1.

a)

Since page size is 100 bytes, virtual space of 780 byte program can be saved in 8 pages. Page 0 ~ page 7

Sol) 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

b)

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		

12 page fault

c)

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

7	7	7	2		2		2			2			2				7		
	0	0	0		0		4			0			0				0		
		1	1		3		3			3			1				1		

9 page faults

2.

- Bitmap: #of allocation unit = 128MB/2KB = $(128 \times 2^{20})/(2 \times 2^{10}) = 2^{27}/2^{11} = 2^{16}$ units
  Size of the bitmap = 2^{16} bits = **2^{13} byte**
- The linked list: number of node for linked list= 128 MB /64KB = $2^{27}/2^{16}$ or 2^{11} 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 = $2^{11} \times 2^3$ bytes = **2^{14} bytes**.

3.

Sol)

- At time T_0 :Let's assume at time T_0 : empty = N, full = 0, mutex =1.
  - Consumer is scheduled by CPU scheduler. consumer down mutex (now mutex =0), try to down full. Since full =0, consumer cannot finish down operation and sleep on semaphore full.
- At time T_1 :
  - Producer is scheduled by CPU scheduler. Consumer try to down empty (now emputy = N-1) , then try to down mutex. Since mutex is already down by consumer, producer cannot finish down operation and sleep on semaphore mutex.
- Now producer and consumer both sleep ! -Deadlock

4.

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 <u>0.375 CPU minute per minute</u> 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.

5.

a)

virtual address 46851/(4×2^{10}) = 11 in page #11.
virtual page #11 begin with address $11 \times 4 \times 2^{10}$
page frame #7 is start with address $7 \times 4 \times 2^{10}$
physical address = $7 \times 4 \times 2^{10}$ + (47891 - $11 \times 4 \times 2^{10}$) = 28672 + 2835 = 31507

b)

virtual address 8765/(4×2^{10})= 2 in page #2
virtual page #2 begin with address $2 \times 4 \times 2^{10}$
page frame #1100 is start with address $1100 \times 4 \times 2^{10}$
physical address = $1100 \times 4 \times 2^{10}$+ (8765 - $2 \times 4 \times 2^{10}$) = 4505600 + 573 = 4506173

6.

P_1	P_2	P_3	P_2	P_3	P_4	P_1

0 2 6 10 13 16 21 27

Average Waiting time = ((21 – 2) + (10 – 6) + ((6 – 4) + (13 – 10)) + (16 – 6))/4 = 19+4+5+10/4= 9.5

Average Turnaround time = ((27 – 0) + (13 – 2) + (16 – 4) + (21 – 6))/4 =27+11+12+15/4= 16.25

7.

- What is the size of each page (KB)?
  2^{24} = 2^{14} KB = 16MB
- How many page tables are there?
  1 + 2^{20} page tables
- How many pages are there?
  there are 2^{20} page table and each has 2^{20} entries. # of pages = $2^{20} \times 2^{20}$ = 2^{40} pages
- If system has 8GB memory, how many bits need to be reserved for saving page frame number in the each of page table entry?
  Sol) # of page frames = 8GB /16MB = 8×2^{30} / 16×2^{20} = 2^{33} /2^{24} = 2^9 page frames. The system needs reserve 9 bit for page frame number.

8.

a)

1 - p^n

b)

n= (8 GB – 2 GB)/1GB = 6

p= 0.9 CPU utilization =1- $(0.9)^6$ = 0.468 about 46%

c)

n = (16 GB – 2GB) / 1GB = 14, p = 0.9 1 – $(0.9)^{14}$ = 0.771 about 77 %

9. Short answer questions
   a) OS take care mutual exclusion and synchronization for message queue but not shared memory.
   b) Try to use less memory space for page table with different structure.
   c) Try to reduce search time due to a page fault
   d) since second generation OS does not support multiprogramming.
   e) process status, snapshot of CPU, scheduling information, memory management information I/O status information
   f)
      a. Protection between jobs
      b. Job scheduling
   g) Since instruction cycle are three steps: fetch, encoding, and execute
   h) limited size of memory, big size processes.

10.

- For 3KB: page number = 35250 / 3 x 2^{10} = 11 offset= (35250 – 11 x 3 x 2^{10}) = 1458
- For 4KB: page number= 35250 / 4 x 2^{10} =8 offset = (35250 – 8 x 4 x 2^{10})= 2482

11.

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.
4. Control back to consumer, consumer will consume all items eventually, then go to sleep. Now, both consumer and producer sleep forever.

12.

- Sol) 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 = $2^{22}/2 = 2^{21}$ pages .
  Size of page table per a process = number of page $\times$ one entry size
  = $2^{21} \times 64$ bits = $(2^{21} \times 64)/8$ Byte= $2^{21} \times 8$ byte= **2^{24} Byte = 16 MB**

- Sol) simply need calculate the number of page frame
  # of page frame = size of RAM / size of page = 4GB / 2KB = $4 \times 2^{30} / 2 \times 2^{10} = 2^{32}/2^{11} = 2^{21}$ page frames $\therefore$ **21 bits for page frame number.**

13.

◊ First Fit 9(21) -10(12) -15(20)- 18(18)

◊ Best Fit 9(14) – 10(15)-15(18)-18(20)

◊ Worst Fit 9 (35) – 10(26) -15 (25) – 18 (23)

◊ Next Fit 9(21)-10(20)-15(18) -18(25)

14.

lets assume Permit =0 at time T0
P0 tries to enter C.S. and can enter since Permit =0.
P0 finish its job in C.S. and set Permit =1
P1 is currently running outside C.S ,it is terminated with fatal error.
P0 tries to enter C.S. again but P0 never can.