CS 537: Introduction to Operating Systems
Fall 2023: Midterm Exam #1

This exam is closed book, closed notes.

No calculators may be used.

You have 1 hour 15 minutes to complete this exam.

Write all of your answers on the accu-scan form with a #2 pencil:

- CANVAS LAST NAME - fill in your last (family) name starting at the leftmost column
- CANVAS FIRST NAME - fill in the first five letters of your first (given) name
- IDENTIFICATION NUMBER - This is your UW Student WisCard ID number
- ABC of SPECIAL CODES - Write your lecture number as a three digit value 001, 002, or 010.

These exam questions must be returned at the end of the exam, but we will not grade anything in this booklet.

You may separate the pages of this exam if it helps you.

Unless stated (or implied) otherwise, you should make the following assumptions:

- The OS manages a single uniprocessor (single core)
- All memory is byte addressable
- The terminology lg means $\log_2$
- 2^{10} bytes = 1KB
- 2^{20} bytes = 1MB
- Page table entries require 4 bytes
- Data is allocated with optimal alignment, starting at the beginning of a page
- Assume leading zeros can be removed from numbers (e.g., 0x06 == 0x6).
- Hex numbers are represented with a proceeding “0x”

This exam has 60 questions. Each question has the same number of points.

Good luck!

Virtualizing the CPU

Designate if the statement is True (A) or False (B).

1. The **CPU dispatcher** determines the policy for which process should be run and when.

2. With **cooperative multitasking**, it is possible for a process to keep running on the CPU for as long as it chooses.

3. When executing the **return-from-trap** instruction, hardware restores the user process's registers from the kernel stack, changes to user mode, and jumps to a new code location.

4. When **an I/O operation completes**, the previously blocked process moves into the RUNNING state.

5. The **fork**() system call clones the calling process and executes a new program in the child process.

6. A **FIFO scheduler** has lower average turnaround time when long jobs arrive after short jobs, compared to when short jobs arrive after long jobs.

7. An **SJF scheduler** may preempt the currently running job.

8. An **SJF scheduler** can suffer from the convoy effect where a lot of short-running jobs can get stuck behind a long-running job.

9. A **RR scheduler** may preempt the currently running job.

10. An **STCF scheduler** cannot cause jobs to starve.

11. If all jobs arrive at the same time, an SJF and an STCF scheduler will behave the same.

12. A **process** is another name for a program.

13. When the OS relinquishes control of the CPU to a process, it retains control by setting the kernel bit.

14. The OS provides the initial arguments to a new process by placing them in the heap.

15. **Limited Direct Execution** consists of privilege levels for user and kernel modes and a timer interrupt and handler to regain control of the CPU.

16. A **trap** into a system call switches from user to kernel modes.

17. A **context switch** occurs when a process performs a system call.

18. Different processes accessing the same address within their virtual address space see the same contents.

19. The **exec()** call returns a value of zero on success.

20. The purpose of **boosting in MLFQ** is to ensure no processes starve.

Select the one best answer, A - E.

Consider three processes with the following execution trace. All IO takes 5 time slices and each process needs 3 CPU time slices to complete running.

```
./process-run.py -l 3:50,3:50,3:50
```

Time	PID: 0	PID: 1	PID: 2	CPU	IOs
1	RUN:io	READY	READY	1	
2	BLOCKED	RUN:cpu	READY	1	1
3	BLOCKED	RUN:io	READY	1	1
4	BLOCKED	BLOCKED	RUN:io	1	2
5	BLOCKED	BLOCKED	BLOCKED		3
6	BLOCKED	BLOCKED	BLOCKED		3
7*	RUN:io_done	BLOCKED	BLOCKED	1	2
8	RUN:io	BLOCKED	BLOCKED	1	2
9*	BLOCKED	RUN:io_done	BLOCKED	1	2
10*	BLOCKED	RUN:cpu	READY	1	1

21. What will the state of process 0 be in the next time slice?

A. DONE B. BLOCKED C. RUN:io_done D. RUN:io E. READY

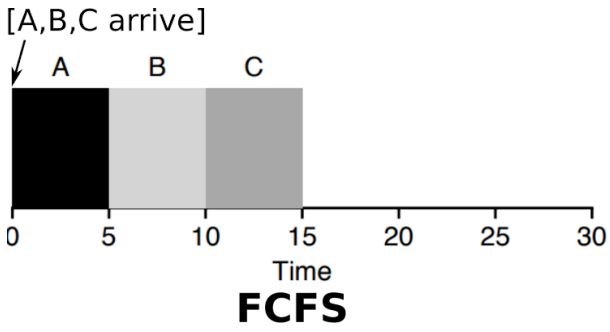
22. What will the state of process 1 be in the next time slice?

A. DONE B. BLOCKED C. RUN:io_done D. RUN:io E. READY

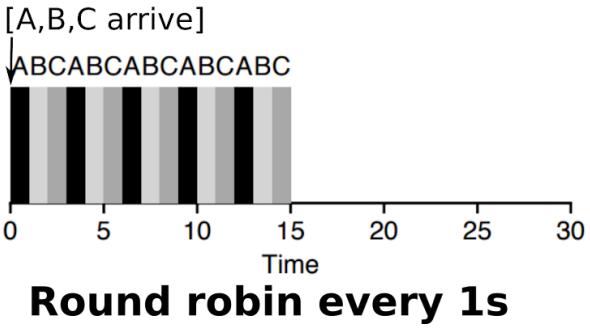
23. What will the state of process 2 be in the next time slice?

A. DONE B. BLOCKED C. RUN:io_done D. RUN:io E. READY

Consider the following graphs showing the scheduling of three processes, A, B, and C using different scheduling policies.



FCFS



Round robin every 1s

24. What is the turnaround time for process B using FCFS?

A. 5 B. 7.5 C. 10 D. 12.5 E. 15

25. What is the response time for process B using FCFS?

A. 5 B. 7.5 C. 10 D. 12.5 E. 15

26. What is the turnaround time for process C using RR?

A. 5 B. 7.5 C. 10 D. 12.5 E. 15

27. What is the average response time using FCFS?
A. 1 B. 5 C. 10 D. 14 E. 15

28. What is the average time to completion using RR?
A. 1 B. 5 C. 10 D. 14 E. 15

29. Which of the following is NOT possible for the MLFQ scheduler?
A. MLFQ learns which jobs are long running.
B. MLFQ may starve jobs.
C. MLFQ may use different length time slices for jobs
D. MLFQ may use round robin
E. MLFQ may forget what it has learned about jobs

30. Which of the following is NOT true about a MLFQ scheduler?
A. It can preempt running jobs
B. It may change the priority of a process if the boost time has transpired
C. Jobs with higher priority preempt jobs of lower priority
D. It knows the runtime of jobs before scheduling them
E. Jobs with the same priority run using RR

Virtualizing Memory

Designate if the statement is True (A) or False (B).

31. One goal in **memory virtualization** is to ensure that the OS's method of virtualization is transparent to the processes.

32. The major difference between **static and dynamic relocation** is who converts a virtual address to a physical address -- The OS or hardware.

33. One **shortcoming of static relocation** is that it provides no protection of one process accessing memory of another process.

34. **Base+Bounds** is one type of static relocation.

35. **Segmentation** enables sparse allocation of an address space.

36. **Base+Bounds** provides no protection of one process accessing memory of another process.

37. In **segmentation**, to convert a virtual address to a physical address, the offset is always added to the base register for the segment.

38. The **address space** is another name for physical memory.

39. In an address space of 16 KB with a page size of 4 KB, the VPN uses 2 bits of the virtual address.

40. One of **paging's major advantages** is that it makes memory management easier because all of physical memory is divided into fixed sized units called virtual page numbers.

41. Using **ASIDs in the TLB** means fewer TLB misses may occur after a context switch.

42. In a **hardware-managed TLB**, the MMU raises an exception on a TLB miss.

43. The **valid bit** in the TLB indicates if there is a mapping between the VPN and the PFN. If it is not set then a segmentation fault should be raised.

44. The **present bit** in the PTE indicates if the mapping between the VPN and the PFN is present in the page table. If it is not present then a segmentation fault should be raised.

45. A **page fault** occurs when a process tries to access a page of memory that is outside its address space.

46. When converting a virtual address to a physical address using a **multi-level page table** the first thing the MMU consults is the page directory.

47. In a **multi-level page table** the VPN is divided into a portion for the page directory index and a portion for indexing into a page of the page table.

48. In a paging memory model, if the **page size increases** but the address space stays the same size then the number of entries in the page table must increase.

49. The **clock algorithm** attempts to approximate the least-recently used algorithm.

50. **Thrashing** is when memory demands exceed physical memory and the system is constantly paging.

Select the one best answer, A - E.

Segment Base/Bounds Translation

Consider a virtual address space of size 1K and physical memory of size 16K with the base and limit(bounds) registers for each segment as shown below:

Segment 0 base (grows positive): 0x00001603 (decimal 5635)
Segment 0 limit : 274

Segment 1 base (grows negative) : 0x00002468 (decimal 9320)
Segment 1 limit : 408

Convert each virtual address to its physical address or select segmentation violation if it is out-of-bounds. For these problems, you should assume a simple address space with two segments: the top-bit of the virtual address can thus be used to check whether the virtual address is in segment 0 (topbit=0) or segment 1 (topbit=1).

51. Virtual Address: **0x017f (decimal: 383)**
A. 0x167f (decimal: 5759)
B. 0x1768 (decimal: 5992)
C. 0x17c2 (decimal: 6082)
D. segmentation violation
E. none of the above

52. Virtual Address: **0x02f5 (decimal: 757)**
A. 0x235d (decimal: 9053)
B. 0x255d (decimal: 9565)
C. 0x275d (decimal: 10077)
D. segmentation violation
E. none of the above

53. Virtual Address: **0x03e7 (decimal: 999)**
A. 0x255d (decimal: 9565)
B. 0x17c2 (decimal: 6082)
C. 0x244f (decimal: 9295)
D. segmentation violation
E. none of the above

Size of Linear Page Table

What is the size of a linear page table for one process given the following assumptions? Remember, unless otherwise specified, assume PTEs are 4 bytes each.

54. **Bits in virtual address: 20, Page size: 4KB**
A. 2KB
B. 512 bytes
C. Situation not possible
D. Not enough information to determine page table size
E. None of the above

55. **Bits in virtual address: 20, Number of Physical Pages: 1024**
A. 4KB
B. 16KB
C. Situation not possible
D. Not enough information to determine page table size
E. None of the above

Multi-level Page Tables

Page Sizes are 32 bytes
Virtual Address space is 1024 pages (or 32 KB)
Physical Memory consists of 128 pages
A Virtual Address needs 15 bits (5 for the offset, 10 for the VPN)
A Physical Address needs 12 bits (5 for the offset, 7 for the PFN)

The system uses a multi-level page table. The upper five bits of a VA are the index into the page directory to get the PDE. If the PDE is valid, it points to a page of the page table. Each page holds 32 PTEs. If the PTE is valid, it holds the desired translation (PFN).

The format of a PTE is:
VALID | PFN6 ... PFN0

The format of a PDE is identical:
VALID | PT6 ... PT0

The PDBR holds the value: 0x51 (decimal: 81) [This means the page directory is held in this page]
The memory dump of the pages of memory are on the last pages of the exam.

56. When accessing the **virtual address 0x65d8**, what will be the first page accessed?
A. 0x65 (decimal: 101)
B. 0x19 (decimal 25)
C. 0xd8 (decimal 216)
D. 0x51 (decimal: 81)
E. Error or None of the above

57. What **index of the page directory** will be accessed?
A. 0x19 (decimal: 25)
B. 0x06 (decimal: 25)
C. 0x65 (decimal: 101)
D. 0x05 (decimal: 5)
E. Error or None of the above

58. What will be the **second page accessed**?
A. 0x0e (decimal: 14)
B. 0x65 (decimal: 101)
C. 0x7f (decimal: 127)
D. 0x25 (decimal: 37)
E. Error or None of the above

59. What are the **contents of the corresponding PTE** that will be read?
A. 0x7f (decimal: 127)
B. 0xfa (decimal: 250)
C. 0x00 (decimal: 0)
D. 0x93 (decimal: 147)
E. Error or None of the above

60. What is the **final physical address** for this virtual address?
A. 0xff8 (decimal: 4088)
B. 0xf38 (decimal: 3896)
C. 0x2b8 (decimal: 696)
D. 0x018 (decimal: 24)
E. Error or None of the above

hex offset	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
dec offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x00 (0):	7f	7f	7f	7f	d2	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	c6	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x01 (1):	10	08	1e	00	19	14	18	1c	03	02	04	05	10	19	08	0c	0b	0c	11	1e	07	15	1a	14	1a	17	02	0b	11	01	00	05
0x02 (2):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x03 (3):	7f	7f	7f	7f	bd	7f	7f	7f	7f	7f	7f	7f	7f	7f	87	7f	7f	a9	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	9f	90	7f
0x04 (4):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x05 (5):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x06 (6):	0d	19	0a	17	1d	0c	17	00	08	1e	0f	0b	1d	0d	15	14	02	13	18	16	02	04	16	13	16	09	03	00	09	0b	08	04
0x07 (7):	19	02	1a	1a	13	0f	0e	11	18	1b	0d	19	14	09	0e	04	01	03	1b	0a	16	0f	05	07	0d	0d	10	04	0b	08	0c	0a
0x08 (8):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x09 (9):	1a	03	17	19	1b	01	0a	17	04	0b	05	19	17	19	05	0d	0c	14	07	0d	08	17	0d	10	09	19	0e	19	0b	1d	1e	0e
0x0a (10):	01	0d	09	07	02	1d	19	11	1d	1e	14	08	01	17	0e	14	1c	05	12	13	0f	02	0a	0a	14	1a	0a	15	08	1d	19	11
0x0b (11):	09	17	05	04	10	1e	10	1c	19	07	19	0e	19	17	0a	03	1d	04	1d	1a	16	1e	1d	18	0b	18	00	10	0e	14	14	12
0x0c (12):	01	12	15	01	16	0c	07	06	1a	01	0f	08	19	16	09	12	14	09	09	04	14	06	09	01	1d	1b	1c	13	0d	0f	1e	1d
0x0d (13):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	d4	7f	7f	7f	fb	7f	7f	7f	7f	7f	7f	7f	c7
0x0e (14):	0f	1c	0a	16	19	19	07	04	06	12	17	14	05	17	0f	17	17	0d	1c	11	13	13	1a	13	04	02	0d	09	08	01	0f	09
0x0f (15):	1a	0e	1b	0d	02	15	1a	09	0a	02	10	1b	1a	16	1c	13	18	0f	03	06	04	18	00	11	0b	18	11	12	02	09	1e	16
0x10 (16):	0c	11	00	13	04	0e	01	0b	06	0a	17	0b	17	19	07	02	00	10	1e	0a	14	18	14	17	1d	06	00	04	03	14	11	06
0x11 (17):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	c5	7f	7f	c3	7f	7f	7f	7f	7f
0x12 (18):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x13 (19):	0b	17	0e	1a	09	15	18	1c	11	1e	11	04	07	06	14	1c	1a	02	16	05	08	14	12	1b	05	17	16	11	0e	1a	0a	1d
0x14 (20):	05	02	00	0d	12	09	07	15	15	0e	15	1c	18	13	14	1c	0d	10	14	1c	19	02	05	09	17	11	08	03	15	15	1d	0f
0x15 (21):	17	03	19	1d	03	00	09	14	1d	0c	16	02	15	13	03	17	1a	12	03	1e	18	0a	0d	0b	0f	0a	1a	19	03	1d	13	19
0x16 (22):	0b	15	0f	19	18	02	1a	01	00	1c	1a	11	11	15	0c	03	00	0a	18	13	19	1c	02	1a	07	12	10	0c	09	0a	0a	05
0x17 (23):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x18 (24):	18	1a	0a	03	0b	1b	17	1b	0b	1e	0f	0f	1c	10	18	16	02	12	19	10	09	02	14	09	12	0d	15	0a	01	1a	0a	1e
0x19 (25):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	8b	7f	7f	7f	7f	7f	7f	7f	a0	7f	7f	7f	7f	7f	7f
0x1a (26):	0a	0a	17	16	13	15	12	05	07	11	06	1e	09	08	06	15	09	0a	1c	18	08	03	14	0b	1e	19	1d	18	09	08	16	0a
0x1b (27):	0d	0a	0b	16	18	11	02	01	04	13	14	08	14	0f	0d	08	17	03	0d	08	15	0f	14	01	0c	12	00	09	06	04	07	0a
0x1c (28):	7f	7f	7f	7f	7f	e2	81	96	7f	7f	7f	7f	7f	7f	7f	f7	8c	7f	7f	7f	7f	7f	7f	7f	7f	a3	7f	7f	7f	7f	7f	9b
0x1d (29):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x1e (30):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x1f (31):	03	08	0b	13	07	1a	04	0a	11	09	17	0f	0f	0f	09	00	1d	0f	1d	06	0a	01	0f	1b	14	0e	10	1a	0d	1b	16	17
0x20 (32):	0b	0e	07	11	11	1e	09	1e	14	08	11	15	17	01	12	0f	1c	08	18	12	0a	13	13	15	16	14	19	13	1c	14	09	0d
0x21 (33):	0d	06	10	16	06	09	1e	14	0d	10	03	06	0a	12	07	06	02	13	07	1c	1a	02	07	14	06	04	1d	11	0e	18	19	05
0x22 (34):	1e	07	11	0b	1c	0f	1b	1a	08	18	0c	1c	0f	19	08	09	12	1e	0f	04	10	0a	11	10	0e	09	05	15	11	07	18	01
0x23 (35):	16	14	1b	02	0d	00	06	0f	00	06	14	10	06	12	19	00	18	15	0a	04	1d	0a	02	02	1a	12	19	16	10	1e	0b	11
0x24 (36):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x25 (37):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x26 (38):	02	0e	1e	1e	02	06	08	1c	1b	1b	0b	04	19	15	12	1e	14	00	19	09	14	1d	04	03	03	11	08	12	16	06	13	08
0x27 (39):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x28 (40):	7f	7f	7f	7f	7f	f0	7f	7f	7f	7f	89	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x29 (41):	1e	0c	0f	1e	14	10	0c	05	0b	17	13	17	06	11	1c	0d	15	03	1e	12	07	04	11	11	02	1e	1c	0e	03	19	0f	16
0x2a (42):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x2b (43):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x2c (44):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x2d (45):	7f	7f	7f	98	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x2e (46):	18	15	1d	04	12	18	01	02	18	0b	0b	11	12	15	1d	0b	17	11	10	0c	14	07	03	16	0f	0b	11	08	08	0d	1e	08
0x2f (47):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	e6	7f	7f	7f	c4	7f	7f	7f	7f	7f	7f
0x30 (48):	18	04	07	13	1b	11	03	1a	1a	08	17	08	1c	04	0d	1d	06	0d	0a	00	01	0f	07	1e	0b	00	1c	1a	14	18	04	08
0x31 (49):	7f	7f	7f	7f	7f	86	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	da	7f	7f	7f	7f
0x32 (50):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x33 (51):	11	06	11	02	0f	17	08	1e	15	03	1e	0c	18	0a	1d	17	06	0f	0f	01	04	0a	0e	0e	12	0f	0a	13	05	1e	16	09
0x34 (52):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x35 (53):	1e	0b	1a	1c	0b	15	14	10	14	0a	05	10	10	16	06	00	00	09	14	10	10	19	07	0a	08	1d	16	03	19	0c	17	1b
0x36 (54):	0e	16	14	1e	03	0c	0a	1a	07	05	0d	0d	08	07	1c	0d	1a	11	01	1e	19	1e	1c	1a	05	0f	06	0c	01	0b	1e	08
0x37 (55):	1d	1b	08	0f	05	1c	1a	09	13	12	04	17	10	18	10	00	0a	00	1c	1b	19	09	01	1b	1d	02	0f	02	17	17	03	0e
0x38 (56):	1a	03	15	10	1e	0b	0c	05	03	1a	0e	14	13	12	00	18	07	03	11	02	17	06	06	1a	0a	04	1b	00	1a	04	04	07
0x39 (57):	00	02	07	19	11	16	10	03	08	09	01	0d	18	0e	03	1c	12	00	0f	07	04	0d	13	07	0c	14	02	1e	02	10	0f	1e
0x3a (58):	7f	7f	7f	b6	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	b7	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x3b (59):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x3c (60):	13	1e	07	00	18	0a	16	13	17	16	0a	01	10	19	05	18	0e	15	13	19	01	18	0e	09	01	06	01	1c	0f	1e	10	07
0x3d (61):	06	10	1c	15	04	13	12	07	14	10	13	01	0c	16	03	17	00	11	1c	0c	1c	1b	0e	06	1d	09	14	1c	02	11	10	16
0x3e (62):	02	06	15	09	12	0e	10	0d	17	0a	15	08	07	03	05	03	10	17	05	06	0f	16	1e	10	08	03	06	07	05	00	10	08
0x3f (63):	15	07	19	1d	13	0f	05	18	05	16	0f	1c	10	13	01	01	1a	1d	14	17	0c	1a	07	15	00	0f	0b	13	14	13	0e	0f

hex offset	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
dec offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x40 (64):	11	04	0a	10	03	06	12	02	0f	19	0e	0f	0e	01	0e	19	16	0c	19	17	11	01	0a	01	1e	1c	02	1c	00	0c	17	17
0x41 (65):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x42 (66):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x43 (67):	0e	12	03	15	1a	14	17	16	06	0d	07	0a	0e	0c	02	0d	14	0b	04	1c	02	19	02	02	16	19	11	12	11	0a	03	0a
0x44 (68):	14	1e	12	18	16	15	00	0e	1d	18	18	11	16	12	05	13	13	1a	04	15	00	1d	03	00	09	04	15	0c	18	1c	1b	16
0x45 (69):	07	15	15	01	0c	10	0c	06	0d	1c	12	15	1a	17	0b	00	0a	17	1a	1d	0c	17	10	12	06	06	0d	00	0a	15	0c	05
0x46 (70):	08	0d	11	14	0e	0d	06	0e	1b	18	05	02	0f	13	0a	19	17	14	06	06	00	07	0e	1a	02	0c	13	06	15	0f	07	14
0x47 (71):	0c	00	0a	02	12	0a	0b	09	0c	02	1b	1c	1e	11	05	0b	04	09	0f	01	0d	0d	0f	02	07	07	13	12	06	03	09	1d
0x48 (72):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	eb	7f	7f
0x49 (73):	7f	7f	7f	7f	7f	d9	7f	7f	7f	b5	7f	ae	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x4a (74):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x4b (75):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x4c (76):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x4d (77):	7f	7f	7f	7f	7f	7f	7f	7f	b9	7f	7f	7f	95	7f	7f	7f	7f	7f	7f	7f	dd	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x4e (78):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x4f (79):	11	15	01	07	08	02	07	03	08	13	0b	0b	06	08	1d	14	12	05	16	05	0b	1e	13	11	15	1a	18	07	00	09	08	06
0x50 (80):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x51 (81):	9c	f4	7f	ba	c8	8d	7f	7f	91	e8	af	f9	d3	7f	83	ec	ea	ef	99	80	e5	e0	cd	b1	ad	ff	ee	7f	7f	e7	a8	c9
0x52 (82):	02	02	15	08	16	14	1c	1b	0a	12	04	0a	1d	15	0c	12	1d	09	0b	18	19	14	19	16	15	10	14	0d	0b	0b	05	06
0x53 (83):	7f	7f	7f	7f	bf	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x54 (84):	00	00	18	07	0a	05	01	16	10	17	0e	18	0f	03	0f	1d	01	18	1a	10	0e	1d	01	0e	0c	15	0f	1c	02	02	12	02
0x55 (85):	05	04	0a	07	08	07	17	1d	09	16	00	14	15	01	03	09	0c	0f	1b	15	09	03	1c	09	13	06	04	04	17	12	0c	11
0x56 (86):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x57 (87):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x58 (88):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x59 (89):	0e	0d	05	01	1d	0e	19	0c	02	13	01	04	11	09	1e	03	17	12	18	06	10	0d	0d	1a	1e	09	13	12	16	1d	06	06
0x5a (90):	14	19	1e	0f	01	0f	12	1a	1b	0d	10	0e	16	0c	14	04	0e	1e	0a	15	14	1a	1a	1a	0b	09	16	17	1b	01	02	13
0x5b (91):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x5c (92):	1e	12	0a	11	06	14	06	03	1a	0b	17	11	19	1a	1e	19	13	13	00	1c	19	08	05	15	09	0a	00	1a	11	0c	04	13
0x5d (93):	1d	08	19	10	0e	0d	16	08	1a	19	02	1b	07	0e	12	0b	00	1a	05	06	18	0a	1b	15	08	00	1d	02	16	0f	17	15
0x5e (94):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x5f (95):	0d	03	13	1b	0d	15	1c	01	18	09	0b	04	10	11	18	05	02	1a	13	07	1c	04	0e	07	07	00	18	1b	15	04	0d	0a
0x60 (96):	7f	7f	7f	7f	94	7f	7f	7f	7f	dc	7f	7f	7f	7f	7f	a6	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x61 (97):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x62 (98):	03	0d	05	1d	10	01	07	1a	0e	18	14	1e	12	1d	1b	12	16	0f	19	10	1b	17	0e	08	07	13	17	10	13	08	02	08
0x63 (99):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x64 (100):	11	1a	02	10	01	06	1a	1b	0e	01	02	1c	1b	11	01	1c	09	1d	12	17	16	0c	02	05	07	19	0c	1b	0a	17	04	1e
0x65 (101):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	be	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x66 (102):	0a	0a	19	1a	1e	04	09	1d	06	09	1d	1e	09	15	0f	11	07	0b	19	0c	03	11	12	10	15	11	1d	0e	15	0d	09	15
0x67 (103):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	8f	7f	7f	7f	7f	df	7f	7f	7f	7f	7f	7f	7f	7f	fc	7f	7f	7f	7f	7f	7f	f5
0x68 (104):	b0	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x69 (105):	02	1e	13	0d	15	0a	1b	18	13	05	1d	0d	1c	01	03	04	05	0a	15	0a	1d	1b	1a	07	13	11	03	09	10	0f	05	1d
0x6a (106):	7f	7f	7f	7f	7f	7f	7f	7f	7f	a2	8e	7f	7f	7f	7f	d5	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	e9	7f	7f	7f	7f	7f
0x6b (107):	19	00	01	13	11	14	17	0c	13	0f	13	08	1d	0e	18	15	09	02	01	0d	0f	06	12	09	16	16	1a	1e	04	0b	11	09
0x6c (108):	7f	7f	c0	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	f3	7f	7f	7f	7f	8a	7f	7f	7f	e4
0x6d (109):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x6e (110):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	f8	7f	7f	7f	7f	7f	7f	cf	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f
0x6f (111):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	a1	7f	7f	7f	7f	bc	7f	7f	7f	7f	7f	7f	7f	7f
0x70 (112):	14	0c	1c	0e	08	07	11	08	12	1b	0c	06	1e	0f	02	01	03	13	18	0d	01	0b	1e	10	1e	1a	00	16	15	10	08	13
0x71 (113):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x72 (114):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x73 (115):	02	0d	08	00	17	13	08	16	11	0d	00	02	1b	1c	10	19	12	04	03	09	1b	18	1a	1b	06	07	03	18	1b	0c	13	04
0x74 (116):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	9a
0x75 (117):	1e	0c	0f	1d	1d	1c	1b	00	11	03	1e	08	1e	10	0f	1d	1a	0e	05	03	05	0e	07	05	16	18	11	17	05	1a	1b	19
0x76 (118):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x77 (119):	1d	03	1e	0e	07	12	06	1c	11	18	0b	10	0b	08	0f	0f	03	1e	0e	1a	1c	0b	0c	11	06	04	08	1c	11	0c	04	0a
0x78 (120):	03	07	00	1b	11	1c	06	01	19	1c	09	0c	04	1d	09	0f	03	1b	04	0e	14	17	1d	0c	17	04	0c	03	0f	0c	1d	01
0x79 (121):	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	b3	7f	7f	7f	7f	b8	7f	7f	7f	7f
0x7a (122):	11	16	04	09	1e	11	10	17	01	12	0f	1a	04	1d	02	05	12	14	07	03	1b	07	12	13	0c	12	10	1c	06	16	07	0c
0x7b (123):	0d	16	09	12	0f	0b	11	07	15	00	1c	10	16	17	14	0b	02	14	0a	09	1a	16	09	09	0c	0c	09	03	0d	1d	14	1b
0x7c (124):	1b	00	16	0a	1c	18	1a	19	08	18	03	1b	1a	06	19	0e	09	18	07	00	05	0a	1a	1d	08	13	0c	1e	10	1d	03	1e
0x7d (125):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x7e (126):	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x7f (127):	7f	7f	fa	7f	93	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f	7f