**EXPERIMENT - 8**

## Relocation loader

## Aim

To implement a relocation loader

**Program**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int get\_bit(int n, int pos)

{

    int offset = 12 - pos;

    int mask = 1 << offset;

    int and\_result = n & mask;

    int bit = and\_result >> offset;

    return bit;

}

void main()

{

    FILE \*record\_ptr;

    int program\_len = 0x0, starting\_addr = 0x0, relocationOffset, headerRecordLengthMismatch = 1;

    char temp[300];

    printf("Enter relocation address in HEX: ");

    scanf("%x", &relocationOffset);

    printf("Reading record\n");

    record\_ptr = fopen("record.txt", "r");

    while (fscanf(record\_ptr, "%[^\n]%\*c", temp) != EOF)

    {

        char instructions[40][300];

        char \*token = strtok(temp, "^");

        int text\_len = 0;

        while (token != NULL)

        {

            strcpy(instructions[text\_len], token);

            token = strtok(NULL, "^");

            text\_len++;

        }

        if (strcmp("H", instructions[0]) == 0)

        {

            starting\_addr = strtol(instructions[2], NULL, 16);

            program\_len = strtol(instructions[3], NULL, 16);

            starting\_addr = relocationOffset;

            printf("\nProgram Name: %s\n", instructions[1]);

            printf("Program Starting Address: %X\n", starting\_addr);

            printf("Program Length: %s\n", instructions[3]);

            printf("\n%6s |  %6s\n", "ADDRESS", "OBJCODE");

            printf("------------------\n");

        }

        else if (strcmp("T", instructions[0]) == 0)

        {

            int text\_starting\_addr = strtol(instructions[1], NULL, 16) + relocationOffset;

            int bitmask = strtol(instructions[3], NULL, 16);

            if (headerRecordLengthMismatch == 1)

            {

                headerRecordLengthMismatch = 0;

                if (starting\_addr != text\_starting\_addr)

                {

                    for (int j = starting\_addr; j < text\_starting\_addr; j += 3)

                        printf("%06X  |  %6s\n", j, "XXXXXX");

                }

            }

            for (int j = 4, i = 1; j < text\_len; j++, i++)

            {

                int obj\_code = strtol(instructions[j], NULL, 16);

                if (get\_bit(bitmask, i)) *// REALOCATE*

                    printf("%06X  |  %06X\n", text\_starting\_addr, obj\_code + relocationOffset);

                else

                    printf("%06X  |  %06X\n", text\_starting\_addr, obj\_code);

                text\_starting\_addr += strlen(instructions[j]) / 2;

            }

        }

        else if (strcmp("E", instructions[0]) == 0)

            break;

        else

            printf("ERROR: UNKNOWN RECORD");

    }

    fclose(record\_ptr);

    printf("\nRelocation Loader done...\n");

}

**Input**

**record.txt**

H^BUFTOR^000000^00101C

T^000006^19^FE0^043016^E03015^303003^50B019^DC3015^2C4019^383003^05^000000

E^000000

**Output**

