CMPT 218 Midterm Fall 2023

Oct 27, 2023

- Time: 10:30 am — 12:15 pm
- Please write your answers on the space provided.
- This is a closed-book, no calculator, closed-notes exam.
- There are 20 questions.

Short Answer Questions

Please answer the following questions in 2-3 sentences *maximum*. You can only use the space provided. Do *not* just write what you think is relevant in hopes to get some partial credit. Verbose answers will result in point *reductions*.

1. **(2 pts)** How is containerization different from virtualization?

2. **(2 pts)** Explain how the Moore's Law changed its focus.

3. **(2 pts)** Explain what preemptive scheduling means.

4. **(2 pts)** List the memory/storage types discussed in class from the fastest to the slowest.

5. **(2 pts)** Explain the source of difference between 32-bit and 64-bit architectures.

6. **(2 pts)** Why do we call a kernel event-driven?

7. **(2 pts)** What does NUMA mean in a NUMA system?

8. **(2 pts)** How is a thread different from a process?

9. **(2 pts)** What is the biggest benefit of using an intermediate representation (IR) as in LLVM bitcode?

10. **(2 pts)** Draw the virtual memory space layout as discussed in class.

11. **(2 pts)** Suppose we manage 128 KB of virtual address space using paging, and the page size is 2 KB (2048 bytes). Answer the following questions. Note: 1 KB = 1024 bytes, 1 MB = 1024 KB, and 1GB = 1024 MB; 128 KB needs 17 bits to represent it.

    a) **(1 pts)** How many bits do we use to represent the virtual page number?

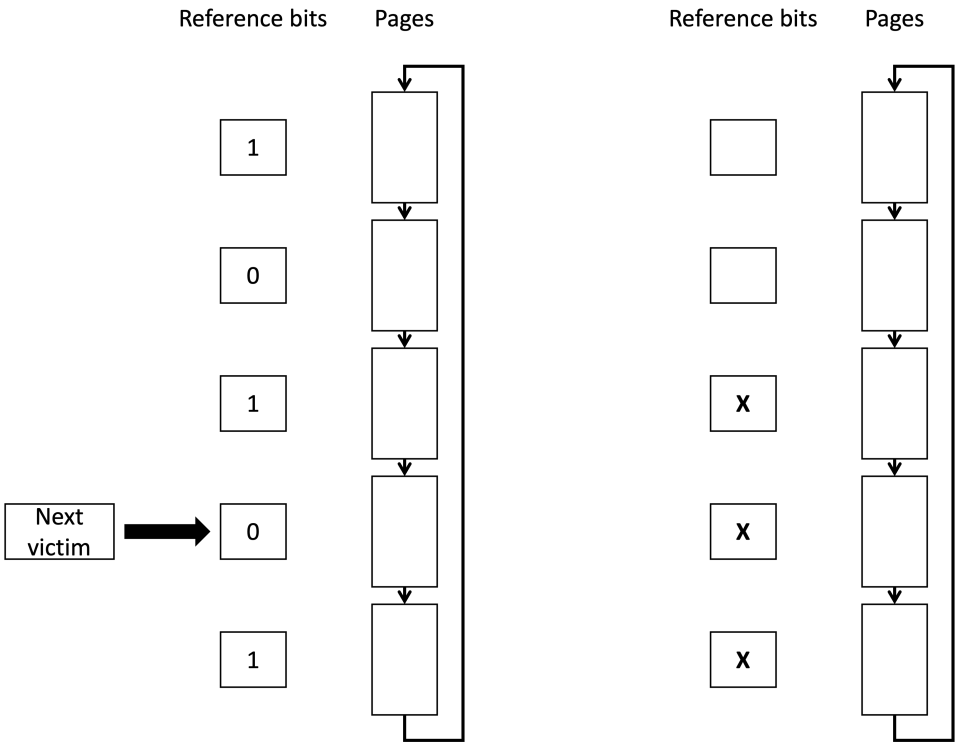
    b) **(1 pts)** How many entries do we have in the page table?

12. (2 pts) For each variable in the following code snippet, specify the segment of the address space where it is stored.

```
int global = 0;
int main(void) {
  float local;
  char *ptr;

  ptr = malloc(100);
  local = 0;
  local += 10*5;
  ...
  foo();
  ...
  return 0;
}
```

13. (2 pts) Consider the second-chance algorithm discussed in class. The diagram on the left shows the current state. Let's assume that we need to pick a page to swap out now. In the diagram on the right, fill in the first (top most) two boxes for reference bits according to what happens *after* picking a page to swap out using the second-chance algorithm. X simply means that we are hiding the value for the sake of this exam question.



14. (2 pts) Suppose you have four threads updating a global variable `sum` (initialized to 0) with the following function, expecting the final result of 4. Explain why that may not be what you get.

```
static void *thread_func(void *arg) {
  sum++;
  return 0;
}
```

15. (2 pts) Suppose that we use the Shortest Remaining Time First scheduling algorithm, and three processes P1, P2 and P3 have the following execution times and arrival times.

Process	Execution Time	Arrival Time
P1	8 ms	0 ms
P2	2 ms	2 ms
P3	5 ms	3 ms

a) **(1 pts)** Show the timeline of the execution of the processes.

b) **(1 pts)** Calculate the average waiting time.

Long Answer Questions

You can only use the space provided. All answers/code should be concise and to the point. Do *not* just write what you think is relevant in hopes to get some partial credit. Verbose answers will result in point *reductions*.

16. (4 pts) Write a program that creates a thread that receives the string `"pthread"` as an argument and prints it out. The following is the signature of `pthread_create()`.

```
#include <pthread.h>

int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
                   void *(*start_routine) (void *), void *arg);
```

17. (4 pts) Write a signal handler for the following program that simply prints out the signal number it is handling. Also, find missing statement(s) in the `main()` function, write the missing statement(s), and indicate where they should be located in the code using the alphabets provided as comments. The definition of `struct sigaction` is also provided.

```
#define _POSIX_C_SOURCE 1
#include <signal.h>
#include <stdio.h>

/*
struct sigaction {
  void     (*sa_handler)(int);
  void     (*sa_sigaction)(int, siginfo_t *, void *);
  sigset_t   sa_mask;
  int        sa_flags;
  void     (*sa_restorer)(void);
};
*/

int main(void) {
  // A
  struct sigaction handler;
  handler.sa_flags = 0;
  sigemptyset(&handler.sa_mask);
  // B
  sigaction(SIGINT, &handler, NULL);
  // C

  while (1)
    ;
  // D
  return 0;
}
```

18. (4 pts) Fill in the following code to create exactly one orphan process. Note that this can be done in a few (4-5 max) lines of code.

```
#include<sys/types.h>
#include<unistd.h>

int main() {
  pid_t pid = fork();
  // Fill in here
}
```

19. (4 pts; 1 pt extra if all correct) Consider the following code.

```
struct as_struct {
  uint8_t first;
  uint8_t second;
  uint16_t third;
};

union as_union {
  uint16_t first;
  uint8_t second;
  uint32_t third;
};

int main(void) {
  void *void_ptr = malloc(4);
  uint32_t *ptr32 = (uint32_t *)void_ptr;
  *ptr32 = 0xFACEB00C;
  struct as_struct *as = (struct as_struct *)void_ptr;
  union as_union *au = (union as_union *)void_ptr;
}
```

a) **(1 pt)** Visualize (i.e., draw) how the variable `i` is represented in memory. Clearly show the byte-by-byte values and mark the lowest address and the highest address.

b) **(1 pt)** Visualize (i.e., draw) how the variable `as` is represented in memory. Clearly show the byte-by-byte values and mark the lowest address and the highest address.

c) **(1 pt)** Visualize (i.e., draw) how the variable `au` is represented in memory. Clearly show the byte-by-byte values and mark the lowest address and the highest address.

20. (4 pts) Assuming the same system as our course VM, consider the following `struct` definition.

```
struct example {
  char a;
  int b;
  short c;
  double d;
};
```

a) **(1 pt)** How many bytes does the `struct` occupy in memory? Note that `char` is 1 byte, `int` is 4 bytes, `short` is 2 bytes, and `double` is 8 bytes. Assume that we do not use packing.

b) **(2 pts)** Explain why.

c) **(1 pt)** How many bytes does the `struct` occupy in memory if we use packing?