Name, SURNAME and ID ⇒ KEY

Middle East Technical University
Department of Computer Engineering

CENG 709

Fall '2018-2019

Midterm

- **Duration: 120** minutes.
- **Exam:**
  - This is a **closed book**, **closed notes** exam.
  - No attempts of cheating will be tolerated. In case such attempts are observed, the students who took part in the act will be prosecuted. The legal code states that students who are found guilty of cheating shall be expelled from the university for **a minimum of one semester**!
  - **Data sheet for some aspects of x86-64 assembly is available on the last page.**
  - **This booklet consists of *8* pages including this page. Check that you have them all!**

Question 1	
Question 2	
Question 3	
Question 4	
Total ⇒	

Name, SURNAME and ID ⇒

1 **(20 pts)**

For this problem, assume the following:

- `int`, `unsigned` representations use 8 bits.
- `short integer` and `short unsigned` representations use 4 bits.
- Right shifts `int`'s are arithmetic.
- `float` representation use 8 bits which consists of 1 bit for `sign`, 3 bits as `exp` and 4 bits as `frac` .
- `double` representation use 8 bits which consists of 1 bit for `sign`, 4 bits as `exp` and 3 bits as `frac` .
- Fill in the following table.

Type	Expression	Binary representation	Decimal representation
int	TMin	1000 0000	-2^7
unsigned	UMax	1111 1111	$2^8 - 1$
short int	(short) TMin	0000	0
int	TMax >> 2	0001 1111	$2^5 - 1$
int	TMin+TMin	0000 0000	0

- Fill in the following table.

Description	Binary representation	Decimal representation
$-\infty$ in float	1 111 0000	-
Minus Zero represented double	1 0000 000	-0
-25/8 represented by float	1 100 1001	-25/8
Largest positive number that can be represented by float	0 110 1111	$(2-2^{-4})\cdot 2^{6-3} = \frac{31}{2}$
Largest positive number that can be represented by double	0 1110 111	$(2-2^{-3})\cdot 2^{14-7} = 240$
Smallest positive number that can be represented by float	0 000 0001	$2^{-4}\cdot 2^{1-3} = 2^{-6} = \frac{1}{64}$

Float Bias: $FBias = 2^2 - 1 = 3$

Double Bias: $DBias = 2^3 - 1 = 7$

$\frac{25}{8} = 3 \cdot \frac{1}{8} = 11,001_2 = 1.\underbrace{1001}_{frac}{}_2 \cdot 2^{1}$

$1 = exp - Bias \Rightarrow exp = 4 = 100_2$

Name, SURNAME and ID ⇒

2 **(30 pts)**

Consider the following x86-64 assembly code:

```
long loop(long x, int n)
x in %rdi, n in %esi
loop:
 movl %esi, %ecx
 movl $1, %edx
 movl $0, %eax
 jmp .L2
.L3:
 movq %rdi, %r8
 andq %rdx, %r8
 orq %r8, %rax
 salq %cl, %rdx
.L2:
testq %rdx, %rdx
jne .L3
rep; ret
```

The preceding code was generated by compiling C code that had the following overall form:

```
long loop(long x, long n)
{

 long result = ______0_______;

 long mask;


 for (mask = _0x1_____; mask __!=0_______; mask = _mask<<n____) {

    result |= __x & mask________;

 }

 return result;
}
```

Fill in the blanks in the code skeleton printed above.

Name, SURNAME and ID ⇒

3 **(10 pts)**

Answer the following questions:

- (2 pts) What is the smallest number that can be represented when using a 13 bit two's complement representation?

  $1\ 0000\ 0000\ 0000_2 = -2^{12} = -4096$

- (2 pts) What is the value of the following C statement (x is a char):

```
(x & 0xFE) == ((x>>1)<<1);
```

  F E
  1111 1110
  x & 0xFE = $x_7 \cdots x_2 x_1 0$ == $x_7 \cdots x_1 0$
  TRUE

- (1 pts) How long is an address (how many bits) in a memory system in which each address corresponds to a single byte, and which can address 1M byte of memory?

  1Mbyte = 2^{20} ⇒ Address = 20 bits

- (1 pts) What is the name of a dedicated cache used to speed up address translation in virtual memory systems?

  TLB (Translation Lookaside Buffer)

- (2 pts) Name and describe (briefly) the two principles of locality.

  Temporal locality ~~~

  Spatial locality ~~~

- (2 pts) Briefly describe how block size effects cache performance (miss/hit rate, hit time/miss penalty).

  Block size ↑ ⇒ Hit rate ↑ Miss rate ↓
  Hit time = (no change)
  Miss Penalty = (no change)

Name, SURNAME and ID ⇒

4 **(40 pts)**

The following problem concerns the way virtual addresses are translated into physical addresses.

- The memory is byte addressable.
- Memory accesses are to **1-byte words** (not 4-byte words).
- Virtual addresses are 16 bits wide.
- Physical addresses are 13 bits wide.
- The page size is 512 bytes.
- The TLB is 8-way set associative with 16 total entries.
- The cache is 2-way set associative, with a 4 byte line size and 16 total lines.

In the following tables, **all numbers are given in hexadecimal**. The contents of the TLB, the page table for the first 32 pages, and the cache are as follows:

TLB			
Index	Tag	PPN	Valid
0	09	4	1
	12	2	1
	10	0	1
	08	5	1
	05	7	1
	13	1	0
	10	3	0
	18	3	0
1	04	1	0
	0C	1	0
	12	0	0
	08	1	0
	06	7	0
	03	1	0
	07	5	0
	02	2	0

Page Table					
VPN	PPN	Valid	VPN	PPN	Valid
00	6	1	10	0	1
01	5	0	11	5	0
02	3	1	12	2	1
03	4	1	13	4	0
04	2	0	14	6	0
05	7	1	15	2	0
06	1	0	16	4	0
07	3	0	17	6	0
08	5	1	18	1	1
09	4	0	19	2	0
0A	3	0	1A	5	0
0B	2	0	1B	7	0
0C	5	0	1C	6	0
0D	6	0	1D	2	0
0E	1	1	1E	3	0
0F	0	0	1F	1	0

2-way Set Associative Cache												
Index	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3	Tag	Valid	Byte 0	Byte 1	Byte 2	Byte 3
0	19	1	99	11	23	11	00	0	99	11	23	11
1	15	0	4F	22	EC	11	2F	1	55	59	0B	41
2	1B	1	00	02	04	08	0B	1	01	03	05	07
3	06	0	84	06	B2	9C	12	0	84	06	B2	9C
4	07	0	43	6D	8F	09	05	0	43	6D	8F	09
5	0D	1	36	32	00	78	1E	1	A1	B2	C4	DE
6	11	0	A2	37	68	31	00	1	BB	77	33	00
7	16	1	11	C2	11	33	1E	1	00	C0	0F	00

Name, SURNAME and ID ⇒

- (5 pts) The box below shows the format of a virtual address. Indicate (by labeling the diagram) the fields (if they exist) that would be used to determine the following: (If a field doesn't exist, don't draw it on the diagram.)

VPO	The virtual page offset
VPN	The virtual page number
TLBI	The TLB index
TLBT	The TLB tag

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- (5 pts) The box below shows the format of a physical address. Indicate (by labeling the diagram) the fields that would be used to determine the following:

PPO	The physical page offset
PPN	The physical page number
CO	The block offset within the cache line
CI	The cache index
CT	The cache tag

12	11	10	9	8	7	6	5	4	3	2	1	0

For the given virtual address, indicate the TLB entry accessed, the physical address, and the cache byte value returned **in hex**. Indicate whether the TLB misses, whether a page fault occurs, and whether a cache miss occurs.

If there is a cache miss, enter "-" for "Cache Byte returned". If there is a page fault, enter "-" for "PPN" and leave parts C and D blank.

Virtual address: `1DDE`

6

- ~~(5~~ pts) Virtual address format (one bit per box)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	1	1	1	0	1	1	1	1	0

8

- ~~(5~~ pts) Address translation

Parameter	Value
VPN	0x0E
TLB Index	0x0
TLB Tag	0x07
TLB Hit? (Y/N)	N
Page Fault? (Y/N)	N
PPN	0x1

6

- ~~(5~~ pts) Physical address format (one bit per box)

12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	1	1	0	1	1	1	1	0

8

- ~~(5~~ pts) Physical memory reference

Parameter	Value
Byte offset	0x2
Cache Index	0x7
Cache Tag	0x1E
Cache Hit? (Y/N)	Y
Cache Byte returned	0x0F

Data sheet for x86-64 Assembly

- Arithmetic operations

```
addq  Src,Dest Dest = Dest + Src
subq  Src,Dest Dest = Dest - Src
imulq Src,Dest Dest = Dest * Src
salq  Src,Dest Dest = Dest << Src Also called shll
sarq  Src,Dest Dest = Dest >> Src Arithmetic
shrq  Src,Dest Dest = Dest >> Src Logical
xorq  Src,Dest Dest = Dest ^ Src
andq  Src,Dest Dest = Dest & Src
orq   Src,Dest Dest = Dest | Src

incq Dest Dest = Dest + 1
decq Dest Dest = Dest - 1
negq Dest Dest = - Dest
notq Dest Dest = ~ Dest
```

- `cmpq Src2,Src1`
  - `cmpq b,a` like computing `a-b` without setting destination
- `testq Src2,Src1`
  - `testq b,a` like computing `a&b` without setting destination
- Condition codes:
  - `CF` set if carry out from most significant bit
  - `ZF` set if `t == 0`
  - `SF` set if `t < 0`
  - `OF` set if two's complement overflow
- Jump operations

jmp	1	Unconditional
je	ZF	Equal / Zero
jne	~ZF	Not Equal / Not Zero
js	SF	Negative
jns	~SF	Nonnegative
jg	~(SF ^ OF)&~ZF	Greater (Signed)
jge	~(SF ^ OF)	Greater or Equal (Signed)
jl	(SF ^ OF)	Less (Signed)
jle	(SF ^ OF) | ZF	Less or Equal (Signed)
ja	~CF&~ZF	Above (unsigned)
jb	CF	Below (unsigned)