Teaching Research of Operating Systems

Yair Wiseman
Computer Science Department
Bar-Ilan University
Ramat-Gan 52900
Israel
http://www.cs.biu.ac.il/~wiseman

Abstract

Almost all the up-to-date operating systems support multicore processors and also distributed systems. The parallel processing is obviously an essential part of any operating system. However, the other portions of the operating systems are still very important. When teaching a course about research in operating systems, many instructors prefer to teach about the parallel processing and it gives the impression that there is nothing novel to research in the other portions of the operating system. In this paper we show that there is a large amount of research about the operating system portions that do not handle the parallel processing and we advocate paying more attention to these researches in the academic courses about research in operating systems.

1. Introduction

An advanced operating systems course is very common in many Computer Science departments all over the world. Obviously, the academic freedom does not dictate the material should be taught in such courses. Most of the courses contain a lot of Distributed Operating Systems stuff and this leads the graduate students to do their research on Distributed Operating Systems. This is obviously legitimate, but we would like to suggest to split the Distributed Operating Systems and the Non-Distributed Operating Systems into two courses. The one-processor machines are still the majority of the computing power far and wide. Therefore, a Non-Distributed Operating Systems course can be beneficial for pushing more graduate students to research this field and to contribute their aptitude.

The Non-Distributed Operating Systems is still a vital and live field. We found 3 main categories that can be taught:

- New and enhanced techniques
- Algorithms improvements
- Statistical studies

We would like to give some subjects in each category and to explain what can be taught in each subject. The suggested course combined of these 3 categories has been taught in Bar-Ilan University during 2004. The course has a web site (containing slides) that can be reached at [1].

2. New and Enhanced Techniques

This category contains the old and known techniques that any Operating System has. The main concepts of the technique are usually taught in the undergraduate Operating System course. In the advanced Operating System course the up to date enhanced techniques are taught.

2.1. Micro-Kernel

Micro-Kernel [2] is a controversial enhancement for the old concept of the operating system kernel. The first Micro-Kernels have been presented in the beginning of the 70's, but there are still a lot of implications and ramifications that are still researched nowadays.

The conception of Micro-Kernel is to implement outside the kernel whatever possible. This automatically raises the discussion in the class what are the advantages and the disadvantages of such an implementation [3,4]. Another important discussion is which sections of the kernel can be taken out and which sections are essential within the Micro-Kernel.

The classic examples of sections that can be taken out are the paging mechanism and the device drivers. Showing the students how this was implemented in some known Micro-Kernels [5,6,7] can let the student think what other part can be taken out the Micro-Kernel and how it can be implemented. The students also have to deal with the question what definitely cannot be taken out of the Micro-Kernel and what is the reason for that.

The opposite approach is to implement inside the kernel whatever possible. This raises the question what definitely should be outside the kernel, or maybe there is no such a component outside the kernel like was suggested in [8]. Such an approach makes the kernel transactions very long. This feature brings up more questions that should be discussed in the class. E.g. can the kernel be interrupted and if it can be, when it will be interrupted and by which interrupts [9].

2.2. Super-Pages

Super-Pages is an enhancement for the well-known paging concept. Super-Pages are larger pages that are pointed by the TLB. The internal memory of modern computers has been significantly increased during the last decade. However, the TLB coverage (i.e. the size of the memory that can be pointed directly by the TLB) has been increased by a much lower factor during the same period [10,11]. Therefore, several new architectures like Itanium, MIPS R4x00, Alpha, SPARC and HP PA RISC support multiple page size of the frames pointed by the TLB. In that way the memory size pointed directly by the TLB is higher and the overhead of the page table access time is reduced. In addition, many modern operating systems support Super-Paging.

This concept brings up several questions to discuss in class. First, when the Operating System should upgrade some base pages into a large Super-Page. This dilemma is even more complicated when the processor supports several sizes of Super-Pages. E.g. the Itanium has 10 sizes of Super-Pages. Another question is where the location of the small pages in the memory should be. One policy can be putting it in a location that when the Operating System decides to upgrade some base pages into a Super-Page, no relocation is needed [12]. Another policy is to put the base page in the first empty location in the memory and to relocate when the Operating System decides to upgrade [13]. One more question is who should handle the relocation, the hardware or the software [14].

Some processors and Operating Systems have addressed these questions [15,16,17]. The course lets the students see what decision the specific processors and Operating Systems have taken and what were their considerations. The students are solicited to express their view and to think whether they can think about better suggestions with better performance.

2.3. Desktop Scheduling

The schedulers of most of the contemporary operating systems are based on the old well-known schedulers that have been used during the years by the traditional Unixes and other popular operating systems. These schedulers do not always perform well with the new and different characteristic of processes that are used by the new desktop machines.

One of the most notable changes is the multi-media processes that were not almost exist on the old machines and very common nowadays [18]. Usually, in the undergraduate operating systems course the students are taught about the common schedulers used by the popular operating systems. In the operating systems graduate course they should be able to criticize the timing of those schedulers [19]. Furthermore, they are expected to suggest methods to improve the effectiveness of the original schedulers.

2.4. Versioning File Systems

On 1995, for $200 you could get a 0.54GB disk, whereas Slackware Linux 2.2 (Basic Applications+X window) was 0.15Gbytes that are 28% of the disk. On 2004, for $200 you can get a 300GB disk, whereas RedHat Linux Advanced Workstation 2.1 (Basic Applications+X window) for the Itanium Processor is 4.2GB that are 1.4% of the disk. This facts lead to the idea that the space pressure on the disk is not high and a portion of the disk can be reserved for backup.

Versioning File Systems are file systems that do not remove the files that have been deleted. The disk retains the blocks of the deleted files and they can be restored if needed. The idea is very old [20] and some versions of VMS have used it a long ago, but some new File Systems have been proposed recently e.g. Elephant [21,22] and Moraine [23] that have different policies for different types of files and a better interface for the user.

The students are asked to think what should be the new deleting timing of the blocks in such a file system [24]. There are some suggestions in the current file systems for this timing. Which suggestion is the best and whether they can think of a better technique that will yield better results. In addition, does the policy offered by the file systems should be selected by the user or automatically by the operating system? There is a dispute between the researchers in this field what is better [25] and the students are asked to take a position.

3. Algorithms Improvements

Many known algorithms are utilized by the operating system. The idea of this section is to show the students how these algorithms can be adapted by the operating system and how to improve the algorithms.

3.1. ARC

On 1946 Von-Neumann suggested a hierarchy of memories. This concept has been accepted by almost all of the hardwares. In such a hierarchy each of the memories has a greater capacity than the preceding but it is less quickly accessible. When there is no more room in the faster memory, the selecting of the "victim" to be taken out of the faster memory has been traditionally done for decades by the LRU algorithm. The LRU is fast and easy for implementation and has been utilized by many operating systems, but can there be a better algorithm?

We suggest to show the students some new techniques that has been suggested like LRU-K [26], 2Q [27], LRFU [28]. The students should be able to criticize the techniques in several parameters:

- The complexity of the algorithm and time overhead.
- How fast the algorithm can identify “stale” block
- How fast the algorithm understands that a block is heavily used and should be in the faster memory.
- The space overhead needed by the algorithm.

There is a new method invented named ARC [29,30,31] that considered to be the best one known today. The students should think whether this is really the best one and how they can improve it.

4. Statistical studies

Some of the research in the field of operating systems is the studying of the different properties of the operating systems. Such studies can help us understand what the common flaws of the operating systems are. In addition, this can lead us to find ways of solving these flaws.

4.1. Operating System Bugs

Operating Systems like many other softwares are not bugs-free. Some studies have been conducted over the years to analyze the bugs on common operating systems like Linux [32,33] and WindowsNT [34]. These studies can help us know how many bugs an ordinary operating system has. Usually Windows has more bugs than Linux. The students have to think what the reason for that is.

Another important question is where most of the bugs emerge. The studies show that the device drivers are usually the buggiest section in the kernel. Many developers, who may understand the devices more than the kernel, usually will write the Device Drivers. In addition only a few users may have a given device; hence it will be less "battle-tested" than the other sections of the kernel.

The students are explained how this information can help them and how they can improve their products [35]. In complex software systems such information is essential [36]. They can know which sections are less trusty. In addition, they understand common reasons for bugs. E.g. Cut-and-Paste in code writing can be harmful sometimes. Bug lifetime is also an interesting parameter that can let the students know what the standards in the operating system market are.

4.2. Benchmarking

Typically, when a product needs to show its performance, it will demonstrate the results by benchmark results. The same usage of benchmarks is done when writing a scientific paper. The one who chooses the benchmark to be used is the author of paper himself. This opens the author a wide spectrum of subjective representation of the paper results [37,38].

The Operating Systems field is not different [39]. There are several common benchmarking standard like SPEC [40], but researchers do not tend to take the standards seriously. Many papers contain just partial results and sometimes even results of just one program out of the benchmark suite [41,42].

This subject is very important for students who are going to be the next generation of the research community. They should study how to test their ideas honestly. Hiding the drawbacks and the limitations will be a bad education for the students and a frank benchmarking is an essential part of a good paper.

5. Conclusions

The undergraduate course of Operating Systems usually focuses on Non-Distributed Operating Systems. However, the graduate Operating Systems course focuses in many universities on Distributed Operating Systems. Many instructors feel that the advanced subjects can be found in the Distributed Operating Systems field and the Non-Distributed Operating Systems has very little to offer. This paper has shown that there are enough topics to teach in the Non-Distributed Operating Systems course; thus the conclusion of this paper is that there should be two advanced courses - one on Non-Distributed Operating Systems and one on Distributed Operating Systems. Ignoring the Non-Distributed Operating Systems is actually ignoring most of the computers in the world which are not distributed.

6. References

[1] Y. Wiseman, The Home Page of Advanced Operating Systems Course, http://www.cs.biu.ac.il/~wiseman/2os, 2004.

[2] T. P. Scheuermann, "Evolution in Microkernel Design", Computer Science Department, University of North Carolina, Chapel Hill, NC, 2002.

[3] J. Liedtke, "Toward Real MicroKernels", Communications of the ACM, Vol. 39(9), September 1996.

[4] J. Liedtke, "On Micro-Kernel Construction", Proceedings of the 15th ACM Symposium on Operating System Principles, ACM, December 1995.

[5] D. Golub, R. Dean, A. Forin and Richard Rashid. "Unix as an Application Program", Proceedings of the USENIX Summer Conference, June 1990.

[6] B. N. Bershad, C. Chambers, S. Eggers, C. Maeda, D. McNamee, P. Pardyak, S. Savage and E. Gun Sirer, "SPIN - An Extensible Microkernel for Application-specific Operating System Services", ACM Operating Systems Review, Vol. 29(1), January 1995.

[7] The L4 MicroKernel, http://www.cse.unsw.edu.au/~disy/L4/

[8] T. Maeda, "Safe Execution of User Programs in Kernel Mode Using Typed Assembly Language", Master Thesis, The Graduate School of The University of Tokyo, February, 2002.

[9] G. Anzinger and N. Gamble, "Design of a Fully Preemptable Linux Kernel", MontaVista Software, September 2000.

[10] J. Navarro. Transparent operating system support for superpages, Ph.D. Thesis, Department of Computer Science, Rice University, April 2004.

[11] J. Navarro, S. Iyer, P. Druschel and A. Cox. Practical, Transparent Operating System Support for Superpages, Fifth Symposium on Operating Systems Design and Implementation (OSDI '02), Boston, MA, December 9-11, 2002.

[12] M. Talluri and M. D. Hill, Surpassing the TLB Performance of Superpages with Less Operating System Support, Sixth International Symposium on Architectural Support for Programming Languages and Operating Systems (ASPLOS), San Jose, California, pp. 171-182, October 4-7, 1994.

[13] T. H. Romer, W. H. Ohllrich, A. R. Karlin, and B. N. Bershad, Reducing TLB and memory overhead using online superpage promotion, In Proceedings of the 22nd International Symposium on Computer Architecture (ISCA), pp. 87-176, Santa Margherita Ligure, Italy, June 1995.

[14] Z. Fang, L. Zhang, J. Carter, S. McKee, and W. Hsieh. Re-evaluating Online Superpage Promotion with Hardware Support. In Proceedings of the Seventh International Symposium on High Performance Computer Architecture, pp. 63-72, January 2001.

[15] I. Subramanian, C. Mather, K. Peterson, and B. Raghunath. Implementation of multiple pagesize support in HP-UX, In Proceedings of the USENIX, New Orleans, Louisiana, June 15-19, 1998.

[16] N. Ganapathy and C. Schimmel, General Purpose Operating System Support for Multiple Page Sizes, In Proceedings of the USENIX, New Orleans, Louisiana, June 15-19, 1998.

[17] S. Winwood, Y. Shuf, H. Franke, Multiple Page Size Support in the Linux Kernel, Ottawa Linux Symposium, Ottawa, Ont, Canada, June 2002.

[18] J. Nieh, J. G. Hanko, J. D. Northcutt, and G. A. Wall, "SVR4 UNIX Scheduler Unacceptable for Multimedia Applications", Proceedings of the Fourth International Workshop on Network and Operating Systems Support for Digital Audio and Video, Lancaster, UK, pp. 35-48, November 1993.

[19] A. Goel, L. Abeni, C. Krasic, J. Snow, and J. Walpole, "Supporting Time-Sensitive Applications on a Commodity OS", Fifth Symposium on Operating Systems Design and Implementation (OSDI '02), Boston, MA, December 9-11, 2002.

[20] M. D. Schroeder, D. K. Gifford and R. M. Needham, "A Caching File System for a Programmer's Workstation", Proceedings of the tenth ACM symposium on Operating systems principles, Orcas Island, Washington, pp. 25-34, 1985.

[21] D. S. Santry, M. J. Feeley, N. C. Hutchinson, A. C. Veitch, R. W. Carton and J. Ofir, "Deciding When to Forget in the Elephant File System", ACM Symposium on Operating System Principles, Kiawah Island Resort, South Carolina, pp. 110-123, December 12-15, 1999.

[22] D. J. Santry, M. J. Feeley, N. C Hutchinson, and A. C. Veitch, "Elephant: The File System That Never Forgets", Proc. Workshop on Hot Topics in Operating Systems, Rio Rico, Arizona, pages 2-7, 1999.

[23] T. Yamamoto, M. Matsushita and K. Inoue, "Accumulative Versioning File System Moraine and Its Application to Metrics Environment MAME", Proceedings of the 8th ACM SIGSOFT international symposium on Foundations of software engineering, pp. 80-87, Shelter Island, San Diego, California, USA, November 6-10, 2000.

[24] C. A. N. Soules, G. R. Goodson, J. D. Strunk, and G. R. Ganger, "Metadata Efficiency in Versioning File Systems," in Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST 2003), pp. 43-58, San Francisco, California, March 31-April 2, 2003.

[25] K. Muniswamy-Reddy, C. P. Wright, A. Himmer and E. Zadok, "A Versatile and User-Oriented Versioning File System", in Proceedings of the Third USENIX Conference on File and Storage Technologies (FAST 2004), pp. 115-128, San Francisco, California, March 31-April 2, 2004.

[26] E. O'Neil, P. O'Neil and G. Weikum, "The LRU-K Page Replacement Algorithm for Database Disk Buffering", Proceedings of SIGMOD `93, Washington, DC, May 1993.

[27] T. Johnson and D. Shasha, "2Q: a low overhead high performance buffer management replacement algorithm", Proceedings of the Twentieth International Conference on Very Large Databases, VLDB' 94, Santiago, Chile, pp. 439-450, September 1994.
[28] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, "LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies," IEEE Trans. Computers, vol. 50, no. 12, pp. 1352-1360, 2001.
[29] N. Megiddo and D. S. Modha, "ARC: A Self-Tuning, Low Overhead Replacement Cache," Proc. of the 2nd USENIX Conference on File and Storage Technologies (FAST'2003), San Francisco, pp. 115-130, March 31 - April 2, 2003.
[30] N. Megiddo and D. S. Modha, "One Up on LRU" ;login: - The Magazine of the USENIX Association, vol. 28, no. 4, pp. 7-11, August 2003.
[31] N. Megiddo and D. S. Modha, "Outperforming LRU with an Adaptive Replacement Cache Algorithm," IEEE Computer, pp. 4-11, April 2004.
[32] A. Chou, J.-F. Yang, B. Chelf, S. Hallem, and D. Engler. "An Empirical Study of Operating Systems Errors", In Proceedings of the 18th ACM Symposium on OS Principles (SOSP), pp. 73-88, Lake Louise, Alta, Canada, October 2001.
[33] D. G. Majors, "An Investigation of the Call Integrity of the Linux System", 14th. IEEE International Symposium on Software, Reliability Engineering ISSRE 2003, Denver, Colorado November 17-20, 2003.
[34] J. Xu, Z. Kalbarczyk, and R. Iyer, "Networked Windows NT System Field Failure Data Analysis," Proc. 1999 Pacific Rim Int'l Symp. Dependable Computing, IEEE CS Press, Los Alamitos, CA, 1999.
[35] M. M. Swift, B. N. Bershad, and H. M. Levy, "Improving the Reliability of Commodity Operating Systems", Proceedings of the 19th ACM Symposium on Operating Systems Principles, Bolton Landing, New York, October 19-22, 2003.
[36] N. E. Fenton and N. Ohlsson, "Quantitative Analysis of Faults and Failures in a Complex Software System," IEEE Transactions on Software Engineering, 26(8), pp. 797-814, August, 2000.
[37] X. Zhang, "Application-Specific Benchmarking", Ph.D. Thesis, Harvard University, May 2001.
[38] M. Seltzer, D. Krinsky, K. Smith, X. Zhang, "The Case for Application-Specific Benchmarking", In Proc. HotOS-VII, pp. 90-95. Rio Rico, AZ, March, 1999.
[39] J. C. Mogul. , "Brittle Metrics in Operating Systems Research", In Proc. HotOS-VII, pp. 90-95. Rio Rico, AZ, March, 1999.
[40] Standard Performance Evaluation Corporation (SPEC), http://www.spec.org
[41] D. Citron, "MisSPECulation: partial and misleading use of spec CPU2000 in computer architecture conferences", Proceedings of the 30th Annual International Symposium on Computer Architecture, pp. 52-59, June 9-11, 2003.
[42] D. Citron, J. Hennessy, D. Patterson, G. Sohi, "The Use and Abuse of SPEC", Proceedings of the 30th Annual International Symposium on Computer Architecture, pp. 73-77, June 9-11, 2003.
[43] Wiseman Y., "Autonomous Vehicles", Encyclopedia of Information Science and Technology, Fifth Edition, Vol. 1, Chapter 1, 2020, pp. 1-11.

[44] Wiseman, Y., (2018), "In an era of autonomous vehicles, rails are obsolete", International Journal of Control and Automation, Vol. 11, No. 2, pp. 151-160.
[45] Wiseman, Y., (2021), "Intelligent Transportation Systems along with the COVID-19 Pandemic will Significantly Change the Transportation Market", The Open Transportation Journal, Vol. 15, No. 1, pp. 11-15.
[46] Wiseman Y., (2018), "Vehicle identification by OCR, RFID and Bluetooth for toll roads", International Journal of Control and Automation, Vol. 11, No. 9, pp. 67-76.
[47] Wiseman Y., (2020), "Conjoint Vehicle License Plate Identification System", The Open Transportation Journal, Vol. 14, No. 1, pp. 164-173.
[48] Wiseman, Y., (2021), "COVID-19 Along with Autonomous Vehicles will Put an End to Rail Systems in Isolated Territories", In IEEE Intelligent Transportation Systems, Vol. 13, No. 3, pp. 6-12, doi: 10.1109/MITS.2021.3049409.
[49] Wiseman, Y. (2019), "Driverless cars will make union stations obsolete", The Open Transportation Journal, Vol. 13, No. 1, pp. 109b 115.
[50] Wiseman Y., (2017, May). "Real-time monitoring of traffic congestions" In proceedings of 2017 IEEE International Conference on Electro Information Technology (EIT-2017, Lincoln, Nebraska, USA, pp. 501-505.
[51] Wiseman, Y., (2017), "Tool for online observing of traffic congestions", International Journal of Control and Automation, Vol. 10, No. 6, pp. 27-34.
[52] Wiseman Y., (2017), "Computerized traffic congestion detection system". International Journal of Transportation and Logistics Management, Vol. 1, No.1, pp. 1-8.
[53] Wiseman Y., (2018)., "Efficient Embedded Computing Component for Anti-Lock Braking System", International Journal of Control and Automation, Vol. 11, No. 12, pp. 1-10.
[54] Wiseman Y., (2018), "Ancillary ultrasonic rangefinder for autonomous vehicles", International Journal of Security and its Applications", Vol. 12, No. 5, pp. 49-58.
[55] Wiseman, Y. (2019), "Driverless cars will make passenger rail obsolete [opinion
[56] Wiseman Y., (2017), "Self-Driving Car - A Computer will Park for You", International Journal of Engineering & Technology for Automobile Security, Vol. 1, No. 1, pp. 9-16.
[57] Wiseman Y., (2017), "Remote Parking for Autonomous Vehicles", International Journal of Hybrid Information Technology, Vol. 10, No. 1, pp. 313-324.
[58] Wiseman Y., (2014), "Device for Detection of Fuselage Defective Parts", Information Journal, Tokyo, Japan, Vol. 17(9(A)), pp. 4189-4194.
[59] Wiseman Y., (2013), "Fuselage Damage Locator System", Advanced Science and Technology Letters, Vol. 37, pp. 1-4.
[60] Wiseman Y., (2010), "Take a Picture of Your Tire!", Proc. IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010) Qingdao, ShanDong, China, pp. 151-156.
[61] Wiseman Y., (2013), "The Effectiveness of JPEG Images Produced By a Standard Digital Camera to Detect Damaged Tyres", World Review of Intermodal Transportation Research, Vol. 4, No. 1, pp. 23-36.
[62] Wiseman Y., (2013), "Camera That Takes Pictures of Aircraft and Ground Vehicle Tires Can Save Lives", Journal of Electronic Imaging, Vol. 22, No. 4, 041104.

[63] Wiseman Y., (2017), "Safety Mechanism for SkyTran Tracks", International Journal of Control and Automation, Vol. 10, No. 7, pp. 51-60.
[64] Wiseman Y., (2017), "Automatic Persistent Inspection of SkyTran Track System", http://u.cs.biu.ac.il/~wiseman/skytran1.pdf .
[65] Grinberg I. and Wiseman Y., (2007), "Scalable Parallel Collision Detection Simulation", In Proceedings of Signal and Image Processing, Honolulu, Hawaii, pp. 380-385.
[66] Grinberg I. and Wiseman Y., (2013), "Scalable Parallel Simulator for Vehicular Collision Detection", International Journal of Vehicle Systems Modelling and Testing, Inderscience Publication, Vol. 8, No. 2, pp. 119-144.
[67] Wiseman Y., K. Schwan and P. Widener, (2004), "Efficient End to End Data Exchange Using Configurable Compression", Proceedings of The 24th IEEE Conference on Distributed Computing Systems (ICDCS 2004), Tokyo, Japan, pp. 228-235.
[68] P. Weisberg and Wiseman Y., (2009), "Using 4KB Page Size for Virtual Memory is Obsolete", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2009), Las Vegas, Nevada, pp. 262-265.
[69] P. Weisberg and Wiseman Y., (2015), "Virtual Memory Systems Should Use Larger Pages rather than the Traditional 4KB Pages", International Journal of Hybrid Information Technology, Vol. 8(8), pp. 57-68.
[70] Wiseman Y., (2017), "Automatic Alert System for Worn Out Pipes in Autonomous Vehicles", International Journal of Advanced Science and Technology, Vol. 107, pp. 73-84.
[71] Wiseman Y. and Grinberg I., (2016), "When an Inescapable Accident of Autonomous Vehicles is Looming", International Journal of Control and Automation, Vol. 9 No. 6, pp. 297-308.
[72] Wiseman Y. and Grinberg I., (2016), "Autonomous Vehicles Should Not Collide Carelessly", Advanced Science and Technology Letters, Vol. 133, pp. 223-228.
[73] Wiseman Y. and Grinberg I., (2016), "Circumspectly Crash of Autonomous Vehicles", Proceedings of IEEE International Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 382-386.
[74] Y. Wiseman, "Diminution of JPEG Error Effects", The Seventh International Conference on Future Generation Information Technology, Vol. 117, pp. 6-9, (2015).
[75] Y. Wiseman, "Alleviation of JPEG Inaccuracy Appearance", International Journal of Multimedia and Ubiquitous Engineering, Vol. 11(3), pp. 133-142, (2016).
[76] Y. Wiseman, "Enhancement of JPEG compression for GPS images", International Journal of Multimedia and Ubiquitous Engineering, Vol. 10, No. 7, pp. 255-264, (2015).
[77] Y. Wiseman, "Improved JPEG Based GPS Picture Compression", Advanced Science and Technology Letters, (2015).
[78] Y. Wiseman, "The still image lossy compression standard - JPEG", Encyclopedia of Information Science and Technology, Third Edition, Vol. 1, Chapter 28, (2014).
[79] Y. Wiseman, "A Pipeline Chip for Quasi Arithmetic Coding", IEICE Journal - Trans. Fundamentals, Tokyo, Japan, Vol. E84-A No.4, pp. 1034-1041, (2001).
[80] Y. Wiseman, "Burrows-Wheeler Based JPEG", Data Science Journal, Vol. 6, pp. 19-27, (2007).

[81] Y. Wiseman, "Efficient Embedded Images in Portable Document Format (PDF)", International Journal of Advanced Science and Technology, Vol. 124, pp. 129-138, (2019).
[82] Y. Wiseman and E. Fredj, "Contour Extraction of Compressed JPEG Images", ACM - Journal of Graphic Tools, Vol. 6, No. 3, pp. 37-43, (2001).
[83] E. Fredj and Y. Wiseman, "An O(n) Algorithm for Edge Detection in Photos Compressed by JPEG Format", Proc. International Conference on Signal and Image Processing SIP-2001, Honolulu, Hawaii, pp. 304-308, (2001).
[84] Y. Wiseman, "Adjustable and Automatic Flush Toilet", International Journal of Control and Automation, Vol. 13, No. 4, pp. 1-10, (2020).
[85] D. Livshits and Y. Wiseman, "Cache Based Dynamic Memory Management for GPS", Proceedings of IEEE Conference on Industrial Electronics (IEEE ICIT-2011), Auburn, Alabama, pp. 441-446, (2011).
[86] D. Livshits and Y. Wiseman, "The Next Generation GPS Memory Management", International Journal of Vehicle Information and Communication Systems, Vol. 3(1), pp. 58-70, (2013).
[87] R. B. Yehezkael, Y. Wiseman, H. G. Mendelbaum & I. L. Gordin, "Experiments in Separating Computational Algorithm from Program Distribution and Communication", LNCS of Springer Verlag Vol. 1947, pp. 268-278, 2001.
[88] Y. Wiseman, "ARC Based SuperPaging", Operating Systems Review, Vol. 39(2), pp. 74-78, 2005.
[89] Y. Wiseman, "Advanced Non-Distributed Operating Systems Course", ACM - Computer Science Education, Vol. 37(2), pp. 65-69, 2005.
[90] M. Reuven & Y. Wiseman, "Reducing the Thrashing Effect Using Bin Packing", Proc. IASTED Modeling, Simulation, and Optimization Conference, MSO-2005, Oranjestad, Aruba, pp. 5-10, 2005.
[91] M. Reuven & Y. Wiseman, "Medium-Term Scheduler as a Solution for the Thrashing Effect", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 49(3), pp. 297-309, 2006.
[92] Y. Wiseman, "The Relative Efficiency of LZW and LZSS", Data Science Journal, Vol. 6, pp. 1-6, 2007.
[93] Y. Wiseman & I. Gefner, "Conjugation Based Compression for Hebrew Texts", ACM Transactions on Asian Language Information Processing, Vol .6(1), article no. 4, 2007.
[94] I. Grinberg & Y. Wiseman, "Scalable Parallel Collision Detection Simulation", Proc. Signal and Image Processing (SIP-2007), Honolulu, Hawaii, pp. 380-385, 2007.
[95] Y. Wiseman, "ASOSI: Asymmetric Operating System Infrastructure", Proc. 21st Conference on Parallel and Distributed Computing and Communication Systems, (PDCCS 2008), New Orleans, Louisiana, pp. 193-198, 2008.
[96] Y. Wiseman, J. Isaacson & E. Lubovsky, "Eliminating the Threat of Kernel Stack Overflows", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2008), Las Vegas, Nevada, pp. 116-121, 2008.

[97] M. Itshak & Y. Wiseman, "AMSQM: Adaptive Multiple SuperPage Queue Management", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2008), Las Vegas, Nevada, pp. 52-57, 2008.
[98] R. Ben Yehuda & Y. Wiseman, "The Offline Scheduler for Embedded Transportation Systems", Proc. IEEE Conference on Industrial Electronics (IEEE ICIT-2011), Auburn, Alabama, pp. 449-454, 2011.
[99] Y. Wiseman & P. Weisberg, "Economical Memory Management for Avionics Systems", IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), 2013.
[100] Y. Wiseman & Alon Barkai, "Diminishing Flight Data Recorder Size", IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), 2013.
[101] R. Ben Yehuda & Y. Wiseman, "The Offline Scheduler for Embedded Vehicular Systems", International Journal of Vehicle Information and Communication Systems, Vol. 3(1), pp. 44-57, 2013.
[102] Y. Wiseman & Alon Barkai, "Smaller Flight Data Recorders", Journal of Aviation Technology and Engineering, Vol. 2(2), pp. 45-55, 2013.
[103] P. Weisberg & Y. Wiseman, "Efficient Memory Control for Avionics and Embedded Systems", International Journal of Embedded Systems, Vol. 5(4), pp. 225-238, 2013.
[104] Y. Wiseman, "Steganography Based Seaport Security Communication System", Advanced Science and Technology Letters, Vol. 46, pp. 302-306, 2014.
[105] P. Weisberg, Y. Wiseman & J. Isaacson, "Enhancing Transportation System Networks Reliability by Securer Operating System", Open Journal of Information Security and Applications, Vol. 1(1), pp. 24-33, 2014.
[106] Y. Wiseman, "Noise Abatement at Ben-Gurion International Airport", Advanced Science and Technology Letters, Vol. 67, pp. 84-87, 2014.
[107] Y. Wiseman, "Protecting Seaport Communication System by Steganography Based Procedures", International Journal of Security and Its Applications, Sandy Bay, Tasmania, Australia, Vol. 8(4), pp. 25-36, 2014.
[108] Y. Wiseman, "Noise Abatement Solutions for Ben-Gurion International Airport", International Journal of U- & E-Service, Science & Technology, Vol. 7(6), pp. 265-272, 2014.
[109] P. Weisberg & Y. Wiseman, "Virtual Memory Systems Should use Larger Pages", Advanced Science and Technology Letters, Vol. 106, pp. 1-4, 2015.
[110] Y. Wiseman & Y. Giat, "Red Sea and Mediterranean Sea Land Bridge via Eilat", World Review of Intermodal Transportation Research, Vol. 5(4), pp. 353-368, 2015.
[111] Y. Wiseman, "Can Flight Data Recorder Memory Be Stored on the Cloud?", Journal of Aviation Technology and Engineering, Vol. 6(1), 16-24, 2016.
[112] Y. Wiseman & Y. Giat, "Multi-modal passenger security in Israel", Multimodal Security in Passenger and Freight Transportation: Frameworks and Policy Applications, Edward Elgar Publishing Limited, Chapter 16, pp. 246-260, 2016.
[113] Y. Wiseman, "Traffic Light with Inductive Detector Loops and Diverse Time Periods", Contemporary Research Trend of IT Convergence Technology, Vol. 4, pp. 166-170, 2016.

[114] Y. Wiseman, "Unlimited and Protected Memory for Flight Data Recorders", Aircraft Engineering and Aerospace Technology, Vol. 88(6), pp. 866-872, 2016.
[115] Y. Wiseman, "Conceptual Design of Intelligent Traffic Light Controller", International Journal of Control and Automation, Vol. 9(7), pp. 251-262, 2016.
[116] Y. Wiseman, "Compression Scheme for RFID Equipment", Proc. IEEE International Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 382-386, 2016.
[117] Y. Wiseman, "Efficient RFID Devices", Proc. The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016), Firenze (Florence), Italy, pp. 4762-4766, 2016.
[118] Y. Wiseman and I. Grinberg, "The Trolley Problem Version of Autonomous Vehicles", The Open Transportation Journal, Vol. 12, pp. 105-113, 2018.
[119] Y. Wiseman, "Compaction of RFID Devices using Data Compression", IEEE Journal of Radio Frequency Identification, Vol. 1(3), pp. 202-207, 2018.
[120] Y. Wiseman, "High Occupancy Vehicle Lanes are an Expected Failure", International Journal of Control and Automation, Vol. 12(11), pp. 21-32, 2019.
[121] Y. Wiseman, "Israel Complementary International Airport", International Journal of Control and Automation, Vol. 12(7), pp. 1-10, 2019.
[122] Y. Wiseman, "Adjusted JPEG Quantization Tables in Support of GPS Maps", Journal of Mobile Multimedia, Vol. 17(4), pp. 637-656, 2021.
[123] Y. Wiseman, "Blaumilch Canal on Ayalon Highway", Daaton, 2015, Available online at: http://www.daaton.co.il/Article.aspx?id=3290
[124] Y. Wiseman, "Revisiting the Anti-Lock Braking System", Technical Report, 2021.
[125] Y. Wiseman, "Isolated Territories and Infrastructure development: A case for land transportation investment in Madagascar", Interdisciplinary Approaches to the Future of Africa and Policy Development, IGI Global Publishing, Chapter 5, pp. 78-97, 2022.
[126] Y. Wiseman, J. Isaacson, "Safer Operating System for Vehicle Telematics", technical report, 2010.
[127] Y. Wiseman, J. Isaacson, E. Lubovsky and P. Weisberg, "Kernel Stack Overflows Elimination", Advanced Operating Systems and Kernel Applications: Techniques and Technologies, pp. 1-14, IGI Global, 2010.
[128] Y. Wiseman "Airport in Dothan Valley is Ideal", Technical Report, 2020.
[129] M. Itshak and Y. Wiseman, "Enhancing the Efficiency of Memory Management in a Super-Paging Environment by AMSQM", Advanced Operating Systems and Kernel Applications: Techniques and Technologies, pp. 276-293, IGI Global, 2010.
[130] Y. Wiseman, "Conjoint Reliable Vehicle License Plate Identification System", Technical Report, 2020.
[131] M. Reuven and Y. Wiseman, "Alleviating the Thrashing by Adding Medium-Term Scheduler", Advanced Operating Systems and Kernel Applications: Techniques and Technologies, pp. 118-136, IGI Global, 2010.
[132] Y. Wiseman, "Controlling Dynamic Traffic by Road Expansion Can be Accomplished by Double Decker Roads - Case Study", Technical Report, 2022.
[133] Y. Wiseman, "Rail in Islands is an Expected Failure", Technical Report, 2020.

[134] Y. Wiseman, "Road Planners should not Look Just Right and Left But Rather Should Also Look Up", Technical Report, 2022.
[135] Y. Wiseman "Can a Flight Data Recorder be Situated in a Cloud?", Technical Report, 2016.
[136] Y. Wiseman, "JPEG Quantization Tables for GPS Maps", Automatic Control and Computer Sciences, Vol. 55(6), 2021.
[137] Y. Wiseman, "Intelligent Transportation Systems along with the COVID-19 Guidelines will Significantly Change the Transportation Market", Techical Report, 2021.
[138] Y. Wiseman, "Cracked Pipes Alert System for Autonomous Vehicles", Technical Report, 2017.
[139] Y. Wiseman, "EPC Compression", Technical Report, 2016.
[140] Y. Wiseman, "Warning System for Cracked Pipes in Autonomous Vehicles", Advances of Machine Learning in Clean Energy and Transportation Industry, Chapter 9, pp. 261-276, 2021.
[141] Y. Wiseman, "Madagascar had better invest in a single land transportation infrastructure", Interdisciplinary Approaches to the Future of Africa and Policy Development, IGI Global Publishing, 2021.
[142] M. Dreyfuss and Y. Giat, "Optimal spares allocation to an exchangeable-item repair system with tolerable wait", European Journal of Operational Research 261 (2), pp. 584-594, 2017.
[143] Y. Giat, "The effects of output growth on preventive investment policy", American Journal of Operations Research 3 (06), pp. 474-486, 2013.