1.

a) Since second generation OS does not support multiprogramming.

b)
   a. Protection between jobs
   b. Job scheduling

c) A situation where two or more processes are reading or writing some shared data and the final result depends on who runs precisely when, are called race condition.

d) Since context switch between processes need very expensive overhead.

e) Aging

f) Limited size of memory, big size processes.

g) smaller page size means bigger page table, bigger page size means internal fragmentation.

h) OS involve in creation of both. OS take care mutual exclusion and synchronization for message queue but not shared memory.

i) MMU

j)
   a. Throughput – number of processes completed per time unit
   b. Turnaround time – the interval from the time of submission to the time of completion of a process.

2.

- Sol) maximum virtual address space = $2^{32} = 2^{22} \times 2^{10} = 2^{22}$ **KB**
  $\therefore$ Maximum # of pages per a process = virtual space / a page size = $2^{22}/8 = 2^{19}$ pages .
  Maximum size of page table per a process = number of page $\times$ one entry size
  = $2^{19} \times 64$ bits = $(2^{19} \times 64)/8$ Byte= $2^{19} \times 8$ byte= **4 MB**

- Sol) simply need calculate the number of page frame
  # of page frame = size of RAM / size of page = 8GB / 8KB = 8×2^{30} / 8×2^{10} = 2^{20} page frames $\therefore$ **20 bits for page frame number.**

3.

1. Lets assume count is currently N, producer read check count == N and then time out before go to sleep.
2. Consumer remove one item from the buffer and reduce count to N-1. The consumer check count is N- 1, call wakeup() to wake up the producer. Since producer did not sleep yet, the signal will be lost.
3. Control back to producer, the producer already checked count =N, so go to sleep. Control back to consumer, consumer will consume all items eventually, then go to sleep. Now, both consumer and producer sleep

4.

a)

virtual address $19845/(4 \times 2^{10}) = 4.84$ in page #4 (map to page frame #135).
virtual page #4 begin with address $4 \times 4 \times 2^{10}$
page frame #135 is start with address $135 \times 4 \times 2^{10}$
physical address = $135 \times 4 \times 2^{10} + (19845 - 4 \times 4 \times 2^{10}) = 552{,}960 + 3{,}461 = 556{,}421$

b)

virtual address $21581/(4 \times 2^{10}) = 5.26$ in page #5
virtual page #5 begin with address $5 \times 4 \times 2^{10}$
page frame #95 is start with address $95 \times 4 \times 2^{10}$
physical address = $95 \times 4 \times 2^{10} + (21581 - 5 \times 4 \times 2^{10}) = 389120 + 1101 = 390221$

5.

a)

$1 - p^n$

b)

n= (8 GB – 1 GB)/1GB = 7
p= 0.9   CPU utilization $=1- (0.9)^7 = 0.5217$ about 52%

c)

n = (16 GB – 1GB) / 1GB = 15, p = 0.9 $1 – (0.9)^{15} = 0.794$ about 79 %

6.

a) 15 page fault

5, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 5, 0, 1

5	0	1	2		3	0	4	2	3	0			1	2			5	0	1
	5	0	1		2	3	0	4	2	3			0	1			2	5	0
		5	0		1	2	3	0	4	2			3	0			1	2	5

b) 12 page fault

5, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 5, 0, 1

5	0	1	2		2		4	4	4	0			1		1		1		
	5	0	1		3		3	2	2	2			2		2		5		
		5	0		0		0	0	3	3			3		0		0		

c) 9 page fault

5, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 5, 0, 1

5	0	1	1		3		3			3			1				1		
	5	0	0		0		4			0			0				0		
		5	2		2		2			2			2				5		

7.

- 2^{14} = 16 KB
- 2^{25} page tables +1
- 2^{25} x 2^{25} pages
- There are 8GB /16 KB = $8 \times 2^{30} / 16 \times 2^{10}$ = 2^{19} page frames. The system need reserve 19 bit for page frame number.

8.

Sol) let's assume : empty = 0, full = N, mutex =1

Producer is scheduled : produce item ,down mutex (now mutex =0), try to down empty. Since empty =0, producer cannot finish down operation and sleep on semaphore empty.

Consumer is scheduled: down full (now full = N-1), then try to down mutex. Since mutex is already down by producer, consumer cannot finish down operation and sleep on semaphore mutex.

Now producer and consumer sleep forever!

9.

- For 2KB: page number = 45250 / 2 x 2^{10} = 22 offset= (45250 – 22 x 2 x 2^{10}) = 194
- For 4KB: page number= 45250 / 4 x 2^{10} = 11 offset = (45250 – 11 x 4 x 2^{10})= 194

10.

- Bitmap: #of allocation unit = 2GB/4KB = $(2\times 2^{30})/(4 \times 2^{10}) = 2^{31}/2^{12} = 2^{19}$ units
  Size of the bitmap = 2^{19} bits = **2^{16} Byte**

- The linked list: number of node for linked list= 2G /128KB = $2 \times 2^{30}/128 \times 2^{10} = 2^{31}/2^{17} = 2^{14}$ nodes.
  size of each node = 32+16+16 = 64 bit = 8 byte =2^3 bytes
  Total size of linked list = number of node $\times$ size of a node = $2^{14} \times 2^3$ bytes = **2^{17} bytes = 128 KB**.

11.

Sol) If each job has 50% I/O wait, then it will take 40 minutes to complete in the absence of competition. If run sequentially, the second one will finish 80 minutes after the first one starts.
With two jobs, the approximate CPU utilization is $1 - 0.5^2$=0.75. Thus, each one gets 0.375 CPU minute per minute of real time. To accumulate 20 minutes of CPU time, a job must run for20/0.75 +20/0.75 minutes, or about 53.33 minutes. Thus running sequentially the jobs finish after 80 minutes, but running in parallel they finish after 53.33 minutes.