Part 1: (80 + 8 (bonus) points - 4 points for each problem)

1. In an operating system a utility which reads commands from a terminal is called:
   (A) Terminal Handler (B) Kernel (C) Shell (D) None of the above

2. Which scheduler is responsible for controlling the degree of multiprogramming?
   (A) Admission scheduler (B) Memory scheduler (C) CPU scheduler (D) None of the above

3. Which statement about user-level threads and kernel threads is correct?
   (A) Both User-level and kernel threads can write into each other's memory space.
   (B) Both user-level and kernel threads use OS services via system calls.
   (C) Kernel thread scheduling is faster than user-level thread scheduling.
   (D) None of the above

4. Which is not able to solve the race condition?
   (A) Test and Set Lock (B) Shared memory (C) Semaphore (D) Monitor

5. CPU burst distribution is generally characterized as
   (A) Constant (B) Linear (C) Polynomial (D) Exponential or hyper-exponential

6. CPU Scheduling algorithms are used for:
   (A) Picking one of the ready processes in main memory to run next
   (B) Putting to sleep and waking up processes in an efficient manner
   (C) Allocating memory to the processes in a fair and efficient way
   (D) None of the above

7. Which is not a CPU scheduling criterion?
   (A) CPU utilization (B) Throughput (C) Waiting time (D) Burst time

8. Which is a preemptive scheduling?
   (A) SJF (B) FCFS (C) RR (D) None of the above

9. Which is not the necessary condition of a deadlock?
   (A) Mutual exclusion (B) Hold and wait (C) Preemption (D) None of the above

10. Which of the memory allocation schemes are subject to internal fragmentation?
    (A) Multiple Contiguous Fixed Partitions (B) Segmentation
        (C) Multiple Contiguous Variable Partitions (D) None of the above

11. A computer provides the user with virtual address space of $2^{24}$ words. Pages of size $2^{12}$ words. If the hexadecimal virtual address is 123456, the page number in hexadecimal would be:
    (A) 123 (B) 1234 (C) 456 (D) 3456

12. If there are 64 frames, and the frame size is 1024 words, the length of physical address is:
    (A) 15 bits (B) 16 bits (C) 17 bits (D) None of the above

13. The modified (dirty) bit is used for the purpose of:
    (A) Dynamic allocation of memory used by one process to another
    (B) Implementing FIFO page replacement algorithm
    (C) To reduce the average time required to service page faults
    (D) None of the above

14. Which page-replacement algorithm suffers from Belady’s anomaly?
    (A) Least recently used (LRU) (B) Clock (C) Not recently used (NRU) (D) None of the above
15. Working set model is:
(A) Used for finding the minimum number of frames necessary for a job, so that jobs can run without "thrashing"
(B) Used to find out the average number of frames a job will need in order to run smoothly without causing thrashing
(C) Used to determine whether page replacement is needed
(D) All of the above.

16. Which file allocation method suffers from disk fragmentation (except for internal fragmentation in the last block)?
(A) Linked list allocation (B) Contiguous allocation (C) I-nodes (D) None of the above

17. The UNIX system identifies a file as an executable binary file by
(A) File name (B) File extension (C) File descriptor (D) Magic number in the file header

18. A table points to routines that handle interrupts is called:
(A) Interrupt handler (B) Interrupt vector (C) Interrupt indicator (D) Interrupt signal

19. In which of the four I/O software layers is computing the track, sector, and head for a disk read done.
(A) User-level I/O software (B) Device-independent operating system software (C) Device drivers (D) Interrupt handlers

20. Which is an example of public-key cryptography?
(A) Caesar cipher (B) Transposition cipher (C) AES (D) PGP

21. A piece of code which lies dormant until triggered by by some event causing system damage is called:
(A) Logic bomb (B) Trap door (C) Virus (D) Worm

22. Which security method is not used in JVM?
(A) Sandboxing (B) Code Interpretation (C) Code signing (D) None of the above

Part 2: (120 + 10 (bonus) points)

1. (a) What is a process? Describe the process state.
   (b) What is a thread? Give two benefits of using threads.
   (12 points)

(a) A process is a running program.
   running: Instructions are being executed.
   ready: The process is waiting to be assigned to a process.
   blocked: The process is waiting for some event to occur.

(b) A thread is a lightweight process and a basic unit of CPU.
   • Responsiveness: Multiple activities can be done at same time. They can speed up the application.
   • Resource Sharing: Threads share the memory and the resources of the process to which they belong.
   • Economy: They are easy to create and destroy.
   • Utilization of MP (multiprocessor) Architectures: They are useful on multiple CUP systems.
2. How many processes will be created when the following program is executed?
Assume that all fork system calls are successful. What will be printed?
(Hint: Be careful and draw a picture.) (10 points)

```c
main()
{
    int i = 3;
    int ret_val;

    while(i > 0)
    {
        if ((ret_val = fork()) == 0) { /* Child's code */
            printf("In child \%d. \n", i);
            exit(0);
        } else { /* Parent's code */
            printf("In parent \%d. \n", i);
            i = i - 1;
        }
    }
}
```

There are 4 processes (1 parent and 3 child processes) created when this program is executed. The following could be printed:

```
In parent 3.
In child 3.
In parent 2.
In child 2.
In parent 1.
In child 1.
```

3. What are four criteria of a good solution to the critical-section problem? (8 points)

(a) Mutual exclusion is guaranteed.
(b) Progress is maintained. No process running outside its critical region may block other processes.
(c) Bounded waiting is assured. No process should have to wait forever to enter its critical region.
(d) No assumptions are made about the speeds of processes or the number of processors (CPUs).

4. Write a solution to the Producer-Consumer Problem using message passing (UNIX pipes). Recall that producers and consumers have access to a shared buffer that can hold up to N items. Use the notation `write(mbox, msg)` and `read(mbox, msg)`. (12 points)

```
Producer:

---------
- produce item -
recv(mbox1, msg);
send(mbox2, item);

Consumer:

---------
for (i=0; i<N; i++)
    send(mbox1, NULL_MSG);
recv(mbox2, item);
send(mbox1, NULL_MSG);
- consume item -
```
5. Suppose that the following processes arrive for execution at time 0 in the order A, B, C: (12 points)

<table>
<thead>
<tr>
<th>Process</th>
<th>Run Time</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>1=high</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>3=low</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

(a) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a non-preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 2) scheduling.

(b) What is the waiting time of each process for each of the scheduling algorithms?

(c) What is the turnaround time of each process for each of the scheduling algorithms?

(a) The four Gantt charts are

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>FCFS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>SJF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Waiting time:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>SJF</td>
<td>Priority</td>
<td>RR</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

(c) Turnaround time:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>SJF</td>
<td>Priority</td>
<td>RR</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>C</td>
<td>9</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

6. P is a set of processes. R is a set of resources. E is a set of request or assignment edges. The sets P, R, and E are as follows: (10 points)

- \( P = \{P_1, P_2, P_3\} \)
- \( R = \{R_1, R_2, R_3\} \)
- \( E = \{P_1 \rightarrow R_2, P_2 \rightarrow R_1, P_2 \rightarrow R_2, P_2 \rightarrow R_3, R_1 \rightarrow P_1, R_2 \rightarrow P_3, R_3 \rightarrow P_3\} \)

- \( R_1 \) has one instance.
- \( R_2 \) has two instances.
- \( R_3 \) has one instance.

(a) Draw the resource-allocation graph.

(b) Is there any deadlock in this situation? Briefly Explain.

(a) See the graph.

(b) Consider the resource-allocation graph. There are no cycle in the system.

P1, P2, and P3 are not deadlocked.
7. Given memory partitions of 100 KB, 400 KB, 200 KB, and 500 KB (in order). How would each of the first-fit, next-fit, best-fit, and worst-fit algorithms place processes of 200 KB, 396 KB, 100 KB, and 290 KB (in order)? (12 points)

First fit : (a) 400 KB (b) 500 KB (c) 100 KB (d) No fit
Next fit : (a) 400 KB (b) 500 KB (c) remainder of 500 KB (d) No fit
Best fit : (a) 200 KB (b) 400 KB (c) 100 KB (d) 500 KB
Worst fit : (a) 500 KB (b) 400 KB (c) remainder of 500 KB (d) No fit

8. A small computer has five page frames. At the first clock tick, the R bits are 10011 (page 1 and 2 are 0). A subsequent clock ticks, the values are 10100, 10101, 00101, 01100, 01011, and 10101. If the aging algorithm is used, with a 4-bit counter, give the values of the counters after the last tick. Which page would be selected to be removed from memory? (12 points)

The page 3 will be evicted because it has the smallest counter value.

9. Consider a computer is equipped with associative memory that can hold 16 entries of the page table and can be accessed in 10 nanoseconds. The hit ratio is the percentage of the page table entry can be found in the associative memory. The CPU takes total 130 nanoseconds to search the page entry and access a data item when the page entry is not in the associative memory. (10 points)

(a) Find a formula that expresses the effective access time as a function of the hit ratio (h).

(b) What hit ratio is needed to achieve the effective access time to 82 nanoseconds?

(a) Let E = associative memory lookup time, T = memory cycle time, h = hit ratio.

\[ E + 2 \times T = 130, \ E = 10 \Rightarrow T = 60. \ E + T = 70 \]

Effective Access Time (EAT) = \[ 70 \times h + (1 - h) \times 130 = 70 \times h + 130 - 130 \times h = 130 - 60 \times h \]

(b) \[ 130 - 60 \times h \leq 82, \ 60 \times h \geq 130 - 82, \ 60 \times h \geq 48, \ h \geq 80\% \]

10. Suppose there are 16 virtual pages and 4 page frames. Determine the number of page faults that will occur with the reference string 0 1 2 3 2 4 1 3 5 6 1 3 2 7 4 8, if the page frames are initially empty, using each of the following page replacement algorithms: (a) FIFO (b) LRU (c) Optimal. (12 points)

(a) FIFO: 13 (b) LRU: 11 (c) Optimal: 10

11. Encrypt the following plaintext using a transposition cipher based on the key LEOPARD. (10 points)

plaintext = H A V E A G R E A T S U M M E R B R E A K

\[
L E O P A R D
4 3 5 6 1 7 2
H A V E A G R
E A T S U M M
E R B R E A K
\]

ciphertext = A U E R M K A A R H E E V T B E S R G M A
12. What is a virus? What is a polymorphic virus? Give two examples of anti-virus products. (10 points)

- Virus is a program can reproduce itself by attaching its code to another program and additionally do harm.
- A virus that mutates on each copy is called a polymorphic virus.
- Norton Anti-virus, McAfee VirusScan