ICS 53, Spring 2022
Discussion 4
4/25/2022

Assignment 3

Components

The virtual memory system will have 128 addresses (0 - 127)

The main memory has 32 addresses (0 - 31)

Each page contains 8 addresses

Components

The virtual memory system will have 128 addresses (0 - 127) → virtual memory system will have 16 pages

The main memory has 32 addresses (0 - 31) → main memory will have 4 pages

Each page contains 8 addresses

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	0	0	1
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0	
1	
2	
3	

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	0	0	1
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0	
1	
2	
3	

> read 9

A Page Fault Has Occurred

-1

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	1	0	0
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	
1	
2	
3	

> read 9

A Page Fault Has Occurred

-1

> write 9 201

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 201 2 10: 3 11: 4 12: 5 13: 6 14: 7 15:
1	
2	
3	

> read 9

A Page Fault Has Occurred

-1

> write 9 201

> read 9

201

> showmain 0

0: -1
1: 201
2: -1
3: -1
5: -1
6: -1
7: -1

> showptable

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 201 2 10: 3 11: 4 12: 5 13: 6 14: 7 15:
1	
2	
3	

> read 9

A Page Fault Has Occurred

-1

> write 9 201

> read 9

201

> showmain 0

0: -1
1: 201
2: -1
3: -1
5: -1
6: -1
7: -1

> showptable

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	0	0	0
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0	0 1 2 3 4 5 6 7
1	
2	
3	

> write 10 202

A Page Fault Has Occurred

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	0	0	0
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	0	0	9
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 2 10: 202 3 11: 4 12: 5 13: 6 14: 7 15:
1	
2	
3	

> write 10 202

A Page Fault Has Occurred

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	1	0	3
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	1	1	1
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	1	0	2
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 2 10: 202 3 11: 4 12: 5 13: 6 14: 7 15:
1 3	… 15 31: 403
2 9	
3 0	

> write 10 202

A Page Fault Has Occurred

> write 31 403

A Page Fault Has Occurred

> read 72

A Page Fault Has Occurred

-1

> read 0

A Page Fault Has Occurred

-1

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	1	0	3
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	1	1	1
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	1	0	2
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 300 2 10: 202 3 11: 4 12: 5 13: 6 14: 7 15:
1 3	… 15 31: 403
2 9	
3 0	

> read 12

-1

> write 9 300

> showmain 1

8:-1

9:-1

10:-1

11:-1

12:-1

13:-1

14:-1

15:403
> showptable

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	1	0	3
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	1	1	1
32-39	4	0	0	4
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	1	0	2
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 300 2 10: 202 3 11: 4 12: 5 13: 6 14: 7 15:
1 3	… 15 31: 403
2 9	
3 0	

> write 32 40

A Page Fault Has Occurred

-1

PAGE TABLE

	VPN	V	D	PN(MM if V==1, disk if V==0)
0-7	0	1	0	3
8-15	1	1	1	0
16-23	2	0	0	2
24-31	3	0	0	3
32-39	4	1	1	1
40-47	5	0	0	5
48-55	6	0	0	6
56-63	7	0	0	7
64-71	8	0	0	8
72-79	9	1	0	2
80-87	10	0	0	10
88-95	11	0	0	11
96-103	12	0	0	12
104-111	13	0	0	13
112-119	14	0	0	14
119-127	15	0	0	15

Main Memory

0 1	0 8: 1 9: 300 2 10: 202 3 11: 4 12: 5 13: 6 14: 7 15:
1 4	8 32: 40
2 9	
3 0	

Address access order:
10(1), 31(3), 72(9), 0(0), 12(1), 9(1)
LRU: 1,3,9,0,1,1

> write 32 40

A Page Fault Has Occurred

-1

> showmain 1

8:40

9:-1

10:-1

11:-1

12:-1

13:-1

14:-1

15:-1

> showptable

Data Structure

```
struct Memory {
    int address, data;
};
struct PageTable {
    int v_page_num, valid_bit, dirty_bit, page_num, time_stamp;
};
struct Memory main_memory[32];
struct Memory virtual_memory[128];
struct PageTable p_table[8];
```

Initialization

```
int i;
    for (i = 0; i < sizeof(main_memory)/sizeof(main_memory[0]); i++) {
        main_memory[i].data = -1;
        main_memory[i].address = i;
    }
    for (i = 0; i < sizeof(virtual_memory)/sizeof(virtual_memory[0]); i++) {
        virtual_memory[i].data = -1;
        virtual_memory[i].address = i;
    }
    for (i = 0; i < sizeof(p_table)/sizeof(p_table[0]); i++) {
        p_table[i].v_page_num = p_table[i].page_num = i;
        p_table[i].valid_bit = p_table[i].dirty_bit = 0;
        p_table[i].time_stamp = 0;
    }
```

Main

```
int fifo = 0, lru = 0;

int main(int argc, char** argv) {
    if (argv[1] == NULL || strcmp (argv[1], "FIFO") == 0)
        fifo = 1;
    else if (strcmp (argv[1], "LRU") == 0)
        lru = 1;
    init();
    loop();
    return 0;
}
```

```
void loop() {
    char* input[100];
    char** args;
    do {
        printf("> ");
        fgets(input, 80, stdin);
        args = tokenize((char *) input);
        if (strcmp(args[0], "quit") == 0)
            exit(0);
        execute(args);
    } while (1);
}
```

Think about

Virtual to physical address conversion formula

Thank you!!

Questions?