Regular Paper
Journal of Computing Science and Engineering
Vol.00, No.0, Jan 0000, pp. 000-000

An Example Usage of the JCSE LaTeX Class

Debra Park
School of Electrical Engineering and Computer Science, ABC National University, Seoul, Korea **debbie@abc.ac.kr**

Dezhen Zhu*
School of Electrical Engineering and Computer Science, ABC National University, Seoul, Korea **jdj@abc.ac.kr**

Abstract
This is an example tex file which demonstrates the functionalities of JCSE LaTeXtemplate. Lengthening the abstract for testing purpose: Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah blah blah blah blah blah. Gigabit switches must work. After years of key research into checksums, we confirm the simulation of cache coherence, which embodies the confusing principles of cryptoanalysis. GulyFlea, our new methodology for read-write information, is the solution to all of these grand challenges.

Category: Embedded Computing

Keywords: Gigabit switches; gulyFlea; cryptoanalysis; systems; internet technologies; queueing systems; keyword1; keyword2; keyword3; keyword4

I. INTRODUCTION

This is an exmple LaTeX file demonstrating the usage of the JCSE LaTeX class file. Skip to section II for example tables, figures, etc. Other than that, the rest of the paper is gibberish generated by SCIgen [1]. LaTeX codes are worth looking at, though.

The analysis of the transistor has investigated IPv6, and current trends suggest that the emulation of expert systems will soon emerge[1]. But, despite the fact that conventional wisdom states that this question is mostly fixed by the emulation of Web services, we believe that a different approach is necessary. In fact, few system administrators would disagree with the construction of the transistor, which embodies the private principles of cryptoanalysis [2]. To what extent can IPv7 be constructed to fix this obstacle?

We concentrate our efforts on showing that the foremost low-energy algorithm for the understanding of write-ahead logging by Sun [3] is maximally efficient [4]. The basic tenet of this approach is the development of simulated annealing. Existing wireless and "smart" systems use 802.11b to cache systems. It should be noted that our heuristic is derived from the deployment of Boolean logic. Even though this is largely an unfortunate mission, it fell in line with our expectations. On a similar note, indeed, the UNIVAC[2] computer and Lamport clocks have a long history of interfering in this manner. As a result, GulyFlea turns the atomic archetypes sledgehammer into a scalpel.

Our contributions are as follows. We better understand how scatter/gather I/O can be applied to the exploration of semaphores. Second, we argue that red-black trees can be made game-theoretic,

[1] Foot note test 1

[2] Foot note test2

Open Access yy.5626/JCSE.2011.5.2.xxx http://jcse.kiise.org

Received 00 Month 2011, Accepted 00 Month 2011, Revised 00 Month 2011

* Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

Bayesian, and classical.

The roadmap of the paper is as follows. To begin with, we motivate the need for the Internet. We place our work in context with the related work in this area. We confirm the exploration of RAID. Next, we confirm the construction of courseware. Finally, we conclude.

II. EXAMPLES

A. One Equation

$$F = -\frac{GMm}{r^2} \tag{1}$$

Equations can be easily written in equation environment. See Equation 1 for example.

B. Series of Equations

$$\begin{aligned} y &= x^4 + 4 \\ &= (x^2+2)^2 - 4x^2 \\ &\leq (x^2+2)^2 \end{aligned} \tag{2}$$

Sometimes a series of equations are needed. See Equation 2 for details[3].

C. Algorithms

Algorithm 1 Euclids algorithm

```
1: procedure EUCLID(a, b) ▷ The g.c.d. of a and b
2:     r ← a mod b
3:     while r ≠ 0 do▷ We have the answer if r is 0
4:         a ← b
5:         b ← r
6:         r ← a mod b
7:     end while
8:     return b                          ▷ The gcd is b
9: end procedure
```

The use of "algorithmicx" package is recommended, although it is possible show algorithms manually. An example is shown as Algorithm 1.

D. Theorems, Proofs, etc.

JCSE.cls profides many environments for theorems, proofs, etc, Definitions 1 and 2 are an example. There are more environments:

DEFINITION 1 (A triangle). *A plane figure bounded by 3 straight sides is a triangle.*

DEFINITION 2 (a genus (or family)). *An existing definition that serves as a portion of the new definition; all definitions with the same genus are considered members of that genus, and a definition can be composed of multiple genera (more than one genus).*

[3] `http://www.personal.ceu.hu/tex/cookbook.html`

LEMMA 1 (Example Lemma). *Lemma lemma lemma lemma lemma lemma lemma lemma lemma.*

Proof. Write proof here. □

COROLLARY 1 (Example corollary). *Corollary corollary corollary corollary corollary.*

PROPOSITION 1 (Example proposition). *All humans are mortal.*

AXIOM 1 (Example Axiom). *It is possible to draw a straight line from any point to any other point.*

REMARK 1 (Example Remark). Can't prove axioms.

EXAMPLE 1 (Example example). There can be numbered examples.

THEOREM 1 (Fermat's Last Theorem). *No three positive integers a, b, and c can satisfy the equation $a^n + b^n = c^n$ for any integer value of n greater than two.*

Proof. I have discovered a truly remarkable proof which this margin is too small to contain. □

E. Subsection testing1

Chicken chicken chicken chicken, chicken chicken chicken chicken [5].

1) Subsubsection Testing1:

Subsubsection headings are enumerated by Arabic numerals followed by parentheses. They are indented, italicized, upper and lower case, run into the text in their sections, and are followed by a colon.

2) Subsubsection Referencing:

It is possible to reference sections and subsections. Subsubsection II-E.1 has discussion about chickens. Subsection II-D has examples of definitions, remarks, proofs, etc. Section I is the introduction section.

3) Subsubsection Image that Spans Two Columns:

Fig. 1 is from `http://en.wikipedia.org/wiki/File:Unix_history-simple.svg`.

F. Subsection testing2

Chicken chicken chicken chicken, chicken chicken chicken chicken.

G. Example Tables

1) Simple Tables:

"thline" command provides thicker line for drawing tables, as shown as Table I.

2) (Optional) Booktabs Package:

Table II shows an example of larger table which spans two columns. "booktabs" package

Fig. 1. Evolution of Unix and Unix-like systems. This is an example of large figures which spans two columns.

Table II. Comparison of results by their algorithm [5] and our algorithm.

	Theirs [5]				Ours			
Circuit	Skew (ps)	Hops	Latency (ps)	Error(%)	Skew (ps)	Hops	Time (ps)	Error(%)
s5378	40	3004	19.4	14.2	14.1	2990	12	0.02
s9234	50	2345	23.6	18.3	18.2	2131	16	0.02
s13207	134	2263	54.8	47.7	47.8	1950	36	0.19
s15850	133	2615	56.7	51.5	52.9	1515	49	0.12
s35932	407	8774	129.7	122.4	111.6	6684	121	1.26
s38417	343	7917	113.7	100.1	112.8	7917	81	1.20
s35584	330	268	119.8	113.1	99.8	255	100	0.80
Average	3	4	5	6	7	8	9	10

Table I. Quarks

Name	Symbol	Antiparticle	Charge (e)
up	u	$\overline{u}$	+2/3
down	d	$\overline{d}$	-1/3
charm	c	$\overline{c}$	+2/3
strange	s	$\overline{s}$	1/3
top	t	$\overline{t}$	+2/3
bottom	b	$\overline{b}$	-1/3

Fig. 2. GulyFlea's encrypted location.

allows higher quality tables. Use toprule, midrule, bottomrule, cmidrule commands instead if using booktabs. cmidrule command allows shorter version of "clines" as the two lines between the first and second rows of Table II.

III. MODEL

The properties of our algorithm depend greatly on the assumptions inherent in our framework; in this section, we outline those assumptions. This may or may not actually hold in reality. We estimate that spreadsheets can be made metamorphic, homogeneous, and adaptive. This may or may not actually hold in reality. Rather than creating fiber-optic cables, GulyFlea chooses to request stable theory. Therefore, the framework that GulyFlea uses is unfounded.

Our heuristic relies on the structured architecture outlined in the recent little-known work by J. Dongarra in the field of robotics. This seems to hold in most cases. Rather than learning the emulation of Boolean logic, our framework chooses to request access points. Rather than locating introspective theory, GulyFlea chooses to locate self-learning communication. Even though security experts largely postulate the exact opposite, GulyFlea depends on this property for correct behavior. Along these same lines, rather than evaluating classical methodologies, our heuristic chooses to cache the investigation of the transistor. This seems to hold in most cases. The question is, will GulyFlea satisfy all of these assumptions? Yes, but with low probability [6, 7].

Reality aside, we would like to simulate a model for how GulyFlea might behave in theory. We scripted a trace, over the course of several months, verifying that our framework holds for most cases. Even though physicists always assume the exact opposite, our heuristic depends on this property for correct behavior. We estimate that SMPs [8] and IPv6 can interact to fulfill this aim. See our existing technical report [9] for details.

IV. IMPLEMENTATION

After several weeks of arduous architecting, we finally have a working implementation of our algorithm. Information theorists have complete control over the codebase of 63 Simula-67 files, which of course is necessary so that the little-known reliable algorithm for the refinement of the location-identity split [10] runs in $O(\log \log \log n)$ time. Further, it was necessary to cap the sampling rate used by GulyFlea to 636 man-hours. On a similar note, our methodology is composed of a virtual machine monitor, a centralized logging facility, and a hacked operating system. The collection of shell scripts contains about 58 instructions of Perl. Our algorithm is composed of a hacked operating system, a client-side library, and a centralized logging facility.

V. EVALUATION

As we will soon see, the goals of this section are manifold. Our overall evaluation strategy seeks to prove three hypotheses: (1) that 802.11b has actually shown muted effective popularity of IPv4 over time; (2) that cache coherence no longer influences system design; and finally (3) that an application's effective user-kernel boundary is less important than an application's virtual user-kernel boundary when

Fig. 3. The 10th-percentile response time of our methodology, as a function of latency.

Fig. 4. The expected interrupt rate of our framework, as a function of block size.

maximizing effective complexity. The reason for this is that studies have shown that average instruction rate is roughly 49% higher than we might expect [11]. Our logic follows a new model: performance might cause us to lose sleep only as long as performance takes a back seat to expected complexity. Our work in this regard is a novel contribution, in and of itself.

A. Hardware and Software Configuration

A well-tuned network setup holds the key to an useful performance analysis. We ran a packet-level deployment on our human test subjects to disprove opportunistically reliable models's lack of influence on B. Maruyama's study of SCSI disks in 1993. we removed 300MB of flash-memory from our highly-available testbed to examine epistemologies. Second, we tripled the time since 2001 of our 10-node testbed. We tripled the throughput of CERN's XBox network. It is largely a confirmed aim but fell in line with our expectations. Next, we removed more RAM from Intel's decommissioned Atari 2600s to understand configurations. Next, we tripled the effective hard disk speed of our desktop machines to disprove the provably highly-available nature of electronic technology. Finally, we added 300Gb/s of Ethernet access to our mobile telephones.

GulyFlea does not run on a commodity operating system but instead requires an independently patched version of Minix Version 8.4. we implemented our model checking server in Lisp, augmented with provably independent extensions. All software was compiled using Microsoft developer's studio built on the German toolkit for collectively harnessing Knesis keyboards. Second, we made all of our software is available under a public domain license.

B. Dogfooding Our System

Is it possible to justify having paid little attention to our implementation and experimental setup? It is not. Seizing upon this approximate configuration, we ran four novel experiments: (1) we dogfooded our algorithm on our own desktop machines, paying particular attention to effective flash-memory speed; (2) we ran kernels on 66 nodes spread throughout the millenium network, and compared them against suffix trees running locally; (3) we ran hierarchical

Fig. 5. The mean latency of our system, as a function of throughput.

Fig. 6. The mean energy of our system, as a function of complexity.

Fig. 7. Note that energy grows as instruction rate decreases – a phenomenon worth investigating in its own right.

databases on 15 nodes spread throughout the sensor-net network, and compared them against randomized algorithms running locally; and (4) we measured WHOIS and E-mail throughput on our system.

Now for the climactic analysis of the second half of our experiments. Note the heavy tail on the CDF in Figure 5, exhibiting degraded latency. The key to Figure 3 is closing the feedback loop; Figure 5 shows how our system's effective block size does not converge otherwise. The key to Figure 7 is closing the feedback loop; Figure 6 shows how our system's distance does not converge otherwise.

Shown in Figure 5, the second half of our experiments call attention to our method's 10th-percentile interrupt rate. The data in Figure 5, in particular, proves that four years of hard work were wasted on this project. Second, the curve in Figure 7 should look familiar; it is better known as $F(n) = n$. On a similar note, these expected power observations contrast to those seen in earlier work [12], such as John Hopcroft's seminal treatise on systems and observed effective USB key throughput.

Lastly, we discuss the second half of our experiments. We scarcely anticipated how accurate our results were in this phase of the performance analysis. The data in Figure 5, in particular, proves that four years of hard work were wasted on this project. The data in Figure 5, in particular, proves that four years of hard work were wasted on this project.

VI. CONCLUSION

Our experiences with our system and the emulation of hierarchical databases prove that the well-known pseudorandom algorithm for the exploration of fiber-optic cables by U. Zhou follows a Zipf-like distribution. To fix this quandary for Boolean logic, we introduced an encrypted tool for developing IPv7. We plan to explore more problems related to these issues in future work.

ACKNOWLEDGEMENTS

This research was undertaken as part of the YYYY project and is jointly funded by a XXX Systems and ZZZZ strategic partnership (ABC/U518331/1).

REFERENCES

[1] J. Stribling, M. Krohn, and D. Aguayo, "Scigen - an automatic CS paper generator," http://pdos.csail.mit.edu/scigen/.

[2] R. U. A. Stein, "An example journal article with a looong looooong title," *Transactions on Computing Systems*, vol. 60, no. 1, pp. 11–99, Jan. 2003.

[3] N. Existent and P. Fake, "An example proceedings article on Markov models," in *Proceedings of the Workshop on Pervasive, Adaptive Archetypes*, Aug. 1995, pp. 2–9.

[4] C. A. R. Hoax, J. Dongarra, and M. Nincompoop, "Journal artical example: A case for link-level acknowledgements," *Journal of Replicated Theory*, vol. 73, no. 28, pp. 57–64, Nov. 2005.

[5] D. Zongker, "Chicken chicken chicken: Chicken chicken," *Annals of Improbable Research*, vol. 12, no. 5, pp. 16–21, 2006.

[6] P. Nincompoop, M. Blum, D. Z. Zhu, J. Quinlan, B. U. Li, a. Chandrasekharan, X. Zhao, and K. Nygaard, "A refinement of the memory bus," in *Proceedings of the Conference on Virtual Communication*, May 2005, pp. 30–33.

[7] B. Loll, V. Jacobson, E. Clarke, and P. Bose, "Expert systems no longer considered harmful," in *Proceedings of the Symposium on Trainable, Omniscient Information*, Jun. 1990, pp. 33–36.

[8] Y. Nincompoop, "Architecting digital-to-analog converters and RPCs using MANITU," *Journal of Efficient Methodologies*, vol. 24, no. 6, pp. 1–10, Sep. 2005.

[9] R. S. Gawk, S. Shenker, and D. Estrin, "A refinement of flip-flop gates with *apery*," *Journal of Probabilistic Models*, vol. 89, no. 58, pp. 20–24, Mar. 2000.

[10] A. Yokel and S. Shastri, "Deconstructing write-ahead logging with Sowce," in *Proceedings of the Conference on Pseudorandom and Distributed Theory*, Oct. 2001, pp. 6–10.

[11] F. J. Nincompoop and V. Ramasubramanian, "Cooperative epistemologies for DHTs," in *Proceedings of the USENIX Security Conference*, Jan. 2003, pp. 119–123.

[12] E. Doggerel, "A study on pneumonoultramicroscopicsilicovolcanoconiosis," *Journal of Interactive, Classical Epistemologies*, vol. 74, no. 2, pp. 158–198, Feb. 1999.