Adam Cohen

CS340- Homework 1

2/17/14

Professor Fluture

**PART A:**

Microsoft Windows began as a simple command line interface operating system, named MS-DOS (Microsoft Disk Operating System), released to the public in 1981. Unfortunately for the excited average consumer, MS-DOS lacked a graphical user interface (GUI) and thus was very complex to interact with. Microsoft’s leaders, Paul Allen, Bill Gates and Steve Balmer, quickly got to working on an operating system that would allow the user to interact in a more graphic friendly way. This led to the idea of “Interface Manager”, announced in 1983, which later got renamed to Windows to describe the components that the user interacts with.

Windows 1.0, released in 1985, redefined how users interacted with personal computers emphasizing the word personal in the term personal computers. The Windows 1.0 shell was an extension of MS-DOS called MS-DOS Executive. The operating system included multiple programs and intuitive drop-downs, scrollbars, icons and dialog boxes. In addition the operating system allowed for multiple programs to be open simultaneously although not on top of one-another. Windows 2.0, released in 1987, allowed for expanded memory and improved graphics (overlapping windows) in addition to some keyboard shortcuts and control panel. To accommodate for the different Intel chipsets Windows released it’s 2.1 in two versions, 286 and 386 which both greatly advanced memory management. Specifically these early Windows versions allowed for multitasking with several applications and, through the use of virtual memory, allowed applications to run that were larger than the available memory. Windows 3.0 focused even greater on the visual aspect of the operating system in addition to releasing an SDK for developers to release software for Windows. Windows 95 introduced the first Windows Internet Browser, support for 32-bit applications, plug and play hardware compatibility. It also introduced the much recognized start menu, minimize, maximize and close buttons. Windows 98 included improvements to the user interface and compatibility with DVDs and USB disks. Windows ME, the last DOS-based Windows version, improved user experience through Windows Media Player, Movie Maker, and System Restore among other software.

Windows XP, released in late 2001, was the longest lasting OS for Windows thus far which utilized Windows NT architecture. The redesigned look and feel along with the increased security, help and support, and different editions allowed Windows XP to become one of the most popular Windows OS’s ever. Microsoft released Windows Vista in early 2007 with a strong focus on security and design. Vista is quickly replaced due to extensive consumer criticism with Windows 7 in 2009. Windows 7 allowed for users to interact via multi-touch screens for ease of use and included a redesigned shell and improvements to performance. Finally the release of Windows 8 in late 2012 was the first Windows OS to introduce the Metro design specifically designed for mobile and tablet devices. This Metro design includes tiled apps within a Start menu along with the native desktop view from older Windows versions. Recently Microsoft released Windows 8.1 to address some tweaks needed to Windows 8.

Mac OS began with Steve Jobs in 1981 along with other engineers in an effort to build one of the first operating system with a graphical user interface. Mac OS’s first version was simply called System with a revision number following. The System versions 1-4 were unable to run multiple programs at once. Finder was used to navigate the file system and is still used as the main file system explorer on Mac OS nowadays. System 2.1 introduced the use of a Hierarchical File System structure vs. a flat file system structure. System 3.0 added support for the SCSI standard and the new file server AppleShare. System 4.1 released on the Macintosh II, the first Apple computer with support for color, added support for multiple new hardware technologies such as expansion slots and the Apple Desktop Bus. System 5 introduced multitasking that gave control to background applications so long as the foreground application was not utilizing resources. System 6 added support for two major hardware releases, the 68030 processor and 1.44 MB SuperDrive. System 7 was released in 1991, around the time the internet and Microsoft exploded. System 7 included virtual memory support and 32-bit memory addressing. In terms of software System 7 included shortcuts, extension manager, a scripting language called AppleScript, QuickDraw and TrueType. Beginning with System 7.1, Apple began charging for its operating system.

Mac OS 7.6 was the first Apple operating system to have the Mac OS naming scheme. Mac OS 8 introduced a Finder that allowed for files to be copied in the background and skins for the user to customize the look. Mac OS 9 introduced many new features including basic multi-user support, Sherlock search engine, Apple Software Update and Keychain. Improvements included memory management/implementation, AirPort wireless support, and AppleScript networking control. OS X (10) dropped the Mac from the operating system naming convention. OS X is the first Apple operating system to be Unix-based. OS X v10.2 brought many performance improvements to the original but lacking 10.0 and 10.1 versions. 10.2 included Quartz Extreme (an enhanced graphics feature), an Address Book and a chat client called iChat. 10.3 included multiple features such as iChat videoconferencing, FileVault, Safari, Expose, Fast user switching among others. 10.4 introduced Spotlight, Dashboard, Smart Folders, Automator, VoiceOver and QuickTime and Safari updates. 10.5 brought features such as Time Machine for retrieving previous file versions and Boot Camp which enabled users to install Windows alongside the Mac OS on their computers. 10.6 was mainly a back-end update that didn’t reflect too many noticeable graphical or application enhancements. Many of the core functionalities and applications were rewritten in order to increase efficiency such as Finder and Safari. 10.6 also introduced the Mac App Store. 10.7 included a new feature called Launchpad which displayed all the users’ applications. It also included auto-hiding scrollbars, full-screen applications and Mission Control which combined several previous features. 10.8 introduced many features to Mac OS X that existed in iOS (Apple’s mobile Operating System) such as Notification Center, iMessage, Game Center, iCloud and reminders.

**PART B:**

The Android operating system is an open-source operating system based on the Linux kernel and is primarily loaded onto mobile devices with ARM architecture. Android’s user interface is built specifically for touch based devices and thus utilizes swiping, tapping, pinching to manipulate the space. In addition Android utilizes hardware commonly found in mobile devices such as the accelerometer, gyroscope, proximity sensors and vibration motors.

Android allows for live widgets and application shortcuts on the home screen and allows for application generated notifications. Android allows for vibrational feedback as a response to user input called haptic feedback. Games on android devices are able to utilize the gyroscope to mimic for example the steering of a racecar. Google Play is a native Android app that acts as a market and allows users to download other third-party apps.

The application framework of Android allows for replacement and reuse of existing components. Android applications are written primarily in the Java programming language along with the Android SDK. Android comes loaded with many Google applications such as Gmail, Calendar, Maps, Web Search, Chrome, Google+ and many more.

**PART C:**

A Real Time Operating System is a specific type of operating system that is intended to process incoming data without a buffer delay. Requirements for processing times are measured in 1/10 of seconds. The time it takes to accept and complete an application’s task is extremely consistent in RTOS and the variability is called jitter. RTOS’s are categorized into soft and hard performance category which translates to a high jitter vs a lower jitter respectively. The algorithm used on RTOS’s for scheduling is extremely complex which focus on minimal interrupt latency and minimal thread switching latency. The key quality of a RTOS is its speed/predictability of response rather than quantity of work accomplished. A very strong emphasis is put on time constraints with RTOS’s especially with the hard RTOS category where it’s considered “unacceptable” or a “failure” if the system doesn’t complete a task within its constraint. Thus RTOS’s require multitasking, process threads that can be prioritized and many interrupt levels. RTOS’s allow programmers access to task prioritizations and deadline confirmation for this very reason. Some examples of Real Time Operating Systems are LynxOS, OSE, QNX, RTLinux, VxWorks and WindowsCE.

Embedded systems are computer systems with a specific function sometimes within a larger system, hence the name embedded. In this way embedded systems are not flexible or very user oriented, although they could be part of a user-end device, such as a PC. Embedded systems usually don’t allow for user interactions as a computer does but still has the core components that a computer has namely a processor, software and I/O. In other words embedded systems are not always standalone devices, they often consist of small, computerized parts within a larger device. The fact that the embedded systems are specific function systems, the system design engineers are able to strongly focus on optimizing it to reduce cost and size while increases performance and reliability. Some embedded systems work within the real time operating system performance constraints for safety, security and usability. Embedded systems generally do not need entire operating systems to run and thus use program instructions called firmware which are stored in ROM or flash memory chips. The systems run with limited hardware resources and do not allow much direct user interaction. There are many examples of embedded system in modern life. They’re used in telephones, cell networks and Wi-Fi routers. Simple consumer electronics such as clock radios, DVD players, GPS receivers and printers include embedded systems as well. Medical devices are examples of embedded system use as well with defibrillators and insulin pumps.

Embedded systems are not the same as real-time systems. Many embedded systems require real-time system constraints (as discussed in the RTOS section above). This means that if the embedded system is designed in order to guarantee that real-time application requests will be met within the deadlines that are predefined than it is considered “real-time”. Thus some embedded system require this constraint such as aircraft control and pacemakers. In this way embedded systems and real-time systems are different but rather real-time systems are within the category of embedded systems.

TSR (terminate and stay resident) processes refer to programs within DOS that can remain in memory once they’ve been loaded in order to be easily reactivated by a simple interrupt as opposed to the process being removed from the memory and having to be retrieved again. This was used in the times of DOS to overcome the limitation of single task execution at a time. The advantage of this is these processes are generally fast to load but the disadvantage is that many TSR’s at a time greatly reduce the memory availability to other processes and programs. In addition some TSR’s might not interact well with each other if loaded simultaneously in the memory. A current example of a TSR nowadays is a virus scanner that needs to remain loaded in memory to protect the computer from viruses. Another example of a TSR is a virus itself which would lurk and remain inside of the memory to actively infect your machine.

**PART D:**

An interrupt is a signal sent from either a hardware device or a program notifying the operating system that it needs to take care of something. Computers nowadays use interrupts all the time to ensure attention is given when needed. In this way a program that is currently running can be stopped in order for the operating system to be able to handle something else. This interrupt service is the key to multitasking nowadays. Although using interrupts isn’t exactly perfect multitasking, operating system engineers have designed them to act as though many things can happen at the exact same time but in reality there are many fast interrupts going on in the background. The process specifically starts with the current code or thread being suspended (temporarily saved) then an interrupt handler is executed to deal with the event. Once the event has been dealt with the processor resumes to execute the previous thread. There are two types of interrupts: A hardware interrupt could be a keyboard or mouse press while a software interrupt is an exception which could be caused for example in a case where divide by zero is done. All interrupts have a matching interrupt handler and all hardware interrupts initiations are called interrupt requests. Hardware interrupts are limited by the number of designated interrupt request lines to the processor while software interrupts don’t have a constraint. The interrupt request value is specific to each device thus before Plug and Play was introduced, interrupt request values had to be entered manually into the system in order for them to function properly. The reason why is because only one interrupt request can be handled at a time thus if there are identical values it could conflict. The interrupt requests are also assigned priorities in order to be as efficient as possible.

The trap instruction is used to respond to hardware signals (interrupts). UNIX executes the trap commands based on alphabetic/numeric order. The trap command allows a programmer/user to specify how to respond to specific hardware signals. In other words the trap command handles certain types of hardware signals such as the escape key in ways that the programmer is able to choose. As an example, a programmer could choose to handle an escape key with a text response saying you have hit the escape key.

**Sources Used:**

<http://en.wikipedia.org/wiki/History_of_Microsoft_Windows>

<http://en.wikipedia.org/wiki/History_of_Mac_OS>

<http://developer.android.com/about/index.html>

<http://www.engineersgarage.com/articles/what-is-android-introduction>

<http://en.wikipedia.org/wiki/Embedded_system>

<http://en.wikipedia.org/wiki/Real-time_operating_system>

<http://en.wikipedia.org/wiki/Terminate_and_stay_resident_program>

<http://en.wikipedia.org/wiki/Interrupt_request>

<http://en.wikipedia.org/wiki/Interrupt>

<http://whatis.techtarget.com/definition/IRQ-interrupt-request>

<http://www.techterms.com/definition/irq>

<http://www.computerhope.com/unix/utrap.htm>