**PAGING TECHNIQUE OF MEMORY MANAGEMENT**

**Ex.No:12**

# AIM:

To write a c program to implement Paging technique for memory management.

# ALGORITHM:

Step 1: Start the process

Step 2: Declare page number, page table, frame number and process size. Step 3: Read the process size, total number of pages

Step 4: Read the relative address

Step 5: Calculate the physical address Step 6: Display the address

Step 7: Stop the process

# PROGRAM:

#include<stdio.h> #include<string.h> #include<pthread.h> #include<stdlib.h> #include<unistd.h> pthread\_t tid[2];

int counter; pthread\_mutex\_t lock;

void\* doSomeThing(void \*arg)

{

pthread\_mutex\_lock(&lock); unsigned long i = 0;

counter += 1;

printf("\n Job %d started\n", counter); for(i=0; i<(0xFFFFFFFF);i++);

printf("\n Job %d finished\n", counter); pthread\_mutex\_unlock(&lock);

return NULL;

}

int main(void)

{

int i = 0; int err;

if (pthread\_mutex\_init(&lock, NULL) != 0)

{ printf("\n mutex init failed\n"); return 1;

}

while(i < 2)

{

err = pthread\_create(&(tid[i]), NULL, &doSomeThing, NULL); if (err != 0)

printf("\ncan't create thread :[%s]", strerror(err)); i++;

}

pthread\_join(tid[0], NULL); pthread\_join(tid[1], NULL); pthread\_mutex\_destroy(&lock); return 0;}