

Fall 2009 CIS Colloquium Series

Bare Machine Computing (BMC)

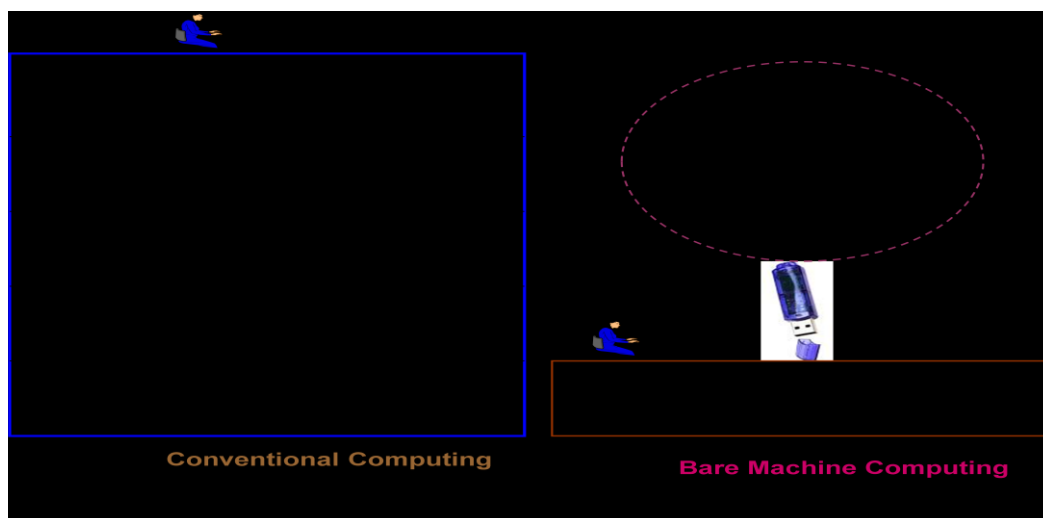
**Ramesh K. Karne and Alexander L. Wijesinha
(Towson University)**

*11am-12pm, Wednesday, November 4
4th Floor Conference Room (Wachman Hall, CC 447)*

An operating system (OS) serves as a middleware between hardware and applications. The OS provides higher levels of abstractions to application programs thus hiding the execution intricacies from them. Modern OSs such as Microsoft Windows and Linux provide such middleware layer for general purpose applications. These OSs have grown in complexity and size over the years resulting in inefficiency and open for security vulnerabilities. A typical OS for a PC may contain 30 to 40 million lines of code and it changes almost on a yearly basis. The large OSs requires longer time to boot, load, and run applications. Lean kernel and Exokernel approaches provide improved performance and possibly more security by moving some critical functionality from OS to application programs. The bare machine computing paradigm goes to the other extreme to avoid OS middleware altogether by dispersing the necessary operating environment into applications.

When a machine is made bare, there is no centralized software running in it to provide an execution environment for the applications, and there is no resident persistent storage. Only a CPU, Memory, and necessary detachable I/O devices are used on the bare machine making the machine ownerless. Each end user application carries its own operating environment with it. The operating environment for each end user application is extremely small and it is not same as OS. Each end user application is referred to as an application object (AO). An AO may consist of one or more typical applications such as Email, Web search, and Web browsing, which may be used by the end user at a given point in time. One or more AOs can be built to suite the user needs and they can be carried on a memory stick (USB mass storage) to run on any bare machine. This approach allows users to carry their own applications in their hand and run them on any available bare machine anywhere. Thus, any user can use any machine available either at home, office, or a public place. There is no need to have an ownership to a machine and there are no valuable assets on the bare machine to be protected. The bare machine computing provides tremendous advantages including: simplicity, small code size, single programming environment, independent of OS platforms, longevity, inherent security, and less prone to obsolescence. This approach inherits its own drawbacks such as non-traditional programming, difficulty in porting existing applications on to bare machines, and making the AO programmer's job harder compared to an application programmer.

In this presentation, we will describe several aspects of bare machine computing including: our motivation, benefits, principles, architecture, current implementations, its impact on computer architecture and software architecture, and new direction to polarize end user applications. We will also entertain questions from audience to stimulate discussions on pros and cons of this computing paradigm.



Bio: Dr. Ramesh K. Karne is a professor in the department of Computer and Information Sciences at Towson University. He is also the program director for the doctoral program. Before joining Towson University, he has worked at the Institute of Systems research at College Park and at IBM for many years. His research interest over last ten years has been in Bare Machine Computing. He is a recipient of NSF SGER grant to pursue the bare computing research and also NSF REU grant to support undergraduate research.

Dr. Alexander L. Wijesinha is an Associate professor in the department of Computer and Information Sciences at Towson University. His research interests are in Wireless Networks, Network Security, and Network Protocols. He is actively involved with bare machine computing research in reference to networking protocols, security algorithms, VoIP, and analytical modeling. He has supervised several doctoral students at Towson University. He has a Ph.D. in Computer Science and also a Ph.D. in Mathematics.

Refreshments will be served!