Zhenyu Yan

Question 1 The Root of Mainstream OS Kernel Design:

Please read the 1992 Andrew Tanenbaum & Linus Torvald debate (see link).

a. What was Tanenbaum's initial complaint(s) about the Linux kernel?

   Linux is obsolete, and it is an monolithic system. Also Linux is not portability.

b. What problem(s) did Linus have with MINIX?

   Minix doesn't do the micro-kernel thing very well, and has problems with real multitasking, doing exploratory work under minix is painful，and system stuck to serving files and didn't interact with terminal i/o.

c. What is the conclusion - what is better - micro- or monolithic kernels (and why)?

   Micro is better, because First is Linux a monolithic style system. This is a giant step back into the 1970s. That is like taking an existing, working C program and rewriting it in BASIC, writing a monolithic system in 1991 is a truly poor idea. Second is portability, it is a gross error to design an OS for any specific architecture, since that is not going to be around all that long.

d. Read the "next" of the above and find out about open source and its contributors, interesting read. Add a paragraph on your thoughts after reading the link - thoughts should demonstrate some depth and that you read the text.

   “Hello netlanders, Due to a project I'm working on (in minix), I'm interested in the posix standard definition. Could somebody please point me to a (preferably) machine-readable format of the latest posix rules? Ftp-sites would be nice.” I found out this is very interesting. Because of nobody give an answer to him, He create Linux systems. The thing that very impress me is his personality. Although he is the father of linux, His personality won't let him take credit for something as his own. He delicate his own work to benefit the whole world.

e. Read the 2006/2008 (slash dot) and Part II of the debate - it is an ever ending debate, add another a paragraph on your thoughts after reading this text.

   Before reading this, I thought mac is monolithic because it is based on Linux. However Mac OS is actually micro kernels or you can say all the operating system right now beside Linux is micro kernels or you can say it is a hybrid kernels just Likes Symbian and windows NT in the part II of the debate. I think in the future hybrid kernel will dominate. Just like the L4Linux, future operating system allowed the TUD people to build new systems such top of L4 but still have access to full Linux without having to modify it for these new features. In this way, they can experiment with new facilities and still run legacy programs.

Question 2 Micro/Monolithic kernels of Mainstream Operating Systems.

a1) What are differences between Linux Mint, and Ubuntu?

Mint is base on Ubuntus. Here is a few differences.

1. Compatibility.
2. Linux Mint is lighter and faster than Ubuntu.
3. Linux Mint (other than the OEM version) is pre-installed. Ubuntu usually don't.
4. Interface differences.
5. Ubuntu has more (and arguably easier) installation options compared with Mint.

a2) Which Linux (above in a1) do you favor and why?

Ubuntu, the majority reason is I have never use Linux Mint at all. Also I do not like pre-installed. Every time I brought a windows laptops, I need to uninstall a lot of application after I turn it on and also when I use recovery, all the pre-installed application is there again, I hate that very much.

a3) Are these micro or monolithing kernels?

These are micro kernels.

b1) What is the latest version of Mac?

OS X 10.7 Lion (Barolo).

b2) Is it a micro or a monolithic kernel?

It is a micro kernel.(or you can say hybrid kernel?)

b3) Outline the timeline of the Mac system through major designs changes (e.g., in terms of being micro or monolithic)

Version	Date	Corresponding releases	Notes
0.1	March 16, 1999	Mac OS X Server 1.0 releases	• Initial release • 0.1 is contrived (for sorting and identification) as this identified itself simply as Rhapsody 5.3
0.2	April 14, 1999		Mac OS X Server 1.0.1
0.3	August 5, 1999		Based on Rhapsody 5.5 • ISO image is available on archive.org • After this point the kernel changed from the NeXTSTEP/OPENSTEP/Rhapsody to the newer XNU for Mac OS X
1.0	April 12, 2000		Developer preview 3 • ISO image is available on archive.org
1.1	April 5, 2000	Developer preview 4	
1.2.1	November 15, 2000	Mac OS X Public Beta	Code-named "Kodiak"
1.3.1	April 13, 2001	Mac OS X v10.0	• Code-named "Cheetah" • First commercial release of Darwin • All releases of Cheetah (v10.0.0–4) had the same version of Darwin.

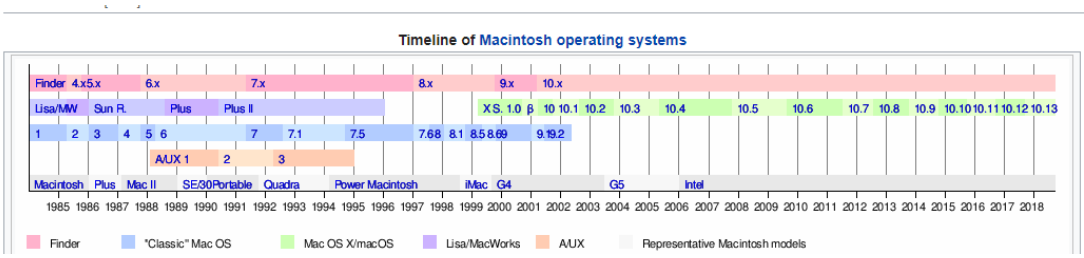
1.4.1	October 2, 2001	Mac OS X v10.1	• Code-named "Puma" • Performance improvements to "boot time, real-time threads, thread management, cache flushing, and preemption handling" • Support for SMB network file system • Wget replaced with cURL.[17]
5.1	November 12, 2001		Mac OS X v10.1.1 • Change in numbering scheme to match the Mac OS X build numbering scheme
5.5	June 5, 2002		Mac OS X v10.1.5
6.0.1	September 23, 2002	Mac OS X v10.2	• Code-named "Jaguar" • GCC upgraded from 2 to 3.1 • IPv6 and IPSec support • mDNSResponder service discovery daemon (Rendezvous) • Addition of CUPS, Ruby, and Python • Journaling support in HFS+ (Darwin 6.2) • Application profiles ("pre-heat files") for faster program launching.[18]
6.8	October 3, 2003		Mac OS X v10.2.8
7.0	October 24, 2003	Mac OS X Panther	Mac OS X v10.3.0 • BSD layer synchronized with FreeBSD 5 • Automatic file defragmentation, hot-file clustering and optional case sensitivity in HFS+ • Bash instead of tcsh as default shell • Read-only NTFS support (Darwin 7.9)[19]
7.9	April 15, 2005		Mac OS X v10.3.9
8.0	April 29, 2005	Mac OS X Tiger Mac OS X for Apple TV	Mac OS X v10.4.0 • Mac OS X for Apple TV in Darwin 8.8.2 • Stable kernel programming interface, finer-grained kernel locking, 64-bit BSD layer • launchd service management framework

			• Extended file attributes, access control lists • Commands such as cp and mv updated to preserve extended attributes and resource forks[20]
8.11	November 14, 2007		Mac OS X v10.4.11
9.0	October 26, 2007	Mac OS X Leopard iPhone OS 1	Mac OS X v10.5.0 • iPhone OS 1 support in Darwin 9.0.0d1 • Full POSIX compliance, improved hierarchical process scheduling model, dynamically allocated swap files, dynamic resource limits (for files and processes), process sandboxing, address space layout randomization, DTrace tracing framework, file system events daemon, directory hard links • Apache 1.3 and PHP 4 updated to Apache 2.2 and PHP 5, read-only ZFS support.[21] • First Darwin core used for iPhone devices.
9.8	August 5, 2009		Mac OS X v.10.5.8
10.0	August 28, 2009	Mac OS X Snow Leopard iOS 4	Mac OS X v10.6.0 • End of official support for PPC architecture (although several fat binaries, such as Kernel, still contain PPC images) • 64-bit kernel and drivers • libdispatch task parallelization framework • OpenCL heterogeneous computing framework • Initial support for Automatic Reference Counting • Support for blocks in C • Transparent file compression in HFS+.[22]
10.8	June 23, 2011		Mac OS X v10.6.8
11.0.0	July 20, 2011	Mac OS X Lion iOS 5[23]	Mac OS X v10.7.0

			• XNU no longer supports PPC binaries (fat binary only for i386, x86_64). • XNU requires an x86_64 processor, except for iOS which is ARM based. • Improved sandboxing of applications • Complete support for Automatic Reference Counting
11.4.2	October 4, 2012		Mac OS X v10.7.5 (supplemental)
12.0.0	February 16, 2012	OS X Mountain Lion	OS X v10.8.0 • Mac OS X was rebranded into OS X. • Objective-C garbage collection was deprecated in favor of Automatic Reference Counting
12.6.0	January 27, 2015		OS X v10.8.5 (with Security Update 2015-001)
13.0.0	June 11, 2013	OS X Mavericks iOS 6	OS X v10.9.0 • Virtual memory compression • Timer coalescing • OpenGL 4.1 and OpenCL 1.2 • Server Message Block version 2 (SMB2) is now the default protocol for sharing files instead of AFP. This is to increase performance and cross-platform compatibility. • IPoTB (Internet Protocol over Thunderbolt Bridge). • The Open Transport API has been removed
13.4.0	September 17, 2014		OS X v10.9.5
14.0.0	September 18, 2014	OS X Yosemite iOS 7, iOS 8	OS X v10.10.0
14.5.0	August 13, 2015		OS X v10.10.5
15.0.0	September 16, 2015	OS X El Capitan	OS X v10.11.0 and iOS 9.0

		iOS 9	• System Integrity Protection. Protects certain system parts from being modified or tampered with by a process even if ran by root or by a user with root privileges. • sudo is configured with the "tty_tickets" flag by default, restricting the session timeout to the terminal session (such as a window or tab) in which the user authenticated the program. • LibreSSL replaces OpenSSL
15.6.0	July 18, 2016		OS X v10.11.6 and iOS 9.3.3
16.0.0	September 13, 2016	macOS Sierra iOS 10	macOS v10.12.0 and iOS 10.0.1 (initial release version) • OS X was rebranded into macOS. • Writing to `/Volumes` directory is now restricted to root user or any user with root priviliges • System Integrity Protection now covers `/Library/Application Support/com.apple.TCC` directory that contains a list of applications that are allowed to "control the computer" • Objective-C garbage collector removed and replaced by Automatic Reference Counting that was introduced with Darwin v12.0 (OS X v10.8). Objective-C applications that use garbage collection will no longer work. • Native support for PPTP was removed.
16.5.0	March 27, 2017		macOS v10.12.4 and iOS 10.3 • Changed filesystem from HFS+ to APFS on iOS devices. APFS is already available on macOS since 10.12.0 but can't be used on boot partition.
16.6.0	July 19,		macOS v10.12.6 and iOS 10.3.3

	2017		
17.0.0	September 19, 2017	macOS High Sierra iOS 11	• APFS replaces HFS+ as the default filesystem for boot partition in macOS on Macs with flash storage. On Macs with HDDs, the boot partition must be reformatted to use APFS. • ntpd replaced by timed as a time synchronization service • FTP and telnet commands are removed. • Kernel extensions ("kexts") will require explicit approval by the user before being able to run.
17.5.0	March 29, 2018		macOS 10.13.4 • Support for external graphics processors using Thunderbolt 3, and removes support for external graphics processors using Thunderbolt 1 and 2.
17.7.0	July 9, 2018		macOS v10.13.6 and iOS 11.4.1
18.0.0	Fall 2018 ETA	macOS Mojave iOS 12	



(wikipedia)

c1) What is the latest version of Windows?

Windows 10.

c2) Is it a micro or a monolithic kernel?

It is a monolithic kernel. (or you can say hybrid kernel?)

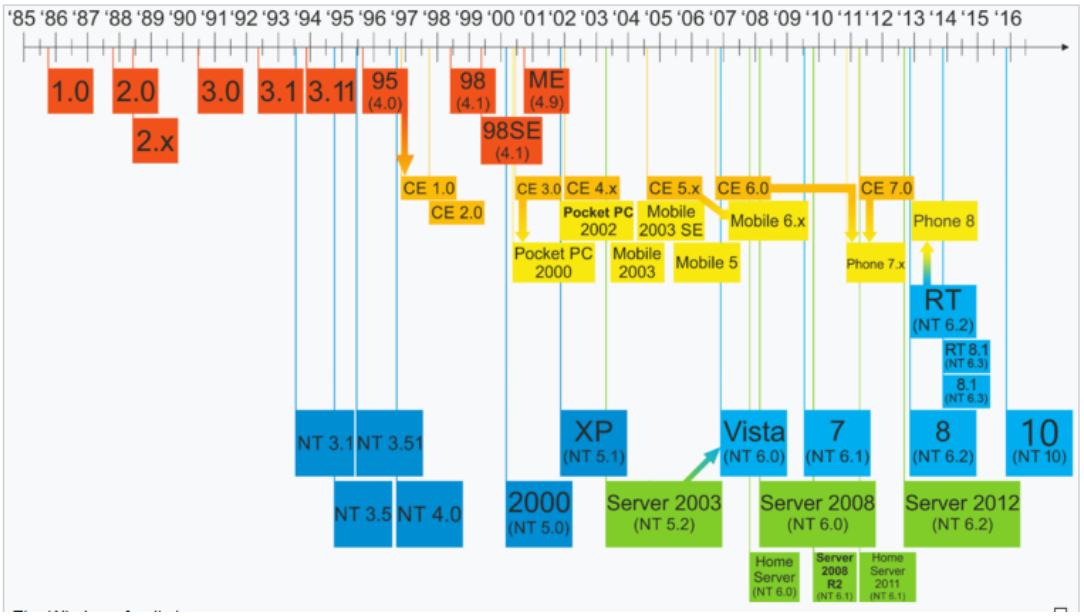
c3) Outline the timeline of Windows going through major designs (e.g., in terms of being micro or monolithic)

Windows NT releases				
Version	**Marketing name**	**Editions**	**Release date**	**Build number**
3.1	Windows NT 3.1	Workstation (named just *Windows NT*), Advanced Server	July 27, 1993	528
3.5	Windows NT 3.5	Workstation, Server	September 21, 1994	807
3.51	Windows NT 3.51		May 30, 1995	1057
4.0	Windows NT 4.0	Workstation, Server, Server Enterprise Edition, Terminal Server, Embedded	August 24, 1996	1381
5.0	Windows 2000	Professional, Server, Advanced Server	February 17, 2000	2195
		Datacenter Server	September 26, 2000	
5.1	Windows XP	Home, Professional, Media Center (original, 2004 & 2005), Tablet PC (original and 2005), Starter, Embedded, Home N, Professional N	October 25, 2001	2600

Windows NT releases				
Version	**Marketing name**	**Editions**	**Release date**	**Build number**
	Windows Fundamentals for Legacy PCs	N/A	July 8, 2006	
5.2	Windows XP	64-bit Edition Version 2003[24]	March 28, 2003	3790
	Windows Server 2003	Standard, Enterprise, Datacenter, Web, Storage, Small Business Server, Compute Cluster	April 24, 2003	
	Windows XP	Professional x64 Edition	April 25, 2005	
	Windows Server 2003 R2	Standard, Enterprise, Datacenter, Web, Storage, Small Business Server, Compute Cluster	December 6, 2005	
	Windows Home Server	N/A	July 16, 2007	
6.0	Windows Vista	Starter, Home Basic, Home Premium, Business, Enterprise, Ultimate, Home Basic N, Business N	Business: November 30, 2006 Consumer: January 30, 2007	6000 (RTM) 6001 (SP1) 6002 (SP2)
	Windows Server 2008	Foundation, Standard, Enterprise, Datacenter, Web	February 27, 2008	6001 (RTM)

Windows NT releases				
Version	**Marketing name**	**Editions**	**Release date**	**Build number**
		Server, HPC Server, Itanium-Based Systems[25]		6002 (SP2)
6.1[26]	Windows 7	Starter, Home Basic, Home Premium, Professional, Enterprise, Ultimate[27]	October 22, 2009[28]	7600 (RTM) 7601 (SP1)
	Windows Server 2008 R2	Foundation, Standard, Enterprise, Datacenter, Web Server, HPC Server, Itanium-Based Systems		
	Windows Home Server 2011	N/A	April 6, 2011	7600 (RTM)
6.2	Windows 8[29]	Windows 8, Windows 8 Pro, Windows 8 Enterprise, Windows RT[30]	October 26, 2012[31]	9200
	Windows Server 2012[32]	Foundation, Essentials, Standard, Datacenter[33]	September 4, 2012	
6.3[34]	Windows 8.1	Windows 8.1, Windows 8.1 Pro, Windows 8.1 Enterprise, Windows RT 8.1	October 18, 2013	9600[35]
	Windows Server 2012 R2	Foundation, Essentials, Standard, Datacenter		
10.0[36]	Windows 10	Home, Pro, Pro Education, Enterprise, Education, Windows 10 S, IoT Core, Mobile, Mobile Enterprise[37][38]	July 29, 2015	10240 (TH1) 10586 (TH2) 14393 (RS1) 15063 (RS2) 16299 (RS3) 17134 (RS4)

Windows NT releases				
Version	**Marketing name**	**Editions**	**Release date**	**Build number**
	Windows Server 2016	Essentials, Standard, Datacenter, Multipoint Premium Server, Storage Server, Hyper-V Server	September 26, 2016	14393 (RS1) 16299 (RS3)[39]



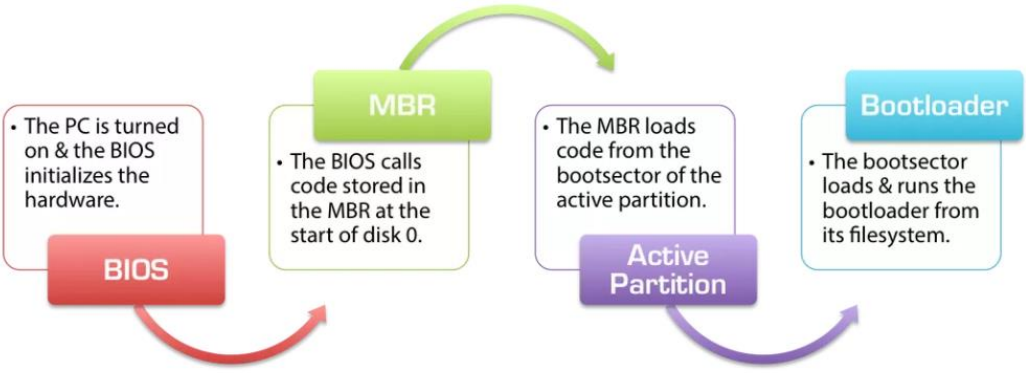
(Wikipedia)

Question 3 (Booting)

a. Outline the boot sequence with simple and brief bullet points - each step should be>
one or two brief sentence[s]. Include a diagram

- The first step of any boot process is applying power to the machine.
- The next step in the boot process is called the POST, or power on self test.

- The I/O system is essential to the operation of the computer because it defines the rules for communications between the CPU and the other devices attached to the computer via the motherboard.
- Once the hardware functionality is confirmed and the input/output system is loaded, the boot process begins loading the operating system from the boot device.
- Once the previous steps are complete and the operating system is safely loaded into RAM, the boot process relinquishes control to the OS.(alan)



(NeoSmart Knowledgebase)

b. What is a typical size of the Master Boot Record - Why this constraint?

2 TiB (232 × 512 bytes). The reason is easy to break compatibility with existing boot loaders and most MBR-compliant operating systems and system tools, and can cause serious data corruption when used outside of narrowly controlled system environments.(Wikipedia)

Some useful web pages as a starting point:
http://en.wikipedia.org/wiki/Booting
http://www.goodells.net/multiboot/

Question 4 (System Call)

Contrast the mechanics of a system call in Linux and Microsoft Windows OS (you can pick any Windows System XP and newer). Highlight differences and similarities. You may discuss references/resources on the class list (Piazza).

In UNIX the system calls results in the creation of kernel objects which returns an unsigned integer, whereas in Windows. windows API are used to create the kernel

objects which return data type called handle. handle represents objects and are used in identifying the specific object.

In windows, process control is CreateProcess ExitProcess and waitForSignalOject. In linux is fork,exit and wait.

For file manipulation, in windows is CreateFile, ReadFile, WriteFile and CloseHandle. In Linux is open, read, write, close.

(Unix and Windows Kernel Comparison and System Calls)

Question 5

What are differences between standard (ANSI) C (current version C11), and POSIX C Library? Include discussing, open(), close(), threads, fopen() and fclose(), threads, and signals.

The C POSIX library is a specification of a C standard library for POSIX systems. It was developed at the same time as the ANSI C standard.(Wikipedia) POSIX is a superset of the standard C library.
open() and close() is a system call and specific to Unix based system and it returns a file descriptor. fopen() and fclose() is an ANSI C function call which returns a file pointer and it is portable to other OS.
For thread, ANSI C doesn't define threading, but there are various libraries available. POXIX is phreads.(wikidedia)
For signal, stand c only have 6 signal, SIGABRT,SIGFPE,SIGIL,SIGINT,SIGSEGV, SIGTERM..POSIX signal has 19 signals.(Linux man page)

submit Q/A as a pdf file on nike in a folder called HW2.

submit HW2 csx730

Reference

Linux Mint vs Ubuntu. (n.d.). Retrieved from https://www.diffen.com/difference/Linux_Mint_vs_Ubuntu

MacOS version history. (2018, August 14). Retrieved from https://en.wikipedia.org/wiki/MacOS_version_history#Timeline

Timeline of Microsoft Windows. (2018, August 22). Retrieved from https://en.wikipedia.org/wiki/Timeline_of_Microsoft_Windows

Master boot record. (2018, August 14). Retrieved from

https://en.wikipedia.org/wiki/Master_boot_record

The BIOS/MBR Boot Process. (2015, February 28). Retrieved from https://neosmart.net/wiki/mbr-boot-process/

C POSIX library. (2018, August 22). Retrieved from https://en.wikipedia.org/wiki/C_POSIX_library

Unix and Windows Kernel Comparison and System Calls. (n.d.). Retrieved from https://www.scribd.com/document/349680783/Unix-and-Windows-Kernel-Comparison-and-System-Calls

Alan, hughus. The Five Steps of the Boot Sequence. (n.d.). Retrieved from https://www.techwalla.com/articles/the-five-steps-of-the-boot-sequence