

# Programming Interface

The other purpose of an OS is to provide an interface for the programmer:

# Programming Interface

The other purpose of an OS is to provide an interface for the programmer:

The programmer who has to write applications for the machine does not want to have to know the details of the hardware: think portability (c.f. von Neumann's model)

# Programming Interface

The other purpose of an OS is to provide an interface for the programmer:

The programmer who has to write applications for the machine does not want to have to know the details of the hardware: think portability (c.f. von Neumann's model)

- How do I get the best performance out of this disk, network, video?

# Programming Interface

The other purpose of an OS is to provide an interface for the programmer:

The programmer who has to write applications for the machine does not want to have to know the details of the hardware: think portability (c.f. von Neumann's model)

- How do I get the best performance out of this disk, network, video?
- How do I prod this hardware to get it to do what I want?

# Programming Interface

The other purpose of an OS is to provide an interface for the programmer:

The programmer who has to write applications for the machine does not want to have to know the details of the hardware: think portability (c.f. von Neumann's model)

- How do I get the best performance out of this disk, network, video?
- How do I prod this hardware to get it to do what I want?
- How should I deal with interrupts?

# Programming Interface

The other purpose of an OS is to provide an interface for the programmer:

The programmer who has to write applications for the machine does not want to have to know the details of the hardware: think portability (c.f. von Neumann's model)

- How do I get the best performance out of this disk, network, video?
- How do I prod this hardware to get it to do what I want?
- How should I deal with interrupts?
- And so on

# Programming Interface

We don't want to have to re-implement everything in every program

# Programming Interface

We don't want to have to re-implement everything in every program

So the OS does this kind of thing for us



# Programming Interface

We don't want to have to re-implement everything in every program

So the OS does this kind of thing for us

Early programmers, before OSs, had to do it all themselves

# Programming Interface

We don't want to have to re-implement everything in every program

So the OS does this kind of thing for us

Early programmers, before OSs, had to do it all themselves

Much better to let someone else do the hard work

# Programming Interface

We don't want to have to re-implement everything in every program

So the OS does this kind of thing for us

Early programmers, before OSs, had to do it all themselves

Much better to let someone else do the hard work  
(A common theme in Computer Science)

# Programming Interface

Having an expert do this stuff once and provide a standard interface to it for us to use is much better for us

# Programming Interface

Having an expert do this stuff once and provide a standard interface to it for us to use is much better for us

- We don't have to do it

# Programming Interface

Having an expert do this stuff once and provide a standard interface to it for us to use is much better for us

- We don't have to do it
- The expert is better at it and (presumably) understands the hardware well

# Programming Interface

Having an expert do this stuff once and provide a standard interface to it for us to use is much better for us

- We don't have do it
- The expert is better at it and (presumably) understands the hardware well
- The expert is a better programmer than us and can get better performance out of the hardware

# Programming Interface

Having an expert do this stuff once and provide a standard interface to it for us to use is much better for us

- We don't have to do it
- The expert is better at it and (presumably) understands the hardware well
- The expert is a better programmer than us and can get better performance out of the hardware
- The programmer knows more Computer Science than us and knows the many pitfalls and necessary tricks that OS programming involves



# Programming Interface

They do it so we don't have to

# Programming Interface

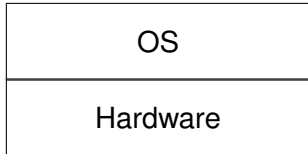
Layer Abstraction:

Hardware

PC, phone, PVR, SatNav

# Programming Interface

Layer Abstraction:

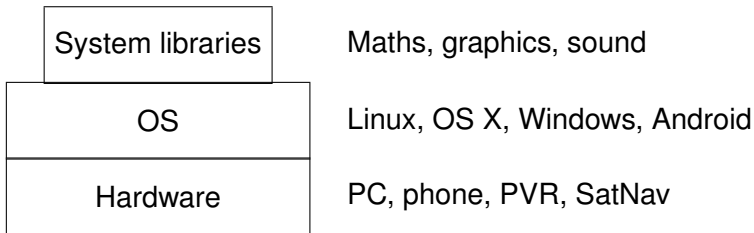


Linux, OS X, Windows, Android

PC, phone, PVR, SatNav

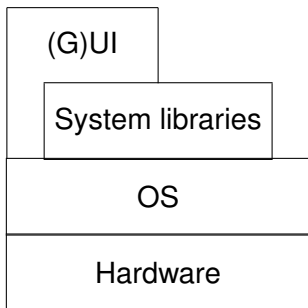
# Programming Interface

Layer Abstraction:



# Programming Interface

Layer Abstraction:



Command line, windowing, touch

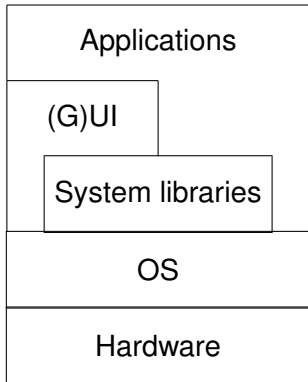
Maths, graphics, sound

Linux, OS X, Windows, Android

PC, phone, PVR, SatNav

# Programming Interface

Layer Abstraction:



Browser, word processor, game

Command line, windowing, touch

Maths, graphics, sound

Linux, OS X, Windows, Android

PC, phone, PVR, SatNav

# Important Point

Reemphasising a very important point:

The GUI is not part of the OS

# Important Point

Reemphasising a very important point:

The GUI is not part of the OS

The GUI is just another program that uses the OS



# Important Point

Reemphasising a very important point:

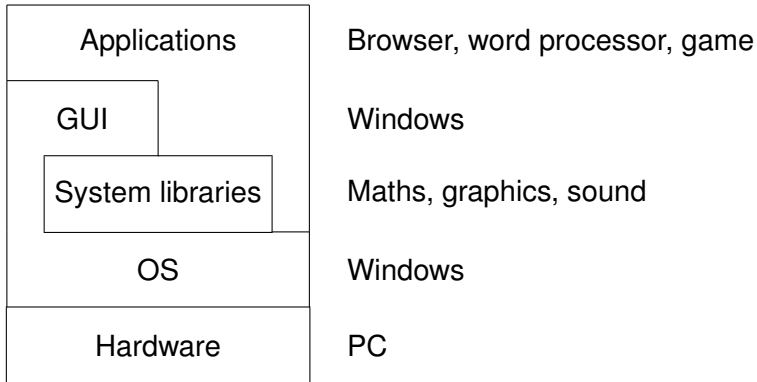
The GUI is not part of the OS

The GUI is just another program that uses the OS

There was a time when certain OS vendors tried to tie the GUI into the OS (to gain speed and commercial advantage)

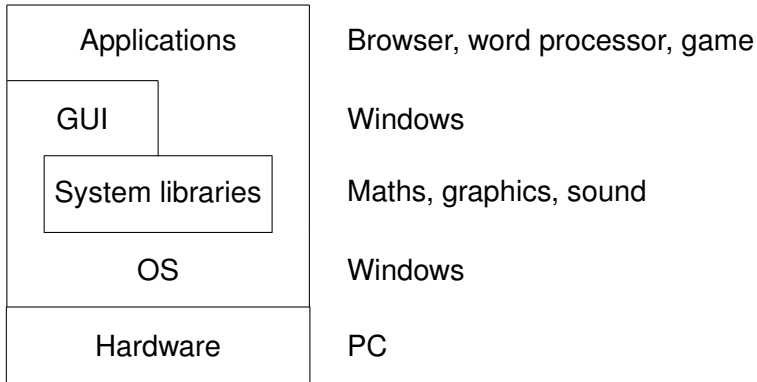
# Programming Interface

Bad Layer Abstraction:



# Programming Interface

Very Bad Layer Abstraction:



# Programming Interface

This is poor design and should be avoided

# Programming Interface

This is poor design and should be avoided

It was a huge source of problems: bugs in the GUI or the application would cause the OS to crash, so a poorly written program could take out the whole machine

# Programming Interface

This is poor design and should be avoided

It was a huge source of problems: bugs in the GUI or the application would cause the OS to crash, so a poorly written program could take out the whole machine

It was an easy way to circumvent the security the OS provides, thus allowing attackers to access the machine

## Another Important Point

Only a small number of CPU cycles in the world deal with GUIs or windows: the PC is far from being the most common type of computer

## Another Important Point

Only a small number of CPU cycles in the world deal with GUIs or windows: the PC is far from being the most common type of computer

Embedded systems out-sell PCs by orders of magnitude



## Another Important Point

Only a small number of CPU cycles in the world deal with GUIs or windows: the PC is far from being the most common type of computer

Embedded systems out-sell PCs by orders of magnitude

This Unit has nothing to do with GUIs

## Another Important Point

Only a small number of CPU cycles in the world deal with GUIs or windows: the PC is far from being the most common type of computer

Embedded systems out-sell PCs by orders of magnitude

This Unit has nothing to do with GUIs

**You will lose marks** in this Unit if you start talking about GUIs as it means you don't understand the concept of an OS

## Another Important Point

Only a small number of CPU cycles in the world deal with GUIs or windows: the PC is far from being the most common type of computer

Embedded systems out-sell PCs by orders of magnitude

This Unit has nothing to do with GUIs

**You will lose marks** in this Unit if you start talking about GUIs as it means you don't understand the concept of an OS

It quite possible to have similar-looking GUIs running on different OSs

## Another Important Point

Only a small number of CPU cycles in the world deal with GUIs or windows: the PC is far from being the most common type of computer

Embedded systems out-sell PCs by orders of magnitude

This Unit has nothing to do with GUIs

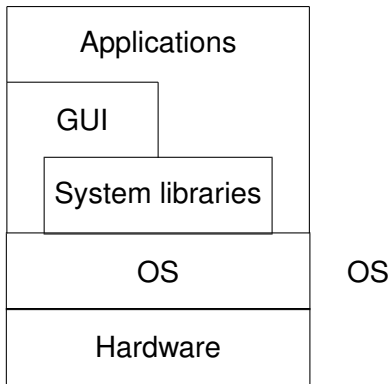
**You will lose marks** in this Unit if you start talking about GUIs as it means you don't understand the concept of an OS

It quite possible to have similar-looking GUIs running on different OSs

But we must be careful as some people don't realise the difference between an OS and everything else

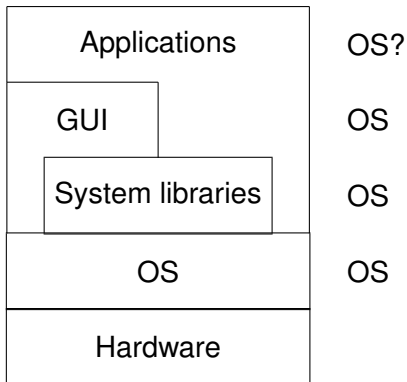
# Programming Interface

CS view:



# Programming Interface

Marketer's view:



## One more thing

There are a couple of vital aspects of an OS that are sometimes overlooked:



## One more thing

There are a couple of vital aspects of an OS that are sometimes overlooked:

1. It should be efficient and lightweight: every CPU cycle that the OS uses is one that is taken away from the user's programs





## One more thing

There are a couple of vital aspects of an OS that are sometimes overlooked:

1. It should be efficient and lightweight: every CPU cycle that the OS uses is one that is taken away from the user's programs
2. It should be flexible and not get in the way of the programmer



## One more thing

There are a couple of vital aspects of an OS that are sometimes overlooked:

1. It should be efficient and lightweight: every CPU cycle that the OS uses is one that is taken away from the user's programs
2. It should be flexible and not get in the way of the programmer

So a perfect OS would be completely invisible!

