

Unit Exam: Persistence

Total points 140/215

Take this exam alone. It is closed book, closed notes, but feel free to use a single 8.5" x 11" sheet of paper (both sides) of notes. Take the exam only once.

The respondent's email (**parth2@pdx.edu**) was recorded on submission of this form.

✓ OS developers generally assume that I/O is slow *

5/5

☒ True



☐ False

In our simplified model of I/O devices, what types of registers are contained by a canonical I/O device? *

	Yes, it is part of a canonical I/O device	No, not part of a canonical I/O device	Score	
data register	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
DMA register	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
command register	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
floating point register	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
status register	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
stack pointer	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓



✓ Which statement best characterizes "Interrupt-driven I/O"? *

5/5

- ☐ the CPU gets an asynchronous notification after each individual I/O operation completes
- ☐ the I/O device can handle all of the data transfer itself and only notify/interrupt the CPU when the complete sequence of transfers is complete.
- ☒ the CPU can do other work and not block while I/O operations complete ✓
- ☐ the CPU synchronously interacts with the I/O device and controls the timing of each I/O operation

✓ Which statement best characterizes DMA (direct memory access) I/O"? * 5/5

- ☒ the I/O device can handle all of the data transfer itself and only notify/interrupt the CPU when the complete sequence of transfers is complete. ✓
- ☐ the CPU synchronously interacts with the I/O device and controls the timing of each I/O operation
- ☐ the CPU can do other work and not block while I/O operations complete
- ☐ the CPU gets an asynchronous notification after each individual I/O operation completes



Which type of I/O is preferred in each case? *

	Programmed I/O	Interrupt- driven I/O	DMA	Score	
when the I/O device is capable of transferring large amounts of data	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
when the I/O device is very slow	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
when interrupts are rare	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
when the excessive I/O interrupts might slow down the system	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	1/1	✓
when the I/O device is trusted to make independent memory reads and writes	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
when the I/O device has very fast bursts	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	1/1	✓



Why do OS kernel developers dislike device drivers? *

	reason for OS developer to dislike device drivers	not a reason for OS developers to dislike device drivers	Score	
there are so many of them	<input type="radio"/>	<input checked="" type="radio"/>	0/1	✗
Device Drivers make it easy to integrate new I/O devices with existing OSs	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
Device Drivers are a common source of crashes	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
device drivers are written by many different developers, many of whom are not OS kernel developers	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
Device Drivers can handle both character devices (like mouse and keyboard) and block devices (like HDDs and SSDs)	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓

Correct answers

	reason for OS developer to dislike device drivers	not a reason for OS developers to dislike device drivers
there are so many of them	<input checked="" type="radio"/>	<input type="radio"/>



Match each advantage with each type of storage device *

	SSDs	HDDs	Score	
faster	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
longer lasting data integrity	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
quieter	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
less expensive (per byte stored)	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
random reads/writes are just as fast as sequential reads/writes	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
no moving parts	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
larger total capacity	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓

Rank each item from smallest to biggest (as they relate to flash chips) *

	smallest	bigger	biggest	Score	
plane	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	2/2	✓
block	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	2/2	✓
page	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	2/2	✓



✓ Data written to a flash chip will tend to degrade if the chip is left without power for 1-2 years. * 5/5

☐ True

☒ False



Feedback

the class lecture said that this is True, but the actual research seems to be inconclusive and contradictory. All seem to agree that if flash-based storage devices are stored at low temperatures then they retain data much, much longer.



What must we do before we update (program) a page of a flash chip (assuming the page's contents are currently/previously valid)? *

	Yes, must be done prior to updating a page of a flash chip	No, not necessarily done prior to updating a page of a flash chip	Score	
sacrifice a chicken at dawn	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
copy the block containing the to-be-updated page to a temporary location	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
erase the entire block containing the page	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
copy the page's data to a temporary location	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
erase the page	<input checked="" type="radio"/>	<input type="radio"/>	0/1	✗
Correct answers				
	Yes, must be done prior to updating a page of a flash chip	No, not necessarily done prior to updating a page of a flash chip		
erase the page	<input type="radio"/>	<input checked="" type="radio"/>		



✗ An SSD will typically keep logical blocks located in same physical flash chip blocks to optimize for spatial locality and wear leveling. * 0/5

☒ True

✗

☐ False

Correct answer

☒ False

Feedback

False. SSDs will tend to move a logical block to varying flash memory locations each time it is updated. moving logical disk blocks around to varying physical locations achieves wear-leveling.

✓ SSDs tend to present the exact same client interface as is used for Hard Disk Drives * 5/5

☒ True. this is done for backward compatibility and interoperability

✓

☐ False. SSDs present a flash-aware programming interface that is optimized for performance and reliability of flash-based devices



✓ An SSD's flash controller (specifically the flash translation layer) will commonly program pages within an erased block in order, from low page to high page to achieve what goal? *

5/5

- ☐ wear leveling
- ☐ write spreading
- ☒ minimize disturbance
- ☐ improve performance
- ☐ prevent data corruption



Why does an SSD include volatile memory (in addition to its persistent flash storage chips)? *

	yes, a reason for including volatile memory on an SSD	no, not a reason for volatile memory	Score	
improve performance	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
reduced cost	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
provide a temporary storage for page updates	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
caching of persistent data	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
persistent storage of data	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
reduce disturbance	<input checked="" type="radio"/>	<input type="radio"/>	0/1	✗
mapping tables for logical/physical translation	<input type="radio"/>	<input checked="" type="radio"/>	0/1	✗

Correct answers

	yes, a reason for including volatile memory on an SSD	no, not a reason for volatile memory
reduce disturbance	<input type="radio"/>	<input checked="" type="radio"/>
mapping tables for logical/physical translation	<input checked="" type="radio"/>	<input type="radio"/>



✓ Spreading writes across the blocks of the flash chips as evenly as possible is known as _____* 5/5

- ☒ wear leveling ✓
- ☐ flash dancing
- ☐ reprogramming
- ☐ write spreading

Match each OS concept with the physical entity that it virtualizes*

	CPU	Physical Memory	Persistent Storage Device	Score	
File System	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	2/2	✓
Virtual Memory	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	2/2	✓
Process	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	2/2	✓



Match each linux system call with its purpose. *

	unlink()	open()	lseek()	rename()	stat()	link()	Score	
create a new location for a file without changing old name	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	0/2	✗
move a file to a different location	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	0/2	✗
jump to a new location within a file	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	0/2	✗
open a file for reading or writing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	0/2	✗
remove a file	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	2/2	✓
get a file's metadata	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	2/2	✓
create a file	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	2/2	✓

Correct answers

	unlink()	open()	lseek()	rename()	stat()	link()
create a new location for a file	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>



without
changing
old name

move a file
to a
different
location

☐

☐

☐

☒

☐

☐

jump to a
new
location
within a file

☐

☐

☒

☐

☐

☐

open a file
for reading
or writing

☐

☒

☐

☐

☐

☐



Which of the following are usually found in a linux file's metadata? *

	Yes, usually found in a linux file's metadata	No, not a common metadata item for linux files	Score	
date of deletion	<input checked="" type="radio"/>	<input type="radio"/>	0/1	✗
access control permissions	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
revision history	<input checked="" type="radio"/>	<input type="radio"/>	0/1	✗
date of creation	<input checked="" type="radio"/>	<input type="radio"/>	2/2	✓
reference to parent	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
name	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
owner	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
favorite Beatles tune	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
mother's maiden name	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓
time of last update	<input checked="" type="radio"/>	<input type="radio"/>	1/1	✓
list of update times	<input checked="" type="radio"/>	<input type="radio"/>	0/1	✗
favorite ice cream flavor	<input type="radio"/>	<input checked="" type="radio"/>	1/1	✓

Correct answers

	Yes, usually found in a linux file's metadata	No, not a common metadata item for linux files
date of deletion	<input type="radio"/>	<input checked="" type="radio"/>

revision history

☐☒

list of update times

☐☒

✓ In Linux, many items (processes, physical memory, network interface, devices, etc.) can be represented with "special" files in the file system. * 5/5

☒ True☐ False

✗ Linux directories are implemented with files containing a list of the contents of the directory. * 0/5

☒ True☐ False

Correct answer

☒ False

Feedback

False. This was true in the early days of Unix, but with modern linux systems the directories are implemented in various other ways to optimize for performance, reliability, size and scalability. Typically, each different type of file system implements its directories in a unique way while conforming to the requirements of the Linux filesystem APIs.



✓ Why does Linux need the fsync() system call? *

5/5

- ☐ To increase performance
- ☐ To decrease wear out of SSD flash chips
- ☒ To force recently updated file data to be written to persistent storage. ✓
- ☐ To implement looser consistency guarantees

✗ The _____ linux system call allows you to remove a storage device's files and directories from the system's directory structure. * 0/5

- ☐ mount()
- ☐ link()
- ☐ unmount()
- ☒ unlink() ✗

Correct answer

- ☒ unmount()



✗ The _____ linux system call allows you to remove a file. *

0/5

- ☐ unmount()
- ☐ link()
- ☐ mount()
- ☐ unlink()
- ☒ rm()

✗

Correct answer

- ☒ unlink()

✗ If we use a free list to keep track of free blocks, then how big is the free list? *

0/5

- ☒ the free list will have one element (one reference) per storage block
- ☐ it depends on how much of the disk storage is utilized or free

✗

Correct answer

- ☒ it depends on how much of the disk storage is utilized or free



✗ When keeping track of all of the allocated storage blocks for a given file, one rarely used strategy is to use contiguous allocation. Why is this technique rarely used? * 0/5

- ☐ External Fragmentation
- ☒ Internal Fragmentation

✗

Correct answer

- ☒ External Fragmentation

✗ Identify one significant disadvantage of using a linked list for file block allocation tracking * 0/5

- ☐ Simple to implement
- ☐ for HDDs, random access into the file is slow
- ☐ difficult to grow/shrink the file after it is created
- ☐ external fragmentation
- ☒ when used with SSDs linked-list allocation causes bit disturbance across adjacent flash memory pages

✗

Correct answer

- ☒ for HDDs, random access into the file is slow

File Allocation Table

Consider the following (small) file allocation table for a FAT filesystem. A "Cluster" is a contiguous group of 4 512 byte blocks. The special value 0xffff indicates the terminating block of a file.



File Allocation Table (FAT)

Cluster	Next
2	0xffff
3	5
4	6
5	0xffff
6	7
7	2
8	3

Details for the file access table (FAT)

Each cluster contains 4 blocks and each block contains 512 bytes



✖ If file A begins at cluster 4 then how many clusters does it contain? * 0/5

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

☒ Other: 16



Correct answer

☒ 4

Feedback

file A comprises clusters 4, 6, 7 and 2.



✗ if file A begins at cluster 4 and currently uses 50% of the space in its terminating cluster then how large is file A? * 0/5

☒ 2048 bytes

✗

☐ 8192 bytes

☐ 1792 bytes

☐ 7168 bytes

☐ Other: _____

Correct answer

☒ 7168 bytes

Feedback

file A complete fills 3 clusters (4, 6 and 7). each of those clusters has 4 blocks.

and then it fills one half of the fourth cluster (cluster 2) so 2 more blocks for a total of $3 \times 4 + 2 = 14$ blocks. Each block is 512 bytes, so we have:

$((3 \times 4) + 2) \times 512 = 14 \times 512 = 7168$ bytes



✗ file B ends at cluster 5 and currently uses only 1 of the 4 blocks within cluster 5. How large is file B? * 0/5

☒ 6144 bytes

✗

☐ 4096 bytes

☐ 4608 bytes

☐ 512 bytes

☐ Other: _____

Correct answer

☒ 4608 bytes

Feedback

if file B ends at cluster 5 then it also comprises clusters 3 and 8. each cluster is 4 blocks in size. the file only uses 1/4 of cluster 5. It therefore uses:

$$(2 * 4 * 512) + (1 * 512) = 4608 \text{ bytes}$$



✗ file B terminates at cluster 5 then how many clusters are used (either fully or partially) by file B? *

☐ 1

☐ 2

☐ 3

☐ 4

☒ 5

✗

☐ Other: _____

Correct answer

☒ 3

Feedback

File B uses clusters 8, 3 and (at least part of) 5.

✓ In linux it is possible to give a file an extension (such as .c, .o, .py, etc.) even though that file does not contain the corresponding type of information. *

5/5

☒ True

✓

☐ False



✗ When memory mapping a file, the OS must reserve virtual memory space equal to the size of the specified region of the file. * 0/5

☐ True

☒ False

✗

Correct answer

☒ True

Feedback

*True. The OS does not need to reserve the full amount of *physical* memory space, but the *virtual* address space needs to be reserved so that the program can read or write any byte in the mapped file.*

✓ If one process updates a file with write() and another process calls read() on the same file then can linux automatically make the first process's updates available to the second process's read() call without first writing to persistent storage? * 5/5

☒ Yes. The linux block buffer cache allows updates to be done in memory and subsequent reads to be satisfied from the cache

✓

☐ Nope, not gonna happen.

✓ If an application writes to a file multiple times in succession, does each write() call cause the data to be updated in persistent storage? * 5/5

☐ Yes

☒ Not always. Many of the updates will be done in the Linux block.buffer cache

✓



✓ If a storage block is found to be not assigned to a file and not on the free 5/5
list then what should the file system consistency checker do? *

- ☐ delete duplicate free list entries
- ☐ remove the block from the free list
- ☒ assign the block to the free list
- ☐ allocate a new block and copy contents to the new block



✓ If a storage block is found to be both assigned to a file and on the free 5/5
list then what should the file system consistency checker do? *

- ☐ allocate a new block and copy contents to the new block
- ☒ remove the block from the free list
- ☐ delete duplicate free list entries
- ☐ assign the block to the free list



✓ If a storage block is found to be on the free list multiple times then what 5/5
should the file system consistency checker do? *

- ☐ remove the block from the free list
- ☐ allocate a new block and copy contents to the new block
- ☒ delete duplicate free list entries
- ☐ assign the block to the free list



✓ If a storage block is found to be assigned to more than one file then what 5/5 should the file system consistency checker do? *

- ☐ remove the block from the free list
- ☒ allocate a new block and copy contents to the new block
- ☐ delete duplicate free list entries
- ☐ assign the block to the free list



Feedback

the OS should also assign the newly copied block to one of the files in place of the shared block. it should also notify the owners of the file sbecause this it is likely their data is corrupted and requires manual inspection.

This form was created inside of Portland State University.

Google Forms

