THREADMENTOR: MULTI-THREAD PROCESSING

By RIJA ,ARIBA ,HAMMAD FATIMA,SHAHID,HUSSAIN

WORD COUNT	738	TIME SUBMITTED	28-NOV-2018 10:05AM
		PAPER ID	42325649

RESEARCH ON

THREADMENTOR: MULTI-THREAD PROCESSING

RIJA FATIMA(59631)

ARIBA SHAHID(59385)

HAMMAD HUSSAIN(59490)

(BSCS)

PAF-KIET

ABSTRACT: (ARIBA SHAHID)

This paper presents our effort in designing pedagogical tools for teaching message passing using channels. These tools include a class library that supports channels, a visualization system that helps students see the execution behavior of threads and message passing, and a topology editor that provides an environment for students to design network topologies. Moreover, since we have made sure the uniformity of the channel definition across the thread, parallel and distributed environments, porting a threaded program to a parallel/distributed environment is easy.

OBJECTIVE (Hammad Hussain)

Thread Mentor [1] is a multiplatform pedagogical tool designed to ease the difficulty in teaching and learning multithreaded programming. It consists of a C++ class library and a visualization system. The class library supports many thread management functions and synchronization primitives in an object-oriented way, and the visualization system is activated automatically by a user program and shows the inner working of every thread and every synchronization primitive on-the-fly. Events can also be saved for playback. In this way, students will be able to visualize the dynamic behavior of a threaded program and the interaction among threads and synchronization primitives.

DESIGN (RIJA FATIMA)

Thread mentor [1] consists of a class library and a visualization system. Both components support thread management (e.g., thread creation, termination, and join) and popular synchronization primitives (e.g., mutex locks, semaphores, monitors, barriers, and synchronous and asynchronous channels). The class library uses textbook syntax so that students do not have to memorize many different parameters, and hides as many system details as possible from its users.[2] the class library translates many thread management and synchronization primitive calls to the corresponding system-supported thread library calls.

METHODOLOGY

(ARIBA)

Numerous papers have been distributed in the SIGCSE Technical Symposium what's more, ITiCSE Conference on multithreaded, multiprocess, parallel and appropriated figuring. There are not very many instructive apparatuses for showing strung programming. Most instruments are varieties of Ben-Ari's Pascal compiler concentrating for the most part on running strings or procedures under the control of a translator with different sorts of synchronization natives. Lester's Multi-Pascal [Lester 1993] is likewise a translator with restricted parallel computing capacity. There are some Java-based devices for disseminated calculations [Ben-Ari 1997, 2001; Schreiner 2002].

(RIJA)

Picturing parallel projects can be on the web or disconnected. The previous produces visuals on-the-fly, while the last spares the occasions and plays back with another framework (i.e., after death). The upside of a disconnected framework is that each occasion identified with the execution of a program is spared and can be replayed whenever.
Be that as it may, its primary hindrances are

- It could be past the point of no return for an understudy to get any bug, since it just shows one example of the program execution;
- An expansive volume of yield might be created that could be fragmented or even ruined if the program terminates unexpectedly.
- The program being envisioned must be instrumented by including additional announcements as well as mandates, which could straightforwardly meddle with the conduct of the program;

- The disconnected framework should likewise synchronize its very own document composing exercises, including an additional dimension of unpredictability that may influence the program's unique conduct.

(HAMMAD)

The best-known arrangement of this type is PARADE [Stasko 1995], which depends on POLKA. Different points of interest can be found in Flinn and Cowan [1990]; Kraemer [1998]; and Stasko and Kraemer [1993a, 1993b]. Along this line of research, Zhao and Stasko [1995] planned an condition for imagining Pthreads programs utilizing POLKA; and Cai et al. [1993] talked about a framework for envisioning programs written in OCCAM. Other valuable data might be found in Bemmerl and Braun [1993]; Chao and Liu [2000]; Miller [1993]; Pancake [1996]; Roman et al. [1992]; Zhang et al. [1999].

(ARIBA)

In any case, nothing unless there are other options referenced frameworks firmly bolster multi threaded programming and its perception, and furnish understudies with a condition for creating strung projects and envisioning program execution and synchronization exercises. In addition, with the exception of Zhao and Stasko [1995], none of the frameworks can uncover the low-level synchronization related data. Truth be told, the greater part of the representation frameworks are for performance as well as investigating as opposed to structured as academic stages to be utilized by tenderfoots and understudies. Thusly, ThreadMentor is maybe the just complete academic framework accessible for instructing and learning multithreaded programming.

THREADMENTOR: MULTI-THREAD PROCESSING

ORIGINALITY REPORT

62%

SIMILARITY INDEX

PRIMARY SOURCES

1	www.cs.mtu.edu Internet	420 words — 55%
2	cs.mtu.edu Internet	47 words — 6%
3	Steve Carr, Jean Mayo, Ching-Kuang Shene. "ThreadMentor", Journal on Educational Resources in Computing, 2003 Crossref	8 words — 1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		