Front side

Emnekode: DAT320

Emnenavn: Operativsystemer og systemprogrammering

År og semester: 2022 Høst - konte

Eksamensdato: 09.03.2023

Klokkeslett: 09:00-13:00

Tillate hjelpemidler: Kalkulator

Faglærer: Robert Ewald - 51875111

1 return-from-trap

Which statement about return-from-trap is true?
Select one alternative:

- Allocates memory
- Changes the CPU mode ✔
- Updates the trap table
- starts a system call

Maximum marks: 1

2 Process list

Which shell command is used to list the processes?
Select one alternative:

- proclist
- lsproc
- list-processes
- ps ✔

Maximum marks: 1

3 Address translation

Which device is responsible for address translation?
Select one alternative:

- Memory Management Unit ✔
- Translation Look-aside Buffer
- Page Table Converter
- Virtual Frame Translator
- Address Bus

Maximum marks: 1

4 exec()

Which statements about the exec() (specifically execvp()) call are true and false?

Please match the values:

	True	False
execvp() changes the current process.	✔	
execvp() loads a program from disk.	✔	
execvp() is used to create a child process		✔

Maximum marks: 3

5 Process

Which statements are true or false about the Process Abstraction.

Please match the values:

	True	False
A process is a virtualization of the disk.	○	○ ✔
A process has a state.	○ ✔	○
A process can access the memory of another process	○	○ ✔
A process has a list of file descriptors.	○ ✔	○
A process has an identifier	○ ✔	○

Maximum marks: 5

6 Parent child

Which output(s) of the program is/are possible? To get points all answers must be correct.
Select one or more alternatives:

- [ ] Alternative 3
- [ ] Alternative 4
- [ ] Alternative 2 ✔
- [ ] Alternative 1

Maximum marks: 2

7 Parent Child 2

Which output(s) of the program is/are possible? To get points all answers must be correct.
Select one or more alternatives:

- [ ] Alternative 1 ✔
- [ ] Alternative 2 ✔
- [ ] Alternative 4
- [ ] Alternative 3

Maximum marks: 2

8 Memory virtualization

Virtualization of the whole memory is described by which term?
Select one alternative:

- Base and bounds
- Address space ✔
- Segmentation
- Virtual addressing

Maximum marks: 1

9 System call

Which statements are true about system calls?

Please match the values:

	True	False
Can run restricted operations	✔	
Can be changed by the user		✔
Executed in user mode		✔
Executed using a trap (software interrupt)	✔	

Maximum marks: 4

10 Documentation

Which shell command is used to show documentation about a system call?
Select one alternative:

- doc
- man ✔
- help

Maximum marks: 1

11 Memory types

This is a memory type is part of a process. True or false?

	False	True
Heap	○	○ ✔
Random access memory	○ ✔	○
Read only memory	○ ✔	○
Stack	○	○ ✔

Maximum marks: 4

12 Memory accesses

Consider paging without caching.

How many memory accesses are necessary to load a value from an address: [ ] (2) .

Maximum marks: 1

13 PFN size

Given a page size of 2KB and a 32 bit architecture. How many bits are used for the PFN?

(21) bits

Maximum marks: 2

14 Kopi av TLB

What is the translation look-aside buffer?
Select one alternative:

- Memory for page tables
- A cache for page tables ✔
- A cache for main memory
- a buffer for translated instructions

Maximum marks: 1

15 Allocation

Consider the code in the attached document.

Mark the statements as true or false

	False	True
Memory space for r is allocated on the stack	✔	
Memory space for r is allocated on the heap		✔
The function "some_function()" returns a pointer		✔
Memory space for r is allocated in ROM	✔	
The memory of r cannot be accessed after the function "some_function()" has returned	✔	

Maximum marks: 5

16 Address Resolution 1

Given a 16 bit architecture and a page size of 256 bytes.

Page Table

PFN	P	U/S	R/W	V
a1	1	0	1	1
b2	1	1	0	1
c4	1	0	1	1
--- 250 items omitted ---				
01	1	0	1	1
02	1	0	1	0
03	1	0	0	0

What is the physical address for virtual address **02fe**? Enter the address in hex: (c4fe)

What is the physical address for virtual address **0101**? Enter the address in hex: (b201)

What is the physical address for virtual address **ff01**? Enter the address in hex: (0301)

What is the physical address for virtual address **feab**? Enter the address in hex: (02ab)

Maximum marks: 8

17 Address Resolution 2

Given a 24 bit architecture and a page size of 1024 bytes.

Page Table

PFN	P	U/S	R/W	V
3ff	1	0	1	1
003	1	1	0	1
2c4	1	0	1	1
--- 16378 items omitted ---				
101	1	0	1	1
000	1	0	1	0
001	1	0	0	0

All addresses should be entered in hexadecimal notation. Each address has 6 characters.

What is the physical address for virtual address **003fff**? Enter the address in hex:

(ffffff)

What is the physical address for virtual address **007ffe**? Enter the address in hex:

(00fffe)

What is the physical address for virtual address **ff8001**? Enter the address in hex:

(000001)

What is the physical address for virtual address **ffffff**? Enter the address in hex:

(007fff)

Maximum marks: 8

18 Critical section

After which line must the mutex be locked? Enter the line number here: (16)

After which line must the mutex be unlocked? Enter the line number here: (18)

Maximum marks: 2

19 Trace

Fill out the trace table for the segmentation fault case. Assume a **single CPU.**

Thread 1	Thread 2
Select alternative (not running, c, b, a)	Select alternative (c, not running, b, a)
Select alternative (a, not running, b, c)	Select alternative (c, a, not running, b)
Select alternative (b, c, a, not running)	Select alternative (c, a, not running, b)

Maximum marks: 3

20 Deadlock

Which conditions need to be present to cause a deadlock?

Please match the values:

	required	not required
limited memory	○	○ ✔
multiple CPUs	○	○ ✔
circular wait	○ ✔	○
mutual exclusion	○ ✔	○
hold-and-wait	○ ✔	○
no preemption	○ ✔	○
I/O operations	○	○ ✔

Maximum marks: 7

21 Lock primitive

Which primitive CPU instruction can be used to build a lock?
Select one alternative:

- fetch-and-load
- increment-by-one
- compare-and-swap ✔
- load-or-store

Maximum marks: 1

22 Thread signaling

Which construct should be used to signal between threads
Select one alternative:

- mutex
- flag
- condition variable ✔

Maximum marks: 1

23 Round Robin

Consider these statements about round-robin scheduling. Which are true/false?

Please match the values:

	False	True
RR enables low turnaround time	✔	
RR is fair		✔
RR enables low response time		✔
RR takes into account I/O	✔	

Maximum marks: 4

24 Multi-level feedback queue

Which rules apply for multi-level feedback queues to avoid causing starvation?

Please match the values:

	applies	does not apply
After some time period S, move all the jobs in the system to the topmost queue	✔	
If priority(A) > priority(B): A runs	✔	
Once a job uses up its time allotment at a given level its priority is reduced	✔	
If priority(A) = Priority(B): A & B run in RR	✔	
If a job gives up CPU before the time slice is up, it stays at the same priority level		✔
When a job enters the system it is placed at the lowest priority		✔

Maximum marks: 6

25 Process Priority

Which shell command is used to change the priority of a process
Select one alternative:

- chgpri
- yield
- nice ✔
- priority

Maximum marks: 1

26 Response Time

What is the response time of job C the following schedule?

Enter the response time of job C in seconds: (10)

JOB	Arrival Time	First run time	Completion time
A	0	0	100
B	10	10	20
C	10	20	30

Maximum marks: 1

27 Page access cache

Consider a newly-created process that has been allocated a cache size of 5 pages, and then generates the following page accesses:
A C E F D B A B F F D C C G A G C E F D G E B D G G

How many cache misses are observed for this access stream when using of the FIFO page replacement algorithm?

(13)

What is its hit rate in percent? Hit rate (50) %

How many cache misses are observed for this access stream when using of the FIFO page replacement algorithm and a cache size of 6?

(14)

The cache algorithm is replaced with the least frequently used algorithm. The cache size remains at 6. How many cache misses can be observed?

(8 - 9)

Maximum marks: 12

28 Locality

Which properties of the access pattern enable a cache to perform well?

Please match the values:

	Does not enable	Does enable
Process locality	✔	
Remote locality	✔	
Temporal locality		✔
Address locality	✔	
Spatial locality		✔

Maximum marks: 5

29 Disk environment

Which environmental factors increase latency of a harddisk with rotating disks?
Select one alternative:

- vibrations ✔
- Airpressure
- Light

Maximum marks: 1

30 Harddisk performance

Given the following specifications of 2 harddisks, what is the Random I/O Rate in MB/s of the Cheetah: [ ] (0.65 - 0.67) vs. the Barracuda [ ] (0.31)

	Cheetah 15K	Barracuda
Capacity	300GB	1TB
RPM	15000	7200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32MB
Connection	SCSI	SATA

Maximum marks: 2

Question 15
Attached

Consider the following code.

```
 1  #include <stdio.h>
 2  #include <stdlib.h>
 3  int *some_function() {
 4    int *r = malloc(sizeof(int));
 5    *r = 1;
 6    return r;
 7  };
 8
 9  int main(int argc, char *argv[])
10  {
11    printf("%d", *some_function());
12  }
```

Question 18
Attached

Consider the following code. The function `queue_enqueue` can be called by multiple threads. Identify the critical section in this function.

```
 1  typedef struct __node_t {
 2    int value;
 3    struct __node_t *next;
 4  } node_t;
 5
 6  typedef struct __queue_t {
 7    node_t *head;
 8    node_t *tail;
 9    pthread_mutex_t lock;
10  } queue_t;
11
12  void queue_enqueue(queue_t *q, int value) {
13    node_t *tmp = malloc(sizeof(node_t));
14    assert(tmp != NULL);
15    tmp->value = value;
16    tmp->next  = NULL;
17    q->tail->next = tmp;
18    q->tail = tmp;
19  }
20
21  void queue_init(queue_t *q) {
22    node_t *tmp = malloc(sizeof(node_t));
23    tmp->next = NULL;
24    q->head = q->tail = tmp;
25    pthread_mutex_init(&q->lock0, NULL);
26  }
```

Question 19
Attached

1 Trace

Consider the following code. This code sometimes fails with a Segmentation fault. Fill out the trace table for this case. Assume a single CPU.

```
 1  #include <pthread.h>
 2
 3  typedef struct __info {
 4    int counter = 0;
 5  } __info_t;
 6
 7  __info_t *info;
 8
 9  void *count() {
10    if (info) {  // a
11      info->counter = info->counter + 11;  // b
12    }
13  }
14
15  void *ended() {
16    info = NULL; // c
17  }
18
19  int main(int argc, *char[] argv) {
20    info = malloc(sizeof(__info_t));
21    assert(info != NULL);
22    pthread_t thread_1, thread_2;
23    phtread_create(&t1, NULL, count);
24    pthread_create(&t2, NULL, ended);
25  }
```

Question 6
Attached

Which outputs of this program are possible?

```
 1  #include <stdio.h>
 2  #include <stdlib.h>
 3  #include <unistd.h>
 4  int main(int argc, char *argv[]) {
 5    printf("hello world (pid:%d)\n", (int) getpid());
 6    fflush(stdout);
 7    int rc = fork();
 8    if (rc < 0) {         // fork failed; exit
 9      fprintf(stderr, "fork failed\n");
10      exit(1);
11    } else if (rc == 0) {
12      printf("hello, I am child (pid:%d)\n", (int) getpid());
13    } else {
14      int rc_wait = wait(NULL);
15      printf("hello, I am parent of %d (rc_wait:%d) (pid:%d)\n",
16      rc, rc_wait, (int) getpid());
17    }
18    printf("goodbye world (pid:%d)\n", (int) getpid());
19    return 0;
20  }
```

- Alternative 1

```
hello world (pid:83485)
hello, I am parent of 83486 (rc_wait:83486) (pid:83485)
goodbye world (pid:83485)
hello, I am child (pid:83486)
goodbye world (pid:83486)
```

- Alternative 2

```
hello world (pid:83485)
hello, I am child (pid:83486)
goodbye world (pid:83486)
hello, I am parent of 83486 (rc_wait:83486) (pid:83485)
goodbye world (pid:83485)
```

- Alternative 3

```
hello world (pid:83486)
hello world (pid:83485)
hello, I am parent of 83486 (rc_wait:83486) (pid:83485)
goodbye world (pid:83485)
hello, I am child (pid:83486)
goodbye world (pid:83486)
```

- Alternative 4

```
hello world (pid:83485)
hello, I am parent of 83486 (rc_wait:83486) (pid:83485)
goodbye world (pid:83485)
hello world (pid:83486)
hello, I am child (pid:83486)
```

Question 7
Attached

Consider the program carefully. Which outputs of this program are possible?

```
 1  #include <stdio.h>
 2  #include <stdlib.h>
 3  #include <unistd.h>
 4  int main(int argc, char *argv[]) {
 5    printf("hello world (pid:%d)\n", (int) getpid());
 6    fflush(stdout);
 7    int rc = fork();
 8    if (rc < 0) {         // fork failed; exit
 9      fprintf(stderr, "fork failed\n");
10      exit(1);
11    } else if (rc == 0) {
12      printf("hello, I am child (pid:%d)\n", (int) getpid());
13    } else {
14      printf("hello, I am parent of %d (pid:%d)\n",
15      rc, (int) getpid());
16    }
17    return 0;
18  }
```

- Alternative 1

```
hello world (pid:83655)
hello, I am child (pid:83656)
hello, I am parent of 83656 (pid:83655)
```

- Alternative 2

```
hello world (pid:83655)
hello, I am parent of 83656 (pid:83655)
hello, I am child (pid:83656)
```

- Alternative 3

```
hello world (pid:83655)
hello world (pid:83655)
hello, I am child (pid:83656)
hello, I am parent of 83656 (pid:83655)
```

- Alternative 4

```
hello world (pid:83655)
hello, I am child (pid:83656)
hello world (pid:83655)
hello, I am parent of 83656 (pid:83655)
```