directly or reserves a memory area for itself.
- ❑ File management
 - ✓ Unix provides the tools needed to store data and retrieve it quickly and easily. It also provides the tools for easy viewing of all files. These files are located on the hard disk, known as a File System.
 - ✓ Unix also provides a file protection mechanism. Several users can work on the same machine at the same time - this is known as multi-user. Each user of the system has his or her own files, and UNIX gives them the means to protect their files, and to grant or deny other users access to their files.

5/ Unix shell:

- To operate the computer, the user has access to software or a utility that allows direct input of commands. This utility is called the shell. Its role is to interpret user commands before transmitting them to the kernel, which is why it's also known as a command interpreter.

- There are several types of shell, differing in their syntax and the richness of their commands.
The most common is the Bourne shell, but there is also the C shell, which is similar to the C programming language, the Korn shell, the Posix shell and, under Linux, the bash shell.
- Non-graphical user interface
 - ✓ Text terminals
 - ✓ Remote access
- Script interpreter
 - ✓ Batch processing
 - ✓ Automation
- Launch multiple tasks
 - ✓ Combined tasks (pipe)
 - ✓ Job control
- It's an ordinary program
- Vast choices:
 - ✓ sh, sash, ksh, csh, tcsh, ash, bash, psh, fish, ...
- Multiple shells can be run in parallel (Unix is multitasking)
- Several users can run a shell at the same time (Unix is multi-user)

Note: To access the shell, use the **ALT+CTRL+T** keystroke combination.



```
leila@leila-HP: ~  
$ sudo adduser section1  
[sudo] password for leila:  
Adding user `section1' ...  
Adding new group `section1' (1002) ...  
Adding new user `section1' (1002) with group `section1' ...  
Creating home directory `/home/section1' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password: 
```