Create a thread:

#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

int MAX = 100;

volatile int count = 0;

pthread\_mutex\_t mutex;

pthread\_cond\_t cond;

void \*printEvenNum(void \*arg)

{

while(count < MAX)

{

pthread\_mutex\_lock(&mutex);

while(count % 2 != 0)

{

pthread\_cond\_wait(&cond, &mutex);

}

printf("%d ", count++);

pthread\_mutex\_unlock(&mutex);

pthread\_cond\_signal(&cond);

}

pthread\_exit(0);

}

void \*printOddNum(void \*arg)

{

while(count < MAX)

{

pthread\_mutex\_lock(&mutex);

while(count % 2 != 1)

{

pthread\_cond\_wait(&cond, &mutex);

}

printf("%d ", count++);

pthread\_mutex\_unlock(&mutex);

pthread\_cond\_signal(&cond);

}

pthread\_exit(0);

}

int main()

{

pthread\_t thread1;

pthread\_t thread2;

pthread\_mutex\_init(&mutex, 0);

pthread\_cond\_init(&cond, 0);

pthread\_create(&thread1, 0, &printEvenNum, NULL);

pthread\_create(&thread2, 0, &printOddNum, NULL);

pthread\_join(thread1, 0);

pthread\_join(thread2, 0);

pthread\_mutex\_destroy(&mutex);

pthread\_cond\_destroy(&cond);

return 0;

}

Output:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

--------------------------------

Process exited after 0.8577 seconds with return value 0

Press any key to continue . . .