Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2017-03-17
Sal (1)	TER4(70)
Tid	14-18
Kurskod	TDIU11
Provkod	TEN1
Kursnamn/benämning **Provnamn/benämning**	Operativsystem Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	5
Jour/Kursansvarig Ange vem som besöker salen	Ahmed Rezine
Telefon under skrivtiden	013 - 28 19 38
Besöker salen ca klockan	Under första tentatimmen
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Anna Grabska Eklund 013-28 23 62 anna.grabska.eklund@liu.se
Tillåtna hjälpmedel	Ordbok, engelska till valfritt språk.
Övrigt	
Antal exemplar i påsen	

TDIU11 Exam

Ahmed Rezine

2017-03-17 14-18
TER4

Admitted material

Dictionary from English to your chosen language.

Jour

Ahmed Rezine (013-281938) visiting after about one hour.

Instructions

- Fill in the exam wrapper and read the instructions on it before you start.
  Read instructions and all assignments carefully and completely before you begin.
- You may answer in either English or Swedish.
- State your interpretation of the question and all assumptions you make.
- Write clearly. **Unreadable text will be ignored.**
- Be precise in your statements. **Prove your point when possible.**
  Ambiguous formulations will lead to reduction of points.
- **Motivate clearly and in depth all statements and reasoning.**
  Explain calculations and solution procedures.
- The exam is 40 points and graded U, 3, 4, 5 (**preliminary** limits: 20p, 30p, 35p).
  Points are given for motivations, explanations, and reasoning.

Definitions

Unless you are more specific, the correcting team will interpret the following terms as follow:

memory Volatile random access memory (DRAM), about 100ns access time.

disk Permanent storage, about 10ms access time.

page A fix size region of virtual memory, possibly on disk.

frame A fix size region of physical memory (DRAM).

block A data block located on disk.

Problem 1 (12p)

Assume a paged virtual memory with 16 bits virtual addresses. Each process can address at most 128 pages. Assume single level paging.

1. What is the size in bytes of each single page? (2pt)
2. How large is a single page table if each page entry is a 4 bytes word? (2pt)

Assume in the following physical addresses are 32 bits (i.e., 4 bytes) long. Virtual addresses are still 16 bits long.

1. How many bits are used in the physical address to identify a frame? (2pt)
2. What is the maximal memory size that can be supported by such a system? (2pt)
3. Is it possible for this system with 16 bits virtual addresses to simultaneously use more than one GiB of memory? (2pt)
4. Suppose the first entries of the page table of some process associate pages to frames according to table (1). What is the physical address (in binary) corresponding to the virtual address $(0000\ 0100\ 1111\ 1001)_{\texttt{binary}}$? (2pt)

frame bits	valid bit	other bits	
$(000\ 0000\ 0000\ 0000\ 0000\ 1000)_{\texttt{binary}}$	$(0)_{\texttt{binary}}$	$(\texttt{XXXX XXXX})_{\texttt{binary}}$	1^{st} page table entry
$(000\ 0000\ 0000\ 0000\ 0000\ 1001)_{\texttt{binary}}$	$(1)_{\texttt{binary}}$	$(\texttt{XXXX XXXX})_{\texttt{binary}}$	2^{nd} page table entry
$(000\ 0000\ 0000\ 0001\ 0001\ 0000)_{\texttt{binary}}$	$(1)_{\texttt{binary}}$	$(\texttt{XXXX XXXX})_{\texttt{binary}}$	3^{rd} page table entry
$(000\ 0000\ 0000\ 0000\ 0001\ 0011)_{\texttt{binary}}$	$(1)_{\texttt{binary}}$	$(\texttt{XXXX XXXX})_{\texttt{binary}}$	4^{th} page table entry
$(000\ 0000\ 0000\ 0010\ 0010\ 0010)_{\texttt{binary}}$	$(0)_{\texttt{binary}}$	$(\texttt{XXXX XXXX})_{\texttt{binary}}$	5^{th} page table entry
...	...	...	
$(\texttt{XXX XXXX XXXX XXXX XXXX XXXX})_{\texttt{binary}}$	$(\texttt{X})_{\texttt{binary}}$	$(\texttt{XXXX XXXX})_{\texttt{binary}}$	n^{th} page table entry

Table 1: Some entries of a process page table. The symbol X is used to mean a bit value that is not relevant for the question.

Problem 2 (4p)

Some Linux supported file systems can make use of ACL:s (Access Control Lists).

1. Explain what ACLs are and give a disadvantage of using them. (2pt)
2. Give a simple example where using ACLs is useful even in the presence of the traditional "user:group:all" Unix/Linux access control mechanism. (2pt)

Problem 3 (12p)

Assume an idle system. Consider the workload depicted in table (2).
First assume an FCFS scheduler.

1. Draw a Gantt diagram for processes' execution and queues' contents. (2pt)
2. What are the individual and the average waiting times? (2pt)

Job	Arrival time	Execution time
J_1	0	7
J_2	1	5
J_3	2	3
J_4	3	1

Table 2: Workload for problem 3.

Now assume a preemptive SJF scheduler.

1. Draw a Gantt diagram for processes' execution and queues' contents. (2pt)
2. What are the individual and the average waiting times? (2pt)

General questions about the SJF scheduler algortihm:

1. What is aging in this context and why is it sometimes combined with SJF? (2pt).
2. Assume that the problem aging solves is not an issue. In addition, assume you are willing to pay the price of the increase in the number of context switches that come with SJF. What do you see then as the biggest problem when adopting the SJF scheduling algorithm? how to remedy this? (2pt).

Problem 4 (8 p)

Assume 28 bits (logical blocks') pointers and a 1TiB (i.e., 2^{40} bytes) hard drive.

1. Give the minimal size of a logical block. (2pt)
2. Assume 4KiB logical blocks. What is the size of the FAT table with 4 bytes per entry and corresponding to having a single 1TiB volume? recall that FAT is a form of linked allocation but where all links are centralized in a single table for the whole volume. (2pt)
3. Still assuming 4 bytes per FAT entry and a 1TiB volume, how large (in bytes) should a single logical block be in order to obtain a FAT table of 128 MiB (i.e., 2^{27} bytes) ? (2pt)
4. Compare advantages and disadvantages of the two logical block sizes (i.e, the block sizes adopted/obtained in questions 2 and 3). (2pt)
5. Give two advantages of FAT based allocation over indexed allocation. (2pt)

Problem 5 (4p)

1. Describe a "dictionary password attack". (2pt)
2. In this context, what is "salt" and how can it help make such an attack harder? (2pt)