Background

Operating Systems have been around nearly as long as computers

Background

Operating Systems have been around nearly as long as computers

. . . but not quite as long

Background

Operating Systems have been around nearly as long as computers

. . . but not quite as long

So long, though, that some (most!) people don’t distinguish between computers and their OS

Background

Operating Systems have been around nearly as long as computers

. . . but not quite as long

So long, though, that some (most!) people don’t distinguish between computers and their OS

Computer Scientists ought to be more careful

Background

An OS is just a program so

Background

An OS is just a program so

- we can have different OSs on the same hardware

Background

An OS is just a program so

- we can have different OSs on the same hardware
- we can have the same OS on different hardware

Background

An OS is just a program so

- we can have different OSs on the same hardware
- we can have the same OS on different hardware

- On Intel hardware (“PC”) you can run Windows, MacOS, Linux (and lots of others)

Background

An OS is just a program so

- we can have different OSs on the same hardware
- we can have the same OS on different hardware

- On Intel hardware (“PC”) you can run Windows, MacOS, Linux (and lots of others)
- Current Mac hardware is the same as PC hardware, so you can run all of the above on a Mac (the new Mac M1 is effectively only slightly different)

Background

An OS is just a program so

- we can have different OSs on the same hardware
- we can have the same OS on different hardware

- On Intel hardware (“PC”) you can run Windows, MacOS, Linux (and lots of others)
- Current Mac hardware is the same as PC hardware, so you can run all of the above on a Mac (the new Mac M1 is effectively only slightly different)
- Linux runs on many kinds of hardware, including Intel, mainframes (IBM, Oracle/Sun), million processor clusters, phones (ARM), gadgets like satnavs and PVRs

Background

An OS is just a program so

- we can have different OSs on the same hardware
- we can have the same OS on different hardware

- On Intel hardware ("PC") you can run Windows, MacOS, Linux (and lots of others)
- Current Mac hardware is the same as PC hardware, so you can run all of the above on a Mac (the new Mac M1 is effectively only slightly different)
- Linux runs on many kinds of hardware, including Intel, mainframes (IBM, Oracle/Sun), million processor clusters, phones (ARM), gadgets like satnavs and PVRs

N.B. when we say "Intel" or "x86" we mean Intel, AMD and all the compatible architectures

Background

A note on portability of OSs

Background

A note on portability of OSs

Since an OS has to deal with the details of hardware, it is not entirely trivial to port an OS from one CPU architecture to another, e.g,. an Intel CPU to an ARM CPU

Background

A note on portability of OSs

Since an OS has to deal with the details of hardware, it is not entirely trivial to port an OS from one CPU architecture to another, e.g,. an Intel CPU to an ARM CPU

Well designed OSs minimise the hardware-dependent parts and try to keep most code hardware-independent

Background

A note on portability of OSs

Since an OS has to deal with the details of hardware, it is not entirely trivial to port an OS from one CPU architecture to another, e.g,. an Intel CPU to an ARM CPU

Well designed OSs minimise the hardware-dependent parts and try to keep most code hardware-independent

Most of the difficult code in the OS can be kept hardware independent, only the stuff that needs the actual physical access need be hardware dependent

Background

A note on portability of OSs

Since an OS has to deal with the details of hardware, it is not entirely trivial to port an OS from one CPU architecture to another, e.g,. an Intel CPU to an ARM CPU

Well designed OSs minimise the hardware-dependent parts and try to keep most code hardware-independent

Most of the difficult code in the OS can be kept hardware independent, only the stuff that needs the actual physical access need be hardware dependent

Exercise. Have a browse through the Linux kernel source code

Background

Historically, some OSs tied themselves so closely to the hardware that porting is very difficult

Background

Historically, some OSs tied themselves so closely to the hardware that porting is very difficult

For much of its lifetime Windows only ran on Intel, so the assumption of Intel hardware ran throughout the code

Background

Historically, some OSs tied themselves so closely to the hardware that porting is very difficult

For much of its lifetime Windows only ran on Intel, so the assumption of Intel hardware ran throughout the code

Microsoft have recently ported Windows to ARM chips

Background

Historically, some OSs tied themselves so closely to the hardware that porting is very difficult

For much of its lifetime Windows only ran on Intel, so the assumption of Intel hardware ran throughout the code

Microsoft have recently ported Windows to ARM chips

They are frantically trying to catch up with a market that is moving to phone/tablet and energy-conscious computing

Background

Historically, some OSs tied themselves so closely to the hardware that porting is very difficult

For much of its lifetime Windows only ran on Intel, so the assumption of Intel hardware ran throughout the code

Microsoft have recently ported Windows to ARM chips

They are frantically trying to catch up with a market that is moving to phone/tablet and energy-conscious computing

And they paid the price of not abstracting away the hardware from the software

Background

Historically, some OSs tied themselves so closely to the hardware that porting is very difficult

For much of its lifetime Windows only ran on Intel, so the assumption of Intel hardware ran throughout the code

Microsoft have recently ported Windows to ARM chips

They are frantically trying to catch up with a market that is moving to phone/tablet and energy-conscious computing

And they paid the price of not abstracting away the hardware from the software

And note this will require rewrites of Windows applications, too

Background

In contrast, the Android OS that runs on phones shares a large amount of code with the Linux that runs everywhere else, even on supercomputers

Background

In contrast, the Android OS that runs on phones shares a large amount of code with the Linux that runs everywhere else, even on supercomputers

The iPhoneOS that runs on iPhones shares code with OS X

Background

In contrast, the Android OS that runs on phones shares a large amount of code with the Linux that runs everywhere else, even on supercomputers

The iPhoneOS that runs on iPhones shares code with OS X

Windows Phone 7 shares little with Windows 7 (though the phone GUI, was adapted to the PC and Windows Phone 8 shares with Windows 8)

Background

In contrast, the Android OS that runs on phones shares a large amount of code with the Linux that runs everywhere else, even on supercomputers

The iPhoneOS that runs on iPhones shares code with OS X

Windows Phone 7 shares little with Windows 7 (though the phone GUI, was adapted to the PC and Windows Phone 8 shares with Windows 8)

Phone OSs are marketed strongly on their GUIs, not the actual OS; many phone vendors try to make it hard for users to access the OS

Background

For the new M1 Macs their operating system had to be ported from x86 to ARM

Background

For the new M1 Macs their operating system had to be ported from x86 to ARM

And Apple provides Rosetta, a clever bit of software that dynamically translates x86 code to ARM code so that user applications written for the old hardware will still run on the new hardware

Background

For the new M1 Macs their operating system had to be ported from x86 to ARM

And Apple provides Rosetta, a clever bit of software that dynamically translates x86 code to ARM code so that user applications written for the old hardware will still run on the new hardware

With a moderate loss in speed

Background

You will often hear “Do you have a PC?” where the asker really means “Do you run Windows on your PC?”

Background

You will often hear “Do you have a PC?” where the asker really means “Do you run Windows on your PC?”

“PC” means “Personal Computer” and refers to the hardware

Background

You will often hear “Do you have a PC?” where the asker really means “Do you run Windows on your PC?”

“PC” means “Personal Computer” and refers to the hardware

I run Linux on my PC; many people run MacOS on their PCs

Background

You will often hear “Do you have a PC?” where the asker really means “Do you run Windows on your PC?”

“PC” means “Personal Computer” and refers to the hardware

I run Linux on my PC; many people run MacOS on their PCs

Such people get confused when I answer “Yes, I have a PC but it doesn’t run Windows”

Background

You will often hear “Do you have a PC?” where the asker really means “Do you run Windows on your PC?”

“PC” means “Personal Computer” and refers to the hardware

I run Linux on my PC; many people run MacOS on their PCs

Such people get confused when I answer “Yes, I have a PC but it doesn’t run Windows”

“Or MacOS”

Background

You will often hear “Do you have a PC?” where the asker really means “Do you run Windows on your PC?”

“PC” means “Personal Computer” and refers to the hardware

I run Linux on my PC; many people run MacOS on their PCs

Such people get confused when I answer “Yes, I have a PC but it doesn’t run Windows”

“Or MacOS”

Macs are PCs, too

Background

The understanding of the general public is such that they get the hardware and software confused

Background

The understanding of the general public is such that they get the hardware and software confused

And, more, they are usually thinking about the GUI, not the OS

Background

The understanding of the general public is such that they get the hardware and software confused

And, more, they are usually thinking about the GUI, not the OS

Some software companies encourage and make capital out of this confusion

Background

There is some reason for this confusion: Microsoft and Apple tie their GUIs indivisibly to their OSs

Background

There is some reason for this confusion: Microsoft and Apple tie their GUIs indivisibly to their OSs

Other OSs, notably Linux, allow the user a choice of many UIs and GUIs all running on the same OS kernel

Background

There is some reason for this confusion: Microsoft and Apple tie their GUIs indivisibly to their OSs

Other OSs, notably Linux, allow the user a choice of many UIs and GUIs all running on the same OS kernel

And typical users base their choices on GUIs, not OSs

Background

There is some reason for this confusion: Microsoft and Apple tie their GUIs indivisibly to their OSs

Other OSs, notably Linux, allow the user a choice of many UIs and GUIs all running on the same OS kernel

And typical users base their choices on GUIs, not OSs

But we shouldn't be talking about GUIs: this Unit is about Operating Systems, taking the programmer's point of view

Background

As programmers, we should choose an OS for the features that it provides:

Background

As programmers, we should choose an OS for the features that it provides:

- Ease of use

Background

As programmers, we should choose an OS for the features that it provides:

- Ease of use
- Efficiency

Background

As programmers, we should choose an OS for the features that it provides:

- Ease of use
- Efficiency
- Security

Background

As programmers, we should choose an OS for the features that it provides:

- Ease of use
- Efficiency
- Security
- Stability

Background

As programmers, we should choose an OS for the features that it provides:

- Ease of use
- Efficiency
- Security
- Stability
- Suitability for the task in hand

Background

As programmers, we should choose an OS for the features that it provides:

- Ease of use
- Efficiency
- Security
- Stability
- Suitability for the task in hand
- And so on

Background

Just a few recent operating systems:

Background

Just a few recent operating systems:

- XP/Vista/Windows 7/Windows 8/Windows 10 from Microsoft. Large, resource intensive, highly featured, previously Intel processor only, now Intel and ARM

Background

Just a few recent operating systems:

- XP/Vista/Windows 7/Windows 8/Windows 10 from Microsoft. Large, resource intensive, highly featured, previously Intel processor only, now Intel and ARM
- OS X from Apple. Large, not quite so intensive, highly featured. Based on BSD (Unix), on Intel and ARM processors (Earlier: PowerPC)

Background

Just a few recent operating systems:

- XP/Vista/Windows 7/Windows 8/Windows 10 from Microsoft. Large, resource intensive, highly featured, previously Intel processor only, now Intel and ARM
- OS X from Apple. Large, not quite so intensive, highly featured. Based on BSD (Unix), on Intel and ARM processors (Earlier: PowerPC)
- Unixes. Solaris from Sun/Oracle, IRIX from SGI, AIX from IBM, OSF/1 from DEC, etc. Large to medium.

Background

Just a few recent operating systems:

- XP/Vista/Windows 7/Windows 8/Windows 10 from Microsoft. Large, resource intensive, highly featured, previously Intel processor only, now Intel and ARM
- OS X from Apple. Large, not quite so intensive, highly featured. Based on BSD (Unix), on Intel and ARM processors (Earlier: PowerPC)
- Unixes. Solaris from Sun/Oracle, IRIX from SGI, AIX from IBM, OSF/1 from DEC, etc. Large to medium.
- Unix derivatives (reimplementations). Various BSD (including MacOS X), Linux, Hurd, etc. From small to large.

Background

Just a few recent operating systems:

- XP/Vista/Windows 7/Windows 8/Windows 10 from Microsoft. Large, resource intensive, highly featured, previously Intel processor only, now Intel and ARM
- OS X from Apple. Large, not quite so intensive, highly featured. Based on BSD (Unix), on Intel and ARM processors (Earlier: PowerPC)
- Unixes. Solaris from Sun/Oracle, IRIX from SGI, AIX from IBM, OSF/1 from DEC, etc. Large to medium.
- Unix derivatives (reimplementations). Various BSD (including MacOS X), Linux, Hurd, etc. From small to large.

Note, since the advent of smartphones, of the above OSs, Unix derivatives (Linux) are the most popular

Background

- Phones. Palm OS, Symbian, Windows CE/Mobile/Phone 7/Phone 8, Android, iPhone OS

Background

- Phones. Palm OS, Symbian, Windows CE/Mobile/Phone 7/Phone 8, Android, iPhone OS
- Experimental. Minix, Plan 9, Mach, Singularity, Amoeba, etc.

Background

- Phones. Palm OS, Symbian, Windows CE/Mobile/Phone 7/Phone 8, Android, iPhone OS
- Experimental. Minix, Plan 9, Mach, Singularity, Amoeba, etc.
- Networking. NetWare (Novell), Cisco IOS, DD-WRT etc. For controlling networking hardware

Background

- Phones. Palm OS, Symbian, Windows CE/Mobile/Phone 7/Phone 8, Android, iPhone OS
- Experimental. Minix, Plan 9, Mach, Singularity, Amoeba, etc.
- Networking. NetWare (Novell), Cisco IOS, DD-WRT etc. For controlling networking hardware
- Distributed OSs. Management of collections of computers, or making a collection appear as a single large computer

Background

- Embedded. μCLinux, Windows Embedded, RTOS, etc. Small, resource frugal

Background

- Embedded. μCLinux, Windows Embedded, RTOS, etc. Small, resource frugal
- Real-time. QNX, μCLinux, etc. For controlling systems where a fast response is critical

Background

- Embedded. μCLinux, Windows Embedded, RTOS, etc. Small, resource frugal
- Real-time. QNX, μCLinux, etc. For controlling systems where a fast response is critical
- OSs for gadgets (MP3 players, etc.) Conservation of battery power is the largest problem

Background

- Embedded. μCLinux, Windows Embedded, RTOS, etc. Small, resource frugal
- Real-time. QNX, μCLinux, etc. For controlling systems where a fast response is critical
- OSs for gadgets (MP3 players, etc.) Conservation of battery power is the largest problem
- Other. OS/2, MacOS 9, RISC OS, BeOS, z/OS (IBM). Various sizes

Background

- Embedded. μCLinux, Windows Embedded, RTOS, etc. Small, resource frugal
- Real-time. QNX, μCLinux, etc. For controlling systems where a fast response is critical
- OSs for gadgets (MP3 players, etc.) Conservation of battery power is the largest problem
- Other. OS/2, MacOS 9, RISC OS, BeOS, z/OS (IBM). Various sizes

Again, remember that embedded OSs outnumber PC OSs by an order of magnitude

Background

- Embedded. μCLinux, Windows Embedded, RTOS, etc. Small, resource frugal
- Real-time. QNX, μCLinux, etc. For controlling systems where a fast response is critical
- OSs for gadgets (MP3 players, etc.) Conservation of battery power is the largest problem
- Other. OS/2, MacOS 9, RISC OS, BeOS, z/OS (IBM). Various sizes

Again, remember that embedded OSs outnumber PC OSs by an order of magnitude

And we’ve not even mentioned historical OSs yet

Background

There are *lots* of operating systems out there, most we don’t notice

Background

There are *lots* of operating systems out there, most we don't notice

The ones we do notice are failing in their purpose!

ARX project Arthur OS RISC OS AmigaOS Amiga Unix AEGIS Domain/OS vikek OS Apple DOS UCSD Pascal ProDOS GS/OS SOS Lisa OS Newton OS Mac OS 8 Mac OS 9 A/UX MkLinux Mac OS X v10.x iOS Atari DOS Atari TOS Atari MultiTOS XTS-400 BeOS Blue Eyed OS Cosmoe GCOS Burroughs MCP COS SIPROS SCOPE MACE KRONOS NOS NOS/BE RDOS AOS DG/UX CTOS DOS Deos HeartOS CP/M DR-DOS OS/8 ITS TOPS-10 WAITS TENEX TOPS-20 RSTS/E RSX-11 RT-11 VMS Domain/OS TSB Digital UNIX HP-UX Ultrix Guardian OSS OSE Towns OS Google Chrome UTX-32 INTEGRITY HDOS HT-11 HP-UX HP MIE OLERT-E Multics HeartOS DEOS iRMX ISIS-II BESYS CTSS GM OS GM-NAA I/O IBSYS IJMON SOS UMES OS/360 OS/VS SVS OS/VSn MVS/SE OS/390 z/OS DOS/360 z/VSE CP/CMS VM/370 VM/XA VM/ESA z/VM AIX/370 OpenSolaris UTS z/Linux BOS/360 MTS MUSIC/SP ORVY WYLBUR PC DOS/IBM DOS OS/2 J MultiJob GEORGE 2/3/4 TME ICL VME iVideOS LynxOS MicroC/OS-II Xenix MS-DOS Windows Singularity Midori TMX NetWare MontaVista RTXC

□