***Context :***

Two important concepts, which are very essential when we use multithreading are “Thread Safety” and using “Reentrant Functions”.

Please study the links below to understand these two concepts.

<http://www.ibm.com/support/knowledgecenter/ssw_aix_61/com.ibm.aix.genprogc/writing_reentrant_thread_safe_code.htm>

<http://www.ibm.com/developerworks/library/l-reent/>

<http://stackoverflow.com/questions/856823/threadsafe-vs-re-entrant>

One code fragment is also given below, which is not a correct code in a multithreaded environment. Please figure why this is not a correct code and what can be done to make it suitable in a multithreaded environment.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FILENAME : Thread\_Reentrant.c

\* DESCRIPTION:Contains Code for a program that demonstrates the

\* use of non reentrancy in strtok function...

\* If the code works correctly, it should print all the tokens,

\* which are there in teststring1, and teststring2.

\* Invoke the Executable as a.out (gcc Thread\_Reentrant.c -lpthread)

\* Copyright 2016 Aricent

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<pthread.h>

#include<string.h>

#include<stdio.h>

#include<stdlib.h>

#include<sys/types.h>

void printids(const char \* s);

void \* th\_f1(void \*arg);

void \* th\_f2(void \*arg);

char teststring1 [] = "This is a Test string let us see 1 2 3 4 5 6 7 8 9 0";

char teststring2 [] = "Delhi Kolkata Mumbai Gurgaon";

int main(void)

{

pthread\_t ntid1, ntid2; /\*Thread Identifier\*/

int status;

/\*Creating a thread\*/

status=pthread\_create(&ntid1,NULL,th\_f1,(void\*)teststring1);

if (status != 0)

{

printf("Error in Creating Thread\n");

exit(status);

}

status=pthread\_create(&ntid1,NULL,th\_f2,(void\*)teststring2);

if (status != 0)

{

printf("Error in Creating Thread\n");

exit(status);

}

pthread\_exit(NULL);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FUNCTION NAME:th\_f

\* DESCRIPTION:A Thread Entry Point.

\* NOTES : No Error Checking is done .

\* RETURNS :void \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void \* th\_f1(void \*arg)

{

char \*token = strtok((char\*)arg, " ");

while(token != NULL)

{

printf("Thread %u [%s]\n", pthread\_self(), token);

//If I comment sleep the code may work, as intended

//That is the strings are tokenised by the two threads

//Though we cannot depend on that ...

//The function strtok is not a rentrant function.

sleep(2);

token = strtok(NULL, " ");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FUNCTION NAME:th\_f2

\* DESCRIPTION:A Thread Entry Point.

\* NOTES : No Error Checking is done .

\* RETURNS :void \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void \* th\_f2(void \*arg)

{

char \*token = strtok((char\*)arg, " ");

while(token != NULL)

{

printf("Thread %u [%s]\n", pthread\_self(), token);

token = strtok(NULL, " ");

}

}