

## The Operating System

### Objective

**2.1.6** Describe the main functions of an operating system.

### Before You Begin

The following image shows the sound setup for an old MS-DOS game ("Nascar Racing") using the DOSBox emulator software.



Image courtesy abandonia.com.

Figure 1: A screenshot of "Nascar Racing" showing user manipulation of the sound card.

In particular, note the level of detail about the system sound card the game requires in order to function properly.

**Question #1** Modern operating systems provide abstraction through device drivers and system libraries for interacting with common system hardware and peripherals (such as sound board, graphic cards, input devices, etc.). How does this abstraction layer impact the development of software on these systems?

**Question #2** The typical user of a computer system does not need to interact directly with his/her hardware; however, many games benefit from game-specific settings which impact performance and gameplay. How important is it for the user of a computer system to be able to manipulate some aspects of their hardware directly?

## Important Terms

Term	Definition
<b>Device Driver</b>	
<b>Hardware Abstraction Layer</b>	
<b>Kernel</b>	
<b>Kernel Space</b>	
<b>Operating System (OS)</b>	
<b>User Space</b>	



## Technical Background

### The Role of the Operating System

The following diagram is an abstraction showing where the operating system exists in relation to the user applications and the system hardware.

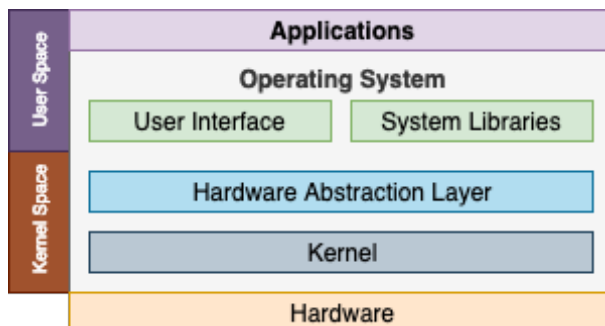


Figure 2: A simple view of the software layers of a computer system.

### Notes

**Question #3** What benefit does the segregation of memory space for core kernel operations from user-initiated operations have on the stability of a computer system? Are there any drawbacks to this model?

## Monolithic vs. Microkernel Operating Systems

The following two diagrams show a sample of fundamental services offered by the operating system, and where they would exist within a monolithic or microkernel operating system architecture.

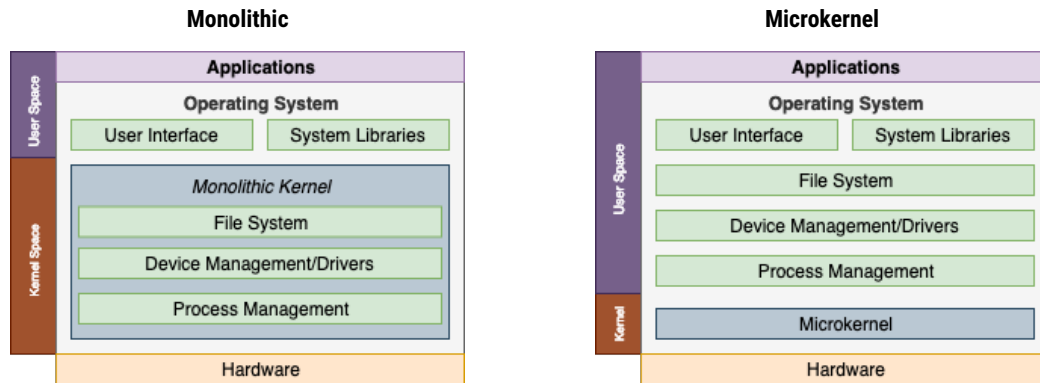


Figure 3: Monolithic vs. Microkernel Architectures.

### Notes

**Question #4** Monolithic kernels run a higher risk of instability due to all services being included in a single process. Despite this, operating systems which are considered more stable and more secure, such as Linux, generally use monolithic kernels, while systems considered less stable and less secure, such as Mac OSX and Windows, use microkernels. Why do you think this contradiction exists?

## Developing Technical Skills

### Kernel Development

As an open-source project, the Linux kernel grants us the opportunity to examine the kernel development process. In particular, we can note that the kernel has to be created in such a way as to handle the wide variety of hardware platforms now available on the market.

The following excerpt is taken from the changelogs for version 5.3.12 of the Linux kernel:

```
Author: Kai-Heng Feng <kai.heng.feng@canonical.com>
Date:   Wed Oct 16 18:38:16 2019 +0800

x86/quirks: Disable HPET on Intel Coffe Lake platforms

commit fc5db58539b49351e76f19817ed1102bf7c712d0 upstream.

Some Coffee Lake platforms have a skewed HPET timer once the SoCs entered
PC10, which in consequence marks TSC as unstable because HPET is used as
watchdog clocksource for TSC.

Harry Pan tried to work around it in the clocksource watchdog code [1]
thereby creating a circular dependency between HPET and TSC. This also
ignores the fact, that HPET is not only unsuitable as watchdog clocksource
on these systems, it becomes unusable in general.

Disable HPET on affected platforms.
```

Navigate to <https://www.kernel.org> and select the changelog link for the most current *stable* release.

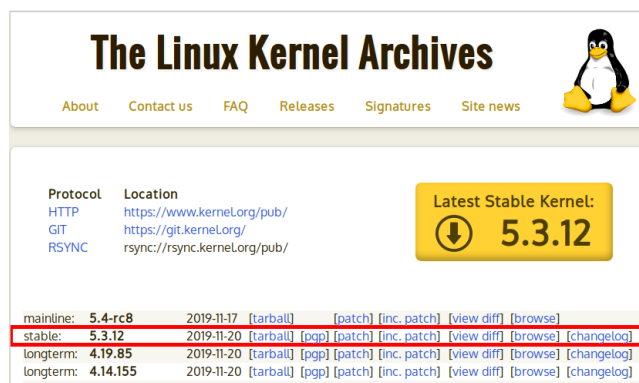


Figure 4: Screenshot of the Linux Kernel Archives.

Skim through the changelog to identify at least one change that has occurred within this release targeting a specific set of hardware or devices.

**Question #5** Significant advances and changes in computer hardware frequently require updates to an operating system's kernel, particularly in monolithic systems such as Linux. Do you think operating system vendors should push for more standardization of computer hardware? Why or why not?

## Reflections

**Question #6** Most Linux distributions pride themselves on being highly customizable, allowing users to change some of the smallest details of how their system runs. Briefly explain the advantage and disadvantage to this approach when compared to more restrictive systems, such as Apple's Mac OS X.

**Question #7** Windows Vista and newer require kernel-mode device drivers to be "digitally signed" by Microsoft. This process verifies a driver's use and authenticity. Why do you think Microsoft implemented this security measure?

**Question #8** Describe at least one new thing you have learned from this lesson. How might you apply this knowledge in the future?

**Question #9** Select the option which best reflects how confident you are in applying what you have learned in this lesson.



**Question #10** What further questions do you still have about this lesson's content?