Operating system concepts

User and Operating-System Interface
Slides Set #3

By Prof K R Chowdhary

JNV University

2023

OS Structure. (Kernel=OS)

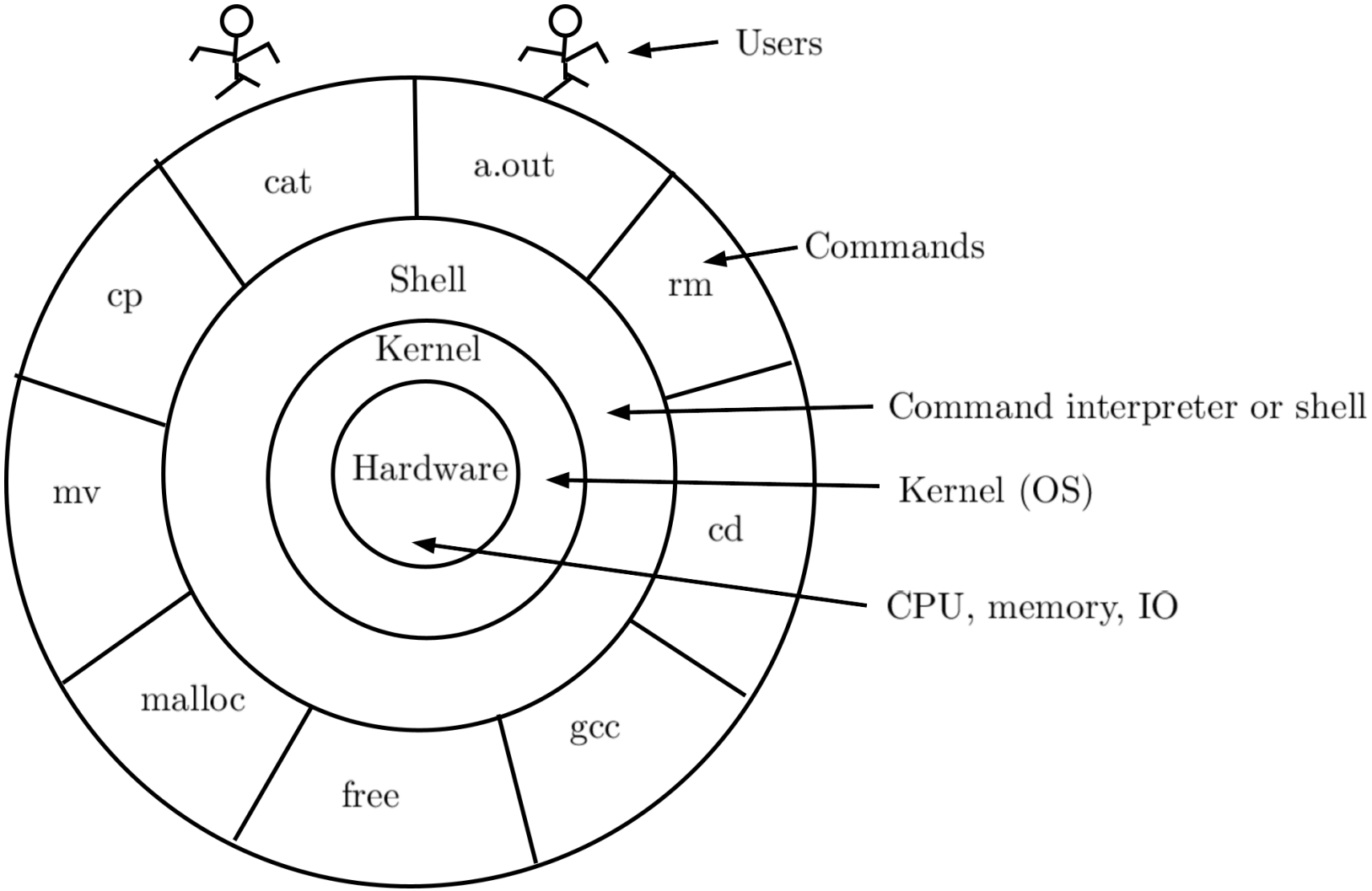

Figure 1: OS Structure

User and Operating-System Interface

There are several ways for users to interface with the operating system. But, there are two fundamental approaches.

- *Command-line interface*, or *command interpreter*, that allows users to directly enter commands to be performed by the operating system.
- The other allows users to interface with the operating system via a *graphical user interface*, or GUI.

Command interpreters:

- Some operating systems include the command interpreter in the kernel (e.g. DOS). Others, such as Windows and UNIX, treat the *command interpreter* as a special program
- On UNIX and Linux systems, a user may choose among several different shells, including the *Bourne shell*, *C* shell, *Korn shell*

Command interpreters..., Some commands.

```
krc@krc-Inspiron-13-5378: ~
krc@krc-Inspiron-13-5378:~$ pwd
/home/krc
krc@krc-Inspiron-13-5378:~$ ls
aa    BCA.docx  Desktop    Downloads     Music  Pictures  temp.doc   Templates  works
aa.m  BCA.pdf   Documents  missfont.log  neo    Public    temp.docx  Videos
krc@krc-Inspiron-13-5378:~$
krc@krc-Inspiron-13-5378:~$ ls -l
total 136
-rw-rw-r--  1 krc krc   613 Jun 20 13:14 aa
-rw-rw-r--  1 krc krc   145 Jun 20 09:19 aa.m
-rw-rw-r--  1 krc krc  6595 May 31 18:15 BCA.docx
-rw-rw-r--  1 krc krc 57246 May 31 18:15 BCA.pdf
drwxr-xr-x 14 krc krc  4096 Sep  1 11:40 Desktop
drwxr-xr-x 42 krc krc  4096 Aug  8 09:40 Documents
drwxr-xr-x  3 krc krc  4096 Aug 27 16:39 Downloads
-rw-rw-r--  1 krc krc   310 Aug 12 12:48 missfont.log
drwxr-xr-x  2 krc krc  4096 Jan 26  2023 Music
drwxrwxr-x  3 krc krc  4096 Jul 15 23:25 neo
drwxr-xr-x  3 krc krc  4096 Aug 31 15:58 Pictures
drwxr-xr-x  2 krc krc  4096 Jan 26  2023 Public
-rw-rw-r--  1 krc krc  9216 Jul 14 10:35 temp.doc
-rw-rw-r--  1 krc krc  4918 Aug  7 22:10 temp.docx
drwxr-xr-x  2 krc krc  4096 Jan 26  2023 Templates
drwxr-xr-x  2 krc krc  4096 Jan 28  2023 Videos
drwxr-xr-x 65 krc krc  4096 Aug 11 09:27 works
krc@krc-Inspiron-13-5378:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
tmpfs             785268     2160    783108   1% /run
/dev/sda4       74501012 35837720  34833124  51% /
tmpfs            3926332        0   3926332   0% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
/dev/sda5       95533536 38928096  51706372  43% /home/krc/works
/dev/sda1         523244     6216    517028   2% /boot/efi
tmpfs             785264     1664    783600   1% /run/user/1000
krc@krc-Inspiron-13-5378:~$
```

Command interpreters work by system calls

- *Function of command interpreter*: Get and execute the next user-specified command.
- The commands given at this level manipulate files: *create*, *delete*, *list*, *print*, *copy*, *execute*, and so on. (The MS-DOS and UNIX shells operate in this way.)
- Commands can be implemented in two ways.
  - Command interpreter itself contains the code to execute the command. Ex. *command.com* in DOS
  - UNIX implements most commands through system programs. For example:

    rm file.txt

    would search for a file called *rm*, load the file *rm* into memory, and execute it with the parameter (argument) *file.txt*.

Some Commands of Unix

Command	Description
ls	Lists the files and directories
cat file.txt	Displays the file file.txt
mv f1 f2	Renames file f1 as f2
cp f1 f2	Copies file f1 into file f2
gcc abc.c	Compiles the abc.c file
df	Display the file system
wc file1	Count words, lines and char in file1
gedit f1	Opens the editor for file f1
vi f1.txt	Opens the vi editor for file f1

Online unix terminal:
https://www.tutorialspoint.com/linux_terminal_online.php

Your program to implementing the '*cat*' Command

Like, compiler allocates space for a *char* (1 byte) and names it as ch, the compiler creates a data structure 'FILE' and points it by pointer fp.

krc@krc-Inspiron-13-5378: ~/works/operating-system/my...

```
krc@krc-Inspiron-13-5378:~/works/operating-system/my_osslides/lect3$ cat cat2.c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
        char ch;
        FILE *fp;
        fp = fopen(argv[1], "r");
        ch=fgetc(fp);
        while(ch!=EOF){
                putchar(ch);
                ch=fgetc(fp);
        }
   fclose(fp);
 return 0;
}

krc@krc-Inspiron-13-5378:~/works/operating-system/my_osslides/lect3$ gcc cat2.c
krc@krc-Inspiron-13-5378:~/works/operating-system/my_osslides/lect3$
```

Running the cat Command program

At command line, ‘cat2.c’ is passed as argument ‘argv[1]’ to main, the program opens cat2.c in ‘r’ mode, reads it, and prints on screen.

krc@krc-Inspiron-13-5378: ~/works/operating-system/my_ossli...

```
krc@krc-Inspiron-13-5378:~/works/operating-system/my_osslides/lect3$ ./a.out cat2.c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
        char ch;
        FILE *fp;
        fp = fopen(argv[1], "r");
        ch=fgetc(fp);
        while(ch!=EOF){
                putchar(ch);
                ch=fgetc(fp);
        }
   fclose(fp);
 return 0;
}

krc@krc-Inspiron-13-5378:~/works/operating-system/my_osslides/lect3$
```

Graphical user Interface

- Another strategy for interfacing with the operating system is through a user friendly graphical user interface, or GUI.
- Because a mouse is impractical for most mobile systems, smartphones and handheld tablet computers typically use a touchscreen interface.
- Traditionally, UNIX systems have been dominated by command-line interfaces. Various GUI interfaces are available. These include the Common Desktop Environment (CDE) and X-Windows systems, which are common on commercial versions of UNIX, such as Solaris and IBM's AIX system.

System Calls: Special programs or functions calls, as part of Kernel

- *System calls* provide an interface to the services made available by an operating system. These calls are generally available as routines written in C or C++
- Writing a simple program to read data from one file and copy them to another file. $ cp file1.txt file2.txt
- In an *interactive system*, this approach will require a sequence of system calls ?
- The user can then use the mouse to select the source name, and a window can be opened for the destination name to be specified.