

# Proposed Booting Screen and Architecture in Regional Language for Linux based Mobile Devices

Milan S. Bhatt  
Programmer  
M. D. Gramseva Mahavidyalaya  
Gujarat Vidyapith, Sadra

Prashant M. Dolia, Ph. D  
Associate Professor & Supervisor  
Maharaja Krishnakumar Sinhji Bhavnagar University,  
Bhavnagar

## ABSTRACT

Specific domain people are working together with the mobile technology in different way. But now a days many of the mobile operating systems only in English Language so, ruler area people can't understand English language properly, So new booting process and its component in regional language as well as all the applications in regional language, entire operating systems and its process (different messages) in regional language.

## KEYWORDS

Regional Language, Linux, Mobile Devices, WYSIWYG, Language, Language Processing, Booting.

## 1. INTRODUCTION

The Information Technology, without doubt, has enormous power to improve how people live and work. Thousands of tools – hardware, software, and embedded – are developed to make life of mankind an efficient and convenient. A revolution is taking place today in the way of people, how to access, learn, and interact with information.

Researcher has also concentrated on the language processing and scheduling of the system with platform for Mobile Devices with Linux Kernel, This operating system concentrated Process scheduling and CPU scheduling on Linux based mobile with regional language.

Therefore, further research work is required to develop more assistive system in areas like **what you see, is what you get (WYSIWYG) concept in the system, fast processing speed, as well as language processing with regional language.** In this research work, i.e. a new booting process in regional language as well as all the applications in regional language, entire operating systems and its process (different messages) in regional language.

The possible motives for doing research may be either one or more of the following.

Desire to get a research degree along with its consequential benefits;

For this research work to find out and solve the understanding of English language to regional language in Mobile Devices.

Desire to face the challenge in solving the unsolved problems, i.e., concern over practical problems initiates' research;

During this solution, many problems are there like: now a days in mobile devices already have regional language

application are already installed so this is the unsolved problem for an operating system in regional languages as well as no any booting process for mobile devices in regional language.

Desire to get intellectual joy of doing some creative work;

In this research work, try to make creative work i.e. a new booting process in regional language as well as all the applications in regional language, entire operating systems and its process (different messages) in regional language.

Desire to be of service to society;

Ultimate objective of technology is to facilitate human being. If same thought is viewed for specific domains (replacing technology with information technology and human being with use of the mobile), then objective would be to facilitate Indian people used the Linux based mobile and its system is working on the concept of the what you see is what you get in regional language.

## 2. DESCRIPTIVE RESEARCH

How many users are used to with Linux based mobile?

In current scenario there are only 0.92 % users are used to with Linux based mobile. (Not Android Phone)

How many user are used to with regional language in Mobile devices?

Today approximately 2 % users are used to with regional language applications in mobile devices.

Price of Linux based mobile phones;

Approximately 25,000 Rs. to 50,000 Rs. i.e. More costly of Android, Windows based and Symbian OS mobile. But controversy is that Linux Operating Systems are totally free.

How many mobile company used booting process in regional language?

No any regional languages are used to in mobile for booting process.

## 3. LANGUAGE PROCESSING FOR BOOTING PROCESS

Language Processing is a field of Computer Science and Linguistics concerned with the interactions between computers and natural (human) languages. And there are four main categories.

## PHONOLOGY

- Modelling the pronunciation of a word as a string of symbols – PHONES.
- Involves study of syllables, how they sound and how they group together to produce the word sound.
- Notice the difference in sound of “E” in the following words:

*Ear            End            Elephant*

## MORPHOLOGY

- Identification, analysis and description of the structure of words.
- Study of structural variations of words.
- **INFLECTIONS** in a word are structural changes, usually through affixes, to express Number, Tense, Case, Gender, Person, etc.:

*dog – dogs            goose – geese*

*hunt – hunted        his – hers*

- **WORD FORMATIONS** includes a group of words that have a specific meaning when they appear together:

*sister in law        future plan*

## SYNTAX

- Study of grammars.
- Syntactic correctness in sentence formation.
- Part of Speech tagging:

*Noun    Verb    Adjective*

## SEMANTICS

- Understanding and representing the meaning.
- Predicate Calculus can be used to represent semantic.

## 4. BOOTING PROCESS NEED LANGUAGE PROCESSOR

Receives a textual representation of an algorithm in a source language, and produces as output a representation of the same algorithm in the object or target language.

**Assemblers:** Language processors that map low-level language instructions into machine code, e.g. the ARM assembler.

**Compilers:** Language processors that map high-level language instructions into machine code, e.g. Delphi, GCC, Visual C++ etc.

**Pre-processors:** Language processors that map a superset of high-level language into the original high-level language, or perform simple text substitutions before translation takes place.

**Interpreters:** Language processors that include an execution component, i.e. they perform the operations specified in the source text, rather than re-expressing them in another language; e.g. Matlab

**Disassemblers:** language processors that attempt to take object code at a low level and regenerate source code at a higher level.

## 5. LANGUAGE TRANSLATOR

A language translator is a computer program that converts a program written in a procedural language such as BASIC into machine language that can be directly executed by the computer. Computers can execute only machine language programs. Programs written in any other language must be translated into a machine language load module, which is suitable for loading directly into primary storage.

### Program translation

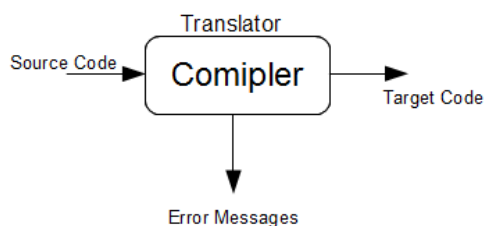
The program translation model bridges the execution gap by translating a program written in a PL, called the *source program* (SP), into an equivalent program in the machine or assembly language of the computer system, called the *target program* (TP).

A program must be translated before it can be executed.

- The translated program may be saved in a file. The saved program may be executed repeatedly.
- A program must be retranslated following modifications.

Language Processing = Analysis of Source Programme + Synthesis of Target Programme

Refer to the collection of language processor components engaged in analysing a source program as the analysis phase of the language processor. Components engaged in synthesizing a target program constitute the synthesis phase.



[Fig 1: Language Processing]

**Lexical Analysis (Scanning)**

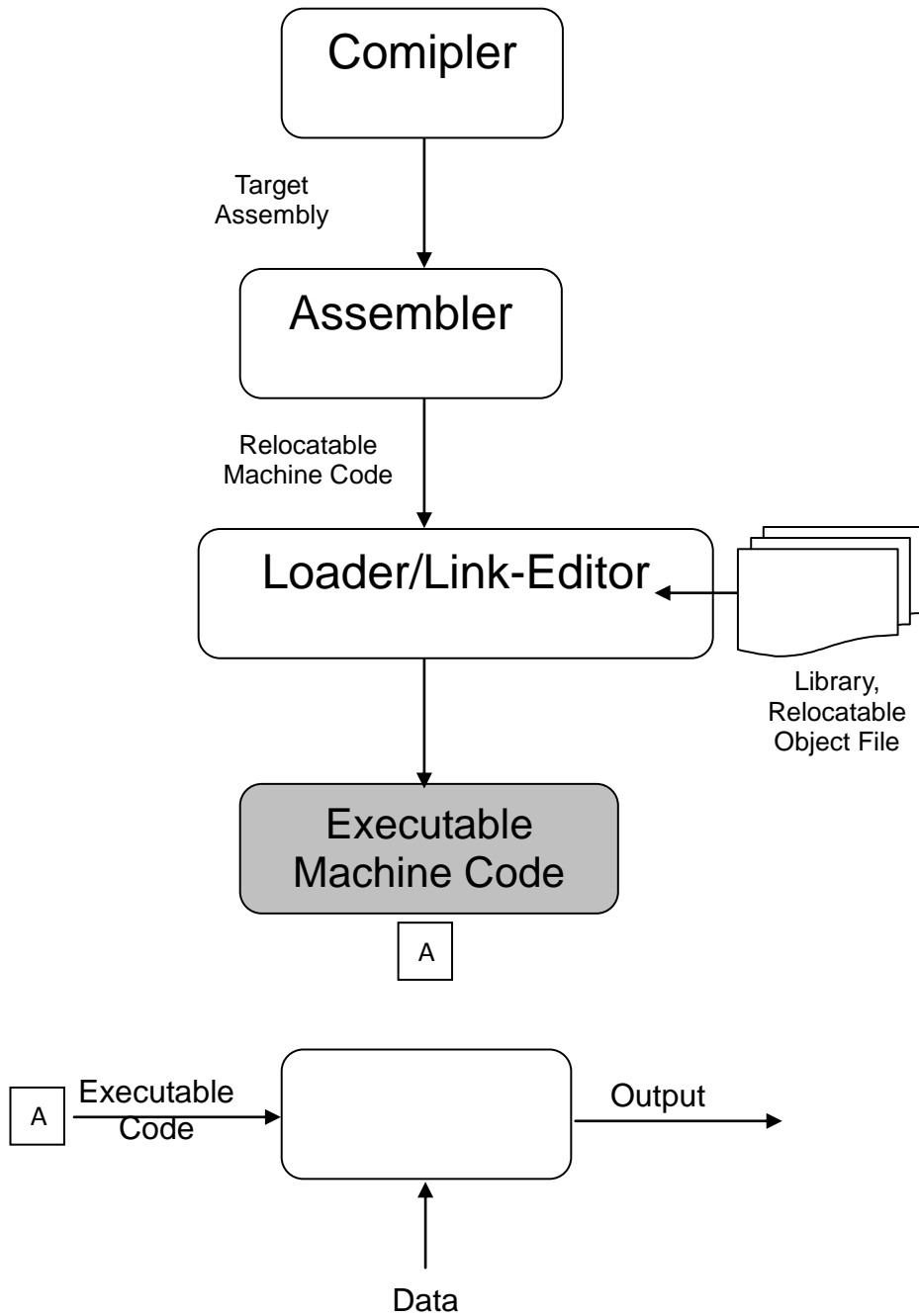
Lexical analysis builds a descriptor, called a *token*, for each lexical unit. A token contains two fields—*class code*, and *number in class*, *class code* identifies the class to which a lexical unit belongs, *number in class* is the entry number of the lexical unit in the relevant table.

**Syntax Analysis (Parsing)**

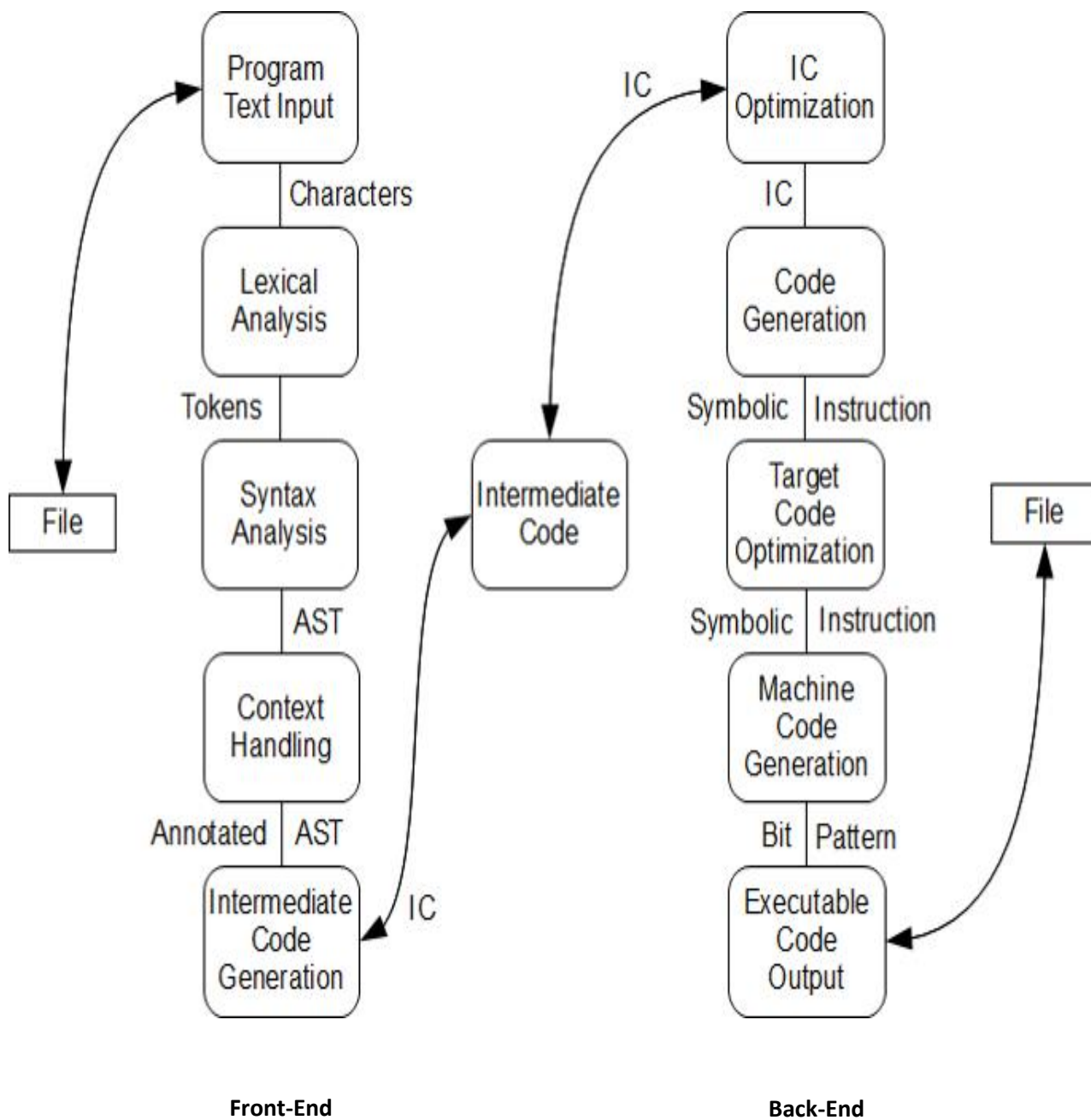
Syntax analysis processes the string of tokens built by lexical analysis to determine the statement class, e.g. assignment statement, if statement, etc.

**Semantic analysis**

It adds information to a table or adds an action to the sequence. It then modifies the IC to enable further semantic analysis. The analysis ends when the tree has been completely processed.



[Fig 2: Language Processing Structure]



[Fig 3: Conceptual Structure of Language Processing]

## 6. GUJARATI MORPHOLOGY

Gujarati has three genders (masculine, neuter and feminine), two numbers (singular and plural) and three cases (nominative, oblique/vocative and locative) for nouns. The gender of a noun is determined either by its meaning or by its termination. The nouns get inflected on the basis of the word ending, number and case. The Gujarati adjectives are of two types – declinable and indeclinable. The declinable adjectives have the termination –ū in neuter absolute. The masculine absolute of these adjectives ends in -o (૦) and the feminine absolute in -ī (ી). For example, the adjective

(sārū - good) takes the form સારૂ (sārū), સારો (sāro) and સારી (sārī) when used for a neuter, masculine and feminine object respectively. These adjectives agree with the noun they qualify in gender, number and case. The adjectives that do not end in -ū in neuter absolute singular are classified as indeclinable and remain unaltered when affixed to a noun. The Gujarati verbs are inflected based upon a combination of gender, number, person, aspect, tense and mood. There are several postpositions in Gujarati which get bound to the

nouns or verbs which they postposition. e.g. -nū (નું : genitive marker), -mā (માં : in), -e (ે: ergative marker), etc. These postpositions get agglutinated to the nouns or verbs and not merely follow them.

## 7. RELAN ALGORITHM

**STEP 1:** Obtain the optimal split position for each word in the word list provided for training using equation and the list of hand crafted suffixes.

**STEP 2:** Repeat Step 1 until the optimal split positions of all the words remain unchanged.

**STEP 3:** Generate signatures using the stems and suffixes generated from the training phase.

**STEP 4:** Discard the signatures which contain either only one stem or only one suffix.

During the *relan* phase, to obtain the optimal split position for each word present in the Gujarati word list provided for training. Obtain the optimal split for any word by taking all possible splits of the word and choosing the split which maximizes the function given in equation as the optimal split position. The suffix corresponding to the optimal split position is verified against the list of 59 Gujarati suffixes created by us. If it cannot be generated by agglutination of the hand crafted suffixes, then the length of the word is chosen as the optimal split position. I.e. the entire word is treated as a stem with no suffix.

{stem<sub>1</sub> + suffix<sub>1</sub>, stem<sub>2</sub> + suffix<sub>2</sub>, stem<sub>3</sub> + suffix<sub>3</sub>...stem<sub>L</sub> + suffix<sub>L</sub>}

ધરન = { ધ + રન, ધર + ન, ધરન + ા, ધરન + NULL }

સપનું = { સ + પનું, સપ + નું, સપન + ું, સપનું + NULL }

$$\text{Equation: } f(i) = i * \log(\text{freq}(\text{stem}_i)) + (L - i) * \log(\text{freq}(\text{suffix}_i))$$

i : Split position (Varies from 1 to L)

L : Length of the Word

## Applying regional Language in Booting Process

In future the change of the booting screen in regional language. This future work done by the configuration file of Linux.

The **bootloader** is the first software program that runs when a computer starts. It is responsible for loading and transferring control to the Linux kernel. The kernel, in turn, initializes the rest of the operating system.

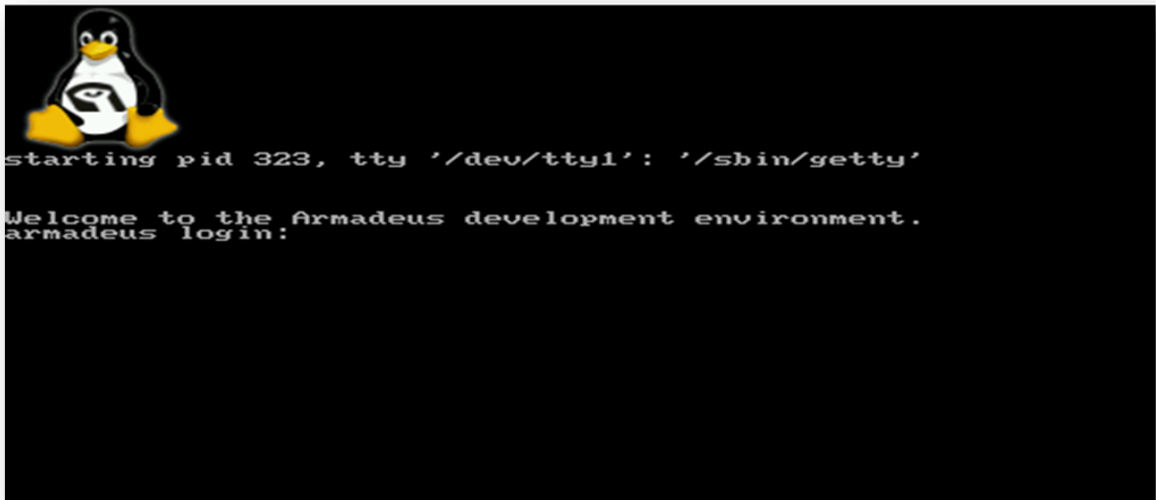
### SYSLINUX

SYSLINUX is a boot loader for the Linux operating system which runs on an MS-DOS/Windows FAT file system. It is intended to simplify first-time installation of Linux, and for creation of rescue and other special purpose boot disks. All the configurable defaults in SYSLINUX can be changed by creating a file called **syslinux.cfg**.

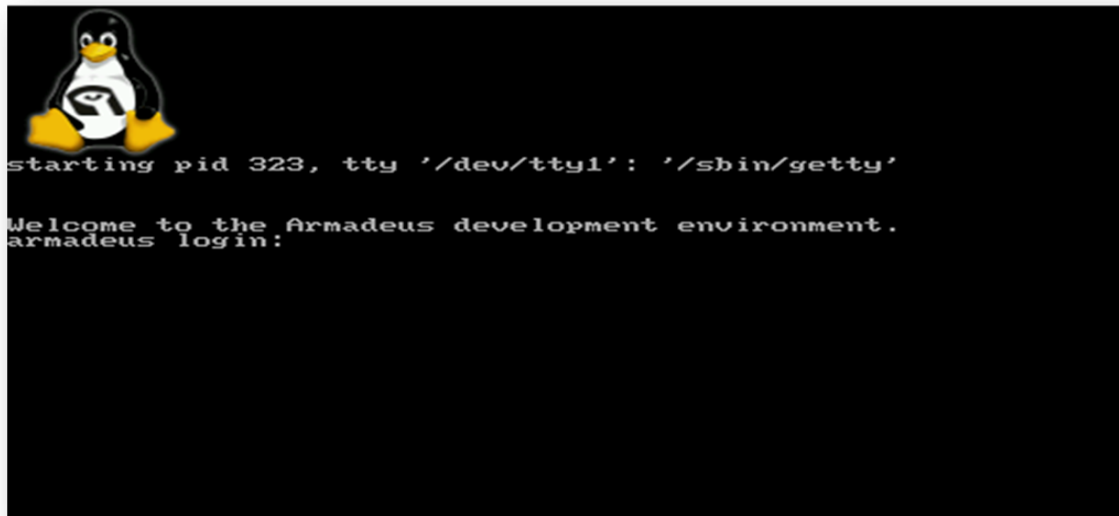
```
/boot/syslinux/syslinux.cfg
/syslinux/syslinux.cfg
/syslinux.cfg
```

Now the append own message to the booting process with modification of **syslinux.cfg**.

```
LABEL mylabel
KERNEL mykernel
APPEND mylabel = "રુલુલુલુ"
```



[Fig 4: Default Booting Screen]



[Fig 5: Proposed Booting Screen in Regional Language]

## 8. CONCLUSION

In future the new booting process and its component in regional language are very helpful to the regional people and easy to understanding the applications, entire operating systems and its process (different messages) in regional language. Also very awareness of the entire process and its messages for the mobile technology in their regional language for the regional people. And in future specific domain people are working together with the mobile technology in different way like in regional language.

## 9. REFERENCES

- [1] Zhou Jin, Demsky Brian, Memory Management for Many-Core Processors with Software Configurable Locality Policies, [International Symposium on Memory Management, 2012].
- [2] Deshpande Vivek, Wadhai Vijay, Helonde J B, Characterizing the Memory Management for Improving the Performance of Embedded system used in Wireless Sensor Networks, [IJCA Proceedings on International Conference in Computational Intelligence, 2012].
- [3] Gitzenis Savvas, Bambos Nicholas, Member, IEEE, “Joint Transmitter Power Control and Mobile Cache Management in Wireless Computing”, IEEE Transactions on Mobile Computing, Vol. 7, No. 4, April 2008.
- [4] Galal Mohamed, Khaled Elsayed, Chief Technology Officer, Software Specialist, “How to build memory management for wired/wireless nets (Part 1)”, Embedded Wireless Business Unit SySDSoft, eetindia.com, EE Times-India.
- [5] Liu Danyi, Lin Ching, Mobile embedded database for remote process management system, [Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian].
- [6] Yu Chang-Hyo, Kim Lee-Sup, Kim Donghyun, an Energy-Efficient Mobile Vertex Processor with Multithread Expanded VLIW Architecture and Vertex Caches, [IEEE Journal of Solid-State Circuits].
- [7] F Hausknecht, R Mzyk, G Dehm-Andone, U Trautwein, G Fischer, R. Weigel, Efficient firmware implementation of a multichannel scanning receiver for surveillance applications, [9th International Multi-Conference on Systems, Signals and Devices (SSD), 2012].
- [8] Xiang Guofu, Jin Hai, Zou Deqing, A comprehensive monitoring framework for virtual computing environment, [International Conference on Information Networking (ICOIN), 2012].
- [9] Li Ning, Y Kinebuchi, H Mitake, H Shimada, Lin Tsung-Han, Nakajima, A Light-Weighted Virtualization Layer for Multicore Processor-Based Rich Functional Embedded Systems, [IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2012].
- [10] Huan, Shiyu, Design and implementation of an embedded audio-on-demand system, [International Conference on Systems and Informatics (ICSAI), 2012].
- [11] Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), pages 51–55, the 23rd International Conference on Computational Linguistics (COLING), Beijing, August 2010.
- [12] Creutz, Mathis, and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. Association for Computing Machinery Transactions on Speech and Language Processing.