CYBERBLITZ

A Bulltin From Department Of Computer sceince & Engineering

Volume I NOV - 2008

Columns

Message from the Director:

It gives me immense satisfaction to know that the Department of Computer Sciences & Engineering is launching its own departmental bulletin "CYBERBLITZ".I am confident that it will play a pivotal role in developing the research aptitude and technical excellence of everyone in the RASET family, and will also provide a platform for the dissemination of creative ideas. I wish this endeavor all success. May god bless you all.

Message from the Principal:

I am sure that the bulletin will give an opportunity for the students of CSE department experience in publishing technical material. I wish success for this worthy effort.

Message from the HOD:

I take this opportunity to express my pleasure in informing you that the Department Of Computer Science and Engineering is releasing its first Bulletin "CYBERBLITZ" in NOV 2008. This has been a joint venture of both the student community and staff and has helped in exploring and pulling out the hidden technical talents in them. I am sure that this attempt will help us in getting ourselves into a technological and research altitude.

We look forward to go on with this effort in the imminent years and to make it an article worth appreciating.

Rajagiri School of Engineering & Technology(RASET)
Rajagiri Valley P.O.,Kochi, Kerala, India 682 039
Phone: +91 484 2427835, 2428238 Fax: +91 484 2426241
Email: office@rajagiritech.ac.in Website: www.rajagiritech.ac.in

TEACHERS CORNER

IMAGE MANIPULATION USING VERGE POINTS

Gopika S
Lecturer
Department of Computer Science and Engineering
RASET , kakkanad

Verge points are those points having high curvature, which describes the image completely. This representation offers a compact and reversible way to preserve the essence of the original image. This representation is inspired by emulating an image surface with a rubber cloth stretched by 3-D pipes. The extractions of the verge points are done by curvature estimation and then by selecting high curvature points from geometry to extract verge points from image surfaces, image reconstruction can be easily achieved via iterative linear interpolation. Progressive representation is also developed based on a multiscale extraction scheme. Some potential applications such as compression, edge detection, image enhancement, and image editing are then presented to demonstrate the versatility of this representation

IMPLEMENTING A SYSTEM CALL IN LINUX

Unnikrishnan
Lecturer
Department of Computer Science and Engineering
RASET , kakkanad

WHAT IS A SYSTEM CALL

System calls are explicit request to the kernel made via software interrupts. Linux has two modes of operation, kernel mode and user mode. system call is different from library functions provided by compilers.

ADDING OUR OWN SYSTEM CALL

We will add a system call named myroutine.The kernel will have a limit on the number of system calls. The 2.4 kernel have to set it to 271 and the 2.6 kernel have set it to 274.The previous versions may have a value less than these values. .The value is stored in the "__NR_syscalls" variable.You can find what are the system call numbers in use by looking at the entry.S file or unistd.h file. Entry.s file will have entries like

*.long SYMBOL_NAME(sys_unlink) /*10*/*
and the unistd.h file will have entries like
"#define __NR_unlink 10"

Meaning that the system call with name unlink have a system call number 10. You can add the system call in two ways. One is to use an unassigned number as our system call number. Other method is to increment the system call number limit by one and use that new limit as our number.

TEACHERS CORNER

DEFINING A SYSTEM CALL

We will follow systematic way. That is we will have a header file and a source file. Let the header file be named myroutine.h in which you will include linkage.h header file provided by linux kernel. Also place the header file in linux directory in the kernel include directory. You can place it any where,what you have to do is to include that header file in the source file. The linkage.h file will resolve the “asmlinkage” keyword to be used in the system call source file.

Now you have to define the system call.You have to add the keyword asmlinkage for the system call.Let it be defined in myroutine.c file.Then that file will look like this.

```
#include <linux/myroutine.h>
asmlinkage int sys_myroutine(int arg1,int arg2){
return arg1+arg2;
}
```

You have to compiler the myroutine.c to object file and link it to the kernel. Then the system call will work.Souce code for the system call is available in http://www.imada.sdu.dk/~svalle/courses/dm14-2005/mirror/linux-gazette-9828-howto.html

Reference:

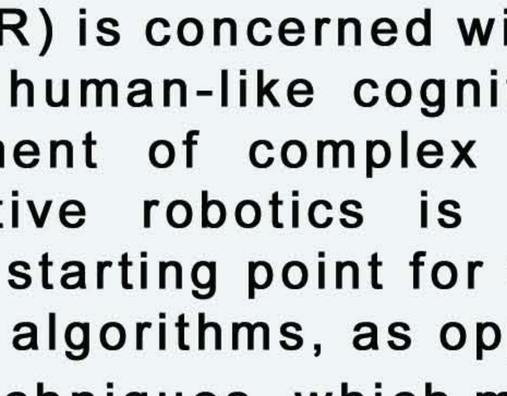
Understanding the linux kernel by Daniel P Bovet

COGNITIVE ROBOTICS

Vinodh P.Vijayan
Lecturer
Department of Computer Science and Engineering
RASET , kakkanad

Cognitive robotics (CR) is concerned with endowing robots with mammalian and human-like cognitive capabilities to enable the achievement of complex goals in complex environments. Cognitive robotics is focused on using animal cognition as a starting point for the development of robotic computational algorithms, as opposed to more traditional Artificial Intelligence AI techniques, which may or may not draw upon mammalian and human cognition as an inspiration for algorithm development. Robotic cognitive capabilities include perception processing, attention allocation, anticipation, planning, reasoning about other agents, and perhaps reasoning about their own mental states. Robotic cognition embodies the behaviour of intelligent agents in the physical world (or a virtual world, in the case of simulated CR).

A cognitive robot should exhibit:



- knowledge
- beliefs
- preferences
- goals
- informational attitudes
- motivational attitudes (observing, communicating, revising beliefs, planning)

Reference:

Humanoid Robotics Group
MIT Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139 USA

STUDENTS CORNER

INTRODUCTION TO GRAPHICAL MODELS AND BAYESIAN NETWORKS

prepared by

Deepti John
S7 CSE
RASET

Graphical models (GM) combine probability theory and graph theory. They provide a tool for dealing with the problems of uncertainty and complexity. Probabilistic graphical models are graphs in which nodes represent random variables. Arcs, or the lack of arcs, represent conditional independence assumptions. Therefore they provide a compact representation of joint probability distributions. There are two types of graphical models: Undirected graphical models and directed graphical models. Undirected graphical models, also called Markov Random Fields (MRFs) or Markov networks, have a simple definition of independence: two (sets of) nodes A and B are conditionally independent given a third set, C, if all paths between the nodes in A and B are separated by a node in C. By contrast, directed graphical models also called Bayesian Networks or Belief Networks (BNs), have a more complicated notion of independence, which takes into account the directionality of the arcs

Reference:

[1]. Matthias O. Franz, Carl Edward Rasmussen, 2006, “Introduction to Graphical Models”, http://www2.tuebingen.mpg.de/agbs/usl/wiki/lect09.pdf.

[2] “A Brief Introduction to Graphical Models and Bayesian Network Repersentation”, http://www.cs.ubc.ca/~murphyk/Bayes/bayes_tutorial.pdf

TELEPORTATION

A Concept or a Reality

prepared by

Cyrin Shelly
S7 CSE
RASET

Ever since the wheel was invented more than 5,000 years ago, people have been inventing new ways to travel faster from one point to another. The chariot, bicycle, automobile, airplane and rocket have all been invented to decrease the amount of time we spend getting to our desired destinations. Yet each of these forms of transportation shares the same flaw: They require us to cross a physical distance, which can take anywhere, from minutes to many hours depending on the starting and ending points.

Teleportation involves de-materializing an object at one point, and sending the details of that object's precise atomic configuration to another location, where it will be reconstructed. What this means is that time and space could be eliminated from travel -- we could be transported to any location instantly, without actually crossing a physical distance.

Many of us were introduced to the idea of teleportation, and other futuristic technologies, by the short-lived Star Trek television series (1966-69) based on tales written by Gene Roddenberry

Viewers watched in amazement as Captain Kirk, Spock, Dr. McCoy and others beamed down to the planets they encountered on their journeys through the universe.

In 1993, the idea of teleportation moved out of the realm of science fiction and into the world of theoretical possibility. It was then that physicist Charles Bennett and a team of researchers at IBM confirmed that quantum teleportation was possible, but only if the original object being teleported was destroyed. This revelation, first announced by Bennett at an annual meeting of the American Physical Society in March 1993, was followed by a report on his findings in the March 29, 1993 issue of Physical Review Letters. Since that time, experiments using photons have proven that quantum teleportation is in fact possible.

Reference:

1. Teleportation!: A Practical Guide for the Metaphysical Traveler : Gwen Totterdale, Jessica Severn, Jessica June Hattie Severn ,Words of Wizdom International, 1996.

MODELING HUMAN-ROBOT INTERACTION with GOMS

prepared by

Eldhose Rajan & Haroon Ashraf,
S7 CSE
RASET

The Goals, Operators, Methods, and Selection rules (GOMS) method is a well-established means of modeling the procedures that humans use to interact with technology. The focus on this paper is on two questions: what is different about using GOMS for human-robot interaction (HRI) versus using GOMS for traditional computer applications, and what are promising approaches for using GOMS to evaluate competing HRI designs? This paper raises issues in using GOMS for modeling HRI and illustrates them with GOMS models that compare two interfaces for urban search-and-rescue robots. Very little work has been done with GOMS so far in the HRI domain, so one of our chief contributions is the insight we introduce for using GOMS for HRI. It is one among the most likely and interesting topics likely to surface in near future.

Reference:

1. The Psychology of Human-Computer Interaction. Hillsdale, New Jersey: Erlbaum.
Card, S., Moran, T., and Newell, A. (1983)

STUDENTS CORNER

ARTIFICIAL IMMUNE SYSTEMS AS A NOVEL SOFT COMPUTING PARADIGM

prepared by

Naveeth Ravindran
S7CSE
RASET

Artificial immune systems (AIS) can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. Their development and application domains follow those of soft computing paradigms such as artificial neural networks (ANN), evolutionary algorithms (EA) and fuzzy systems (FS). AIS have been applied to a wide variety of domain areas, such as pattern recognition and classification, optimization, data analysis, computer security and robotics. The increase in the understanding of several computational intelligence approaches, such as artificial neural networks, fuzzy systems, evolutionary algorithms and probabilistic reasoning (PR), led to the proposal of the soft computing (SC) paradigm. SC is usually viewed as a fusion of these approaches, which in turn provides foundations for the conception, design and development of computational intelligence systems. By combining or hybridizing such paradigms, it has been possible to create a number of successful and sophisticated solutions to complex real-world problems.

The basis of the framework to design artificial immune systems consists of: a representation to create abstract models of immune organs, cells and molecules; a set of functions, termed affinity functions, to quantify the interactions of these ‘‘artificial elements’’, and a set of general purpose algorithms to govern the dynamics of the AIS. The framework reviews and expands a scheme to create abstract models of the immune cells and molecules. Given the framework, the design of an AIS is straightforward.

1. Choose suitable shape-spaces and affinity measures. These are usually problem dependent or determined according to some heuristics.
2. Apply any of the algorithms to determine how the system is going to behave over time.

Artificial neural networks (ANN), evolutionary algorithms (EA) and fuzzy systems (FS) have a great potential to interact with artificial immune systems. Immune network models have been used as novel approaches for the development and improvement of neural network models and vice-versa. A great number of the AIS currently developed can be characterized as having an adaptation akin to an evolutionary algorithm. Artificial immune systems thus constitute a novel computational intelligence paradigm inspired by the immune system. AIS have also been used in conjunction with other soft computing paradigms in order to create more powerful models and improve individual performances, supporting the claim that they compose a new and very useful soft computing approach.

Reference:

1. Leandro N. De Castro, Jonathan, “Artificial Immune Systems”, http://books.google.com/books?id=aMFP7p8DtaQC&printsec=frontcover&dq=Artificial+immune+systems&lr=&sig=ACfU3U0Im1byvDOJVIOVIbu6tOvcxRGmmw