Solution

Problem 1: (26points)

1. [1] TLB [2] VPN1 [3] PD [4] VPN2 [5] PT

2.

Parameter	Value
TLB Index	0x0
TLB Tag	0x0A
TLB Hit?（Y/N）	Y
VPN-1	0x0
VPN-2	0x28
Page Fault?（Y/N）	N
PPN	0x6D
Physical Address	0x3684

Parameter	Value
TLB Index	0x2
TLB Tag	0x180
TLB Hit?（Y/N）	N
VPN-1	0x06
VPN-2	0x02
Page Fault?（Y/N）	N
PPN	0x05
Physical Address	0x281

Problem 2: Networking (25 points)

1. [1] DEFAULT_IP

   [2] htons(DEFAULT_PORT)

   [3] (struct addr*)serveraddr

   [4] sizeof(serveraddr)

   [5] rio_readn(clientfd, buf, MAXBUF);

   [6] rio_writen(clientfd, buf, n);

   [7] bind(listenfd, (struct sockaddr*)&serveraddr, sizeof(serveraddr);

   [8] accept(listenfd, (struct sockaddr*)&clientaddr, &clientlen)

2. Two threads may modify ‘bytes_total’ concurrently.

   Add a mutex to protect it.

3. The memory resources can be freed automatically by the system when it terminates. It is hard for the server to explicitly wait for each peer thread to terminate too.

Problem 3: Signal (20 points)

1. [1] bc [2] a

2. 4 and 5

3. [1] 11 or 21 [2] 0 or 100

4. (ABBDDFG, AAJKLMN)

   or (ABBDDFG, ddJKLMN)

   or (ABdDDFG, AAJKLMN)

Problem 4: Virtual Memory (29 points)

1. [1] ARRAY_LEN * sizeof(int) [2] PROT_WRITE

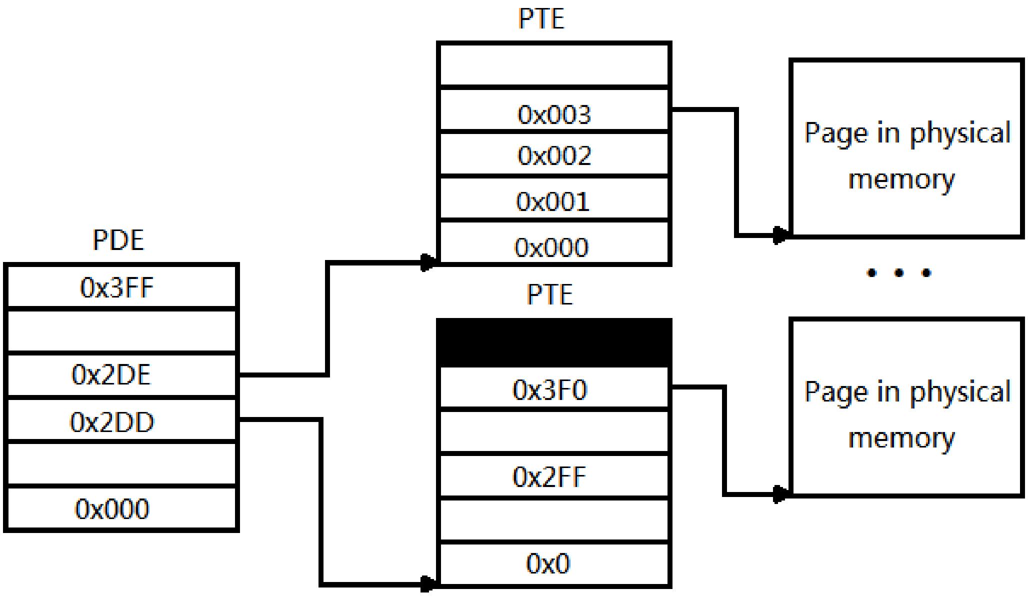
   [3] MAP_SHARED | MAP_ANON or MAP_ SHARED | MAP_ANONYMOUS

   [4] ARRAY_LEN * sizeof(int) [5] PROT_WRITE

   [6] MAP_ PRIVATE | MAP_ANON or MAP_ PRIVATE | MAP_ANONYMOUS

2. [1] 21

3.



4. [1] 20

   [2] Benefits: use less memory to fork a new process. Faster and more efficiency.