**Experiment 7: Implementation of bit map protocol**

#include <stdio.h>

#define MAX\_BLOCKS 32

void displayBitmap(int bitmap[], int n) {

printf("Current Bitmap: ");

for (int i = 0; i < n; i++) {

printf("%d ", bitmap[i]);

}

printf("\n");

}

int allocateBlock(int bitmap[], int n) {

for (int i = 0; i < n; i++) {

if (bitmap[i] == 0) {

bitmap[i] = 1;

printf("Allocated Block: %d\n", i);

return i;

}

}

printf("No Free Blocks Available.\n");

return -1;

}

void deallocateBlock(int bitmap[], int block) {

if (bitmap[block] == 1) {

bitmap[block] = 0;

printf("Block %d deallocated.\n", block);

} else {

printf("Block %d is already free.\n", block);

}

}

int main() {

int bitmap[MAX\_BLOCKS] = {0}; // Initialize all blocks as free

int choice, block;

// Display the menu only once at the beginning

printf("\*\*\*\*\*\*\*\*\*\* Bit Map Protocol \*\*\*\*\*\*\*\*\*\*\n");

printf("Menu:\n");

printf("1. Allocate Block\n");

printf("2. Deallocate Block\n");

printf("3. Display Bitmap\n");

printf("4. Show Menu Again\n");

printf("5. Exit\n");

while (1) {

printf("\nEnter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

allocateBlock(bitmap, MAX\_BLOCKS);

break;

case 2:

printf("Enter block number to deallocate (0 - %d): ", MAX\_BLOCKS - 1);

scanf("%d", &block);

if (block >= 0 && block < MAX\_BLOCKS)

deallocateBlock(bitmap, block);

else

printf("Invalid block number.\n");

break;

case 3:

displayBitmap(bitmap, MAX\_BLOCKS);

break;

case 4:

// Option to display menu again if user wants to see it

printf("\nMenu:\n");

printf("1. Allocate Block\n");

printf("2. Deallocate Block\n");

printf("3. Display Bitmap\n");

printf("4. Show Menu Again\n");

printf("5. Exit\n");

break;

case 5:

printf("Exiting...\n");

return 0;

default:

printf("Invalid choice. Try again.\n");

}

}

return 0;

}

**Sample Output:**

\*\*\*\*\*\*\*\*\*\* Bit Map Protocol \*\*\*\*\*\*\*\*\*\*

Menu:

1. Allocate Block

2. Deallocate Block

3. Display Bitmap

4. Show Menu Again

5. Exit

Enter your choice: 3

Current Bitmap: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Enter your choice: 1

Allocated Block: 0

Enter your choice: 3

Current Bitmap: 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Enter your choice: 2

Enter block number to deallocate (0 - 31): 0

Block 0 deallocated.

Enter your choice: 4

Menu:

1. Allocate Block

2. Deallocate Block

3. Display Bitmap

4. Show Menu Again

5. Exit

Enter your choice: 5

Exiting...

1. **Explain ARP:**  
   ARP (Address Resolution Protocol) is used to map an IP address to a physical MAC address within a LAN.
2. **Explain the purpose of network layer:**  
   The network layer is responsible for logical addressing, routing, and forwarding of data packets from source to destination across networks.
3. **What is the use of ARP?**  
   ARP is used by devices in a local network to determine the MAC address associated with a known IP address.
4. **To which OSI layer does ARP belong?**  
   ARP is considered to operate between the Data Link layer (Layer 2) and the Network layer (Layer 3) in OSI.
5. **Explain RARP:**  
   RARP (Reverse Address Resolution Protocol) allows a device to discover its IP address using its known MAC address.
6. **What is the difference between ARP and RARP?**  
   ARP maps IP addresses to MAC addresses, while RARP maps MAC addresses to IP addresses mainly for booting systems.