# Memory & OS Review

*COSC 208, Introduction to Computer Systems, 2022-05-05*

## Outline

- Memory hierarchy
- Caching
- Processes
- Scheduling
- Threads

## Memory hierarchy

Q1: What is the **fastest volatile** memory?

Q2: What is the **fastest non-volatile** memory?

Q3: Why is a hard disk drive (HDD) slower than a solid state drive (SSD)?

Q4: Why is accessing main memory (i.e., Random Access Memory (RAM)) slower than accessing a cache?

Q5: Why do solid state drives (SSDs) cost less per unit of capacity than main memory (i.e., Random Access Memory (RAM))?

## Caching

Q6: *Assume the cache size is 3 and the **optimal** cache replacement algorithm is used. Indicate what happens with the cache on each data access.*

- Access 2
- Access 4
- Access 1
- Access 2
- Access 4
- Access 3
- Access 2
- Access 4
- Access 1
- Access 2
- Access 4
- Access 1

Q7: *Assume the cache size is 3 and the **least recently used (LRU)** cache replacement algorithm is used. Indicate what happens with the cache on each data access.*

- Access 2
- Access 4
- Access 1
- Access 2
- Access 4
- Access 3
- Access 2
- Access 4
- Access 1
- Access 2
- Access 4
- Access 1

## Processes

Q8: *Write a program that creates a new process. The child process should print "I am a child"; the parent process should print "I am a parent; my child is CPID" (replacing CPID with the child’s PID).*

Q9: *Will the output produced by your program always appear in a particular order? Why or why not?*

## Scheduling

*Consider the following set of processes:*

| Process | Duration | Arrival Time |
|---|---|---|
| A | 20 | 0 |
| B | 15 | 0 |
| C | 25 | 5 |
| D | 5 | 10 |

Q10: *Draw the schedule when a First In First Out (FIFO) scheduling algorithm is used.*

Q11: *Compute the turnaround and wait time for each process based on the above schedule.*

| Process | Turnaround | Wait |
|---|---|---|
| A | | |
| B | | |
| C | | |
| D | | |

Q12: *Draw the schedule when a Shortest Job First (SJF) scheduling algorithm is used.*

Q13: _Compute the turnaround and wait time for each process based on the above schedule._

| Process | Turnaround | Wait |
|---|---|---|
| A | | |
| B | | |
| C | | |
| D | | |

Q14: *Draw the schedule when a Shortest Time to Completion First (STCF) scheduling algorithm is used.*

Q15: *Compute the turnaround and wait time for each process based on the above schedule.*

| Process | Turnaround | Wait |
|---|---|---|
| A | | |
| B | | |
| C | | |
| D | | |

Q16: *Draw the schedule when a Round Round (RR) scheduling algorithm is used with a time quantum of 10.*

Q17: *Compute the turnaround and wait time for each process based on the above schedule.*

| Process | Turnaround | Wait |
|---|---|---|
| A | | |
| B | | |
| C | | |
| D | | |

## Threads

*A program contains the following functions:*

```
void *dec(void *arg) {
    int *t = (int *)arg;
    *t--;
    return NULL:
}

void *inc(void *arg) {
    int *t = (int *)arg;
    *t++;
    return NULL;
}

void *zero(void *arg) {
    int *t = (int *)arg;
    *t = 0;
    return NULL;
}
```

*For each of the following main methods, list **all possible outputs** the program could produce. Assume threads are only preempted if they become blocked waiting for other threads.*

Q18:

```
int main() {
    int *total = malloc(sizeof(int));
    *total = 2;
    pthread_t thrA, thrB;
    pthread_create(&thrA, NULL, &inc, total);
    pthread_create(&thrB, NULL, &inc, total);
    pthread_join(&thrA);
    pthread_join(&thrB);
    printf("%d\n", *total);
}
```

Q19:

```
int main() {
    int *total = malloc(sizeof(int));
    *total = 2;
    pthread_t thrA, thrB;
    pthread_create(&thrA, NULL, &dec, total);
    pthread_create(&thrB, NULL, &zero, total);
    pthread_join(&thrA);
    pthread_join(&thrB);
    printf("%d\n", *total);
}
```

Q20:

```
int main() {
    int *total = malloc(sizeof(int));
    *total = 2;
    pthread_t thrA, thrB;
    pthread_create(&thrA, NULL, &zero, total);
    pthread_join(&thrA);
    pthread_create(&thrB, NULL, &inc, total);
    pthread_join(&thrB);
    printf("%d\n", *total);
}
```