CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn
DIT401	Operating systems

Anonymous code Anonym kod	Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DIT401-0007-LLZ	21 October 2023	07	G

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	✓	27	
2	✓	12	
3	✓	10	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen		49	

[redacted]

5

(a) since A and B contain the same files, a difference in internal fragmentation must be due to the different block sizes. In general, we would expect a disk with a smaller block size to have less internal fragmentation (A).

We can conclude that A would have more blocks used compared to B.

If we were to access the last block of the disk, while the disk is currently pointing to the first block, then A would have to scan through more blocks than B during the seek operation, and hence A would take a longer time to perform read/write operations; B performs read/write operations faster.

4

(b) yes. we assume that ~~the~~ blocks are used and a hard disk is used. internal fragmentation occurs when a file is smaller than the total size of blocks assigned. For example, when a 11 KB file is allocated 3 blocks with block size of 4KB, resulting in 1KB of wastage.

~~At the same time, external fragmentation is also possible. File A can be initially allo in a contiguous allocation algorithm, File A, B, C can be a initially allocated blocks 1-5, 6-8, and 9-12 respectively. If file B were to be deleted, blocks 6-8 would be freed. Subsequently if file D were to request for have a size that requires 5 blocks, then the next contiguous block of storage available would be blocks 13-17, while blocks 6-8 remain empty, resulting in external fragmentation.~~

In a hard disk, external fragmentation is possible when there are "holes" ~~in the~~ between blocks. For example, file A, B, C are initially written to blocks 1-5, 6-8 and 9-12 respectively. Then file B is subsequently deleted. ~~a~~ If file A and file C were read one after another subsequently, the disk would have to spend extra time in the seek operation scanning past the empty blocks 6-8 to reach file C.

4

CHALMERS	Anonymous code	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr 02
	Anonym kod DIT401 - 0007 - LLZ	Poäng på uppgiften (ifylles av lärare)	Question no. Uppgift nr 01

(c) It depends. If Jessica has explicitly ~~set~~ denied the rwx permissions for some or all files for the group which Michael belongs to, and if Michael is in the same group of that particular file, then he will not have ~~a~~ access. 4

If the group rwx permissions of the particular file / directory ~~is not~~ does not explicitly deny Michael's group, then he is free to read and modify all of Jessica's files. Furthermore if any file explicitly denies Michael, he will also be unable to read them.

(d) A kernel thread is likely ? to use more memory than a user-level thread, resulting in less frames available for the ~~process to~~ thread to use. With less memory, it is possible that the page fault frequency would increase as the thread attempts to bring in the required pages to memory. In a multi-threaded environment, it can also lead to a higher chance of thrashing due to the reduced memory availability, where Σ(locality size) > total memory. $\emptyset$

As such, a user-level thread which has less overhead is likely to be better performing.

☆ sorry! Part (e) is on page 3 (next page)

(f) CPU utilization does not necessarily indicate that processes are run in parallel, or concurrently.

If we know how many cores the CPU has we are able to determine whether processes are run in parallel. If the CPU only has one core, then the processes are run one after another. Otherwise they are run in parallel.

To know if processes are run concurrently, we need to know ~~if~~ ~~the~~ there is preemption. If there is no preemption of processes, then they are not run concurrently. If each process is given a quantum, and preempted if they ~~are~~ do not complete in time, then there is concurrent execution.

I'm asking about the degree of parallelism $\emptyset$

5

CHALMERS	Anonymous code / Anonym kod	Points for question (to be filled in by teacher) / Poäng på uppgiften (ifylles av lärare)	Consecutive page no. / Löpande sid nr
	DIT401-0007-LLZ		03
			Question no. / Uppgift nr 01

(g) Mechanism 1: Dirty bit

Using the dirty bit, when there is insufficient memory, and a page needs to be discarded, the CPU can tell whether the page has been modified. If the page is not dirty, then the CPU can discard that page without having to spend time writing that page to disk, reducing time spent.

Mechanism 2: valid/invalid bit.

No, it does not reduce

With the valid/invalid bit, if the page requested by the task has ~~an~~ a valid bit, then we know that the page is already in memory and there is no need to access the disk. Otherwise extra time is spent retrieving the ~~pag~~ page from disk.

Mechanism 3: page numbers. ?

1

With page numbers, the MMU spends lesser time performing the page table walk since ~~the pag~~ the search time is reduced to O(log N), where ~~it~~ it is able to perform binary search on the page ~~to~~ number, then comb through a smaller page.

(h) Yes. Concurrent execution is possible with preemption. Using multiple threads, a thread can call the pthread_yield function to give up its CPU time, allowing another thread to be scheduled. ~~This all~~ If all threads use pthread_yield to relinquish its CPU control, then they can concurrently execute and share CPU.

4

(e) In general, we would expect the CPU to have reduced idle time with faster I/O device, since less time is taken to transfer pages to memory. For example, Mario can expect transfer rates to increase from 1KB per second to 100MB per second, or a 100 000x speed up in I/O. With reduced idle, Mario can expect an increase in CPU utilization, or even processes to become CPU-bound given the high CPU demands of modern software.

4

CHALMERS	Anonymous code / Anonym kod	Points for question (to be filled in by teacher) / Poäng på uppgiften (ifylles av lärare)	Consecutive page no. / Löpande sid nr
	DIT401-0007-LLZ		04
			Question no. / Uppgift nr 01

(i) The initial password P_0, is only shared between the client and server, and the client has a set of ~~passwords~~ passwords, $P_0 \ldots P_n$. These passwords have a special relationship, such that $P_i = f(P_{i+1})$, where f is the one-way hash function. Using P_i, where $1 \leq i \leq n$, the server is able to determine P_0 by chaining the initial password : $P_{m-1} = f(P_m)$, $P_{m-2} = f(P_{m-1})$, ... until it reaches P_0 and has chained the function m times. If the value of P_0 matches what was initially shared between the client and server, then the client request is authenticated.

4

5

CHALMERS	Anonymous code / Anonym kod	Points for question (to be filled in by teacher) / Poäng på uppgiften (ifylles av lärare)	Consecutive page no. / Löpande sid nr
	PIT401-0007-LLZ		05
			Question no. / Uppgift nr 02

						↓					↓					↓					↓		
Timestep	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
most ↑	1	3	5	7	9	2	4	6	8	10	1	3	5	7	9	2	4	6	8	10	1	3	5
		1	3	5	7	9	2	4	6	8	10	1	3	5	7	9	2	4	6	8	10	1	3
cache			1	3	5	7	9	2	4	6	8	10	1	3	5	7	9	2	4	6	8	10	1
				1	3	5	7	9	2	4	6	8	10	1	3	5	7	9	2	4	6	8	10
least ↓					1	3	5	7	9	2	4	6	8	10	1	3	5	7	9	2	4	6	8
stolen						1					2					1					2		
page fault	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	X	✓	✓	✓	✓	X	✓	✓
reclaim											1					2					1		

12

Reference string: 1 3 5 7 9 2 4 6 8 10 1 3 5 7 9 2 4 6 8 10 1 3 5

(a) From the above reference string, we can see that all pages 1-10 are accessed twice, except for pages 1, 3, 5, which are accessed 3 times.

(b) Let P_i be the page accessed at time $t=i$

in the above reference string, $2 \leq |P_{i+1} - P_i| \leq 9$. There is no instance where $|P_{i+1} - P_i| = 1$.

(c) Assuming that the first page that is stolen refers to the oldest frame among the frames in the cache, at $t=5$, ~~a~~ page 1 is stolen, and taken back at $t=10$. Page 2 is stolen at $t=10$ and taken back at $t=15$. Page 1 is stolen again at $t=15$ and taken back at $t=20$. Page 2 is stolen at $t=20$.

I assume that when the process reclaims a stolen frame, it is not considered as a page fault since it is already in memory, and just needs to be re-assigned back to the process.

There are 20 page faults in this reference string and 23 page references.

CHALMERS | Anonymous code / Anonym kod: DIT401-0007-LLZ | Points for question (to be filled in by teacher) / Poäng på uppgiften (ifylles av lärare) | Consecutive page no. / Löpande sid nr: 06 | Question no. / Uppgift nr: 03

Assumption: ① compression ~~and deco~~ is optional, since the goal is to reduce internal fragmentation

② the compression algorithm ~~does not have to c~~ can vary its compression rate to suit the required output size

③ there is no compression on memory

④ A hashing function is available

⑤ Time taken to compress, decompress and hash a file is not a concern.

Data structure of a file: ① char array containing data

② bool flag: compressed

③ int checksum

④ typical attributes of a file.

the file system would have to intercept page faults and write instructions since pages in memory would already be decompressed. ??? why virtual memory (only) here?

① writing files

the file system can check if the file size is a multiple of the block size. If true, the file system can optionally compress the file, depending on requirements since there ~~a~~ would be no internal fragmentation if the file was stored as is (compressed = 0)

if the system were to compress the file, or the file size is not a multiple of the block size, then compression is necessary. The file system can determine a suitable compressed size, and compress the raw data up to a point where file size % block size == 0 or in other words where file size is a multiple of block size. always possible?

Then the filesystem should also store the hash of the original file ~~the~~ as checksum when decompressing the file.

Answer only one question on this page. Do not write on the back of this paper
Behandla endast en uppgift på detta blad. Skriv ej på baksidan

5

CHALMERS	Anonymous code	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr 07
	Anonym kod DAT401-0007-LLZ	Poäng på uppgiften (ifylles av lärare)	Question no. Uppgift nr 03

② Reading files.

If the ~~ER~~ requested file is already in-memory, it is uncompressed and there is no need to intercept these requests

The file system can listen for page faults to intercept, as they require extracting possibly compressed files in the disk.

When a file is retrieved, the system can first check if the compressed bit is set. If it is not set, the transfer can begin as per normal.

However, if decompression is required, the ~~CPU~~ file system has to transfer the file to memory, create another task for the CPU to perform the decompression step,

After the decompression is done, the decompressed data should also be hashed, and then checked against the checksum to verify successful decompression.

After which the original requesting instruction can be restarted, and subsequent requests for the same file would be cached in memory.

data structures?

10