
References

- [1]. Andrew Borg. (2006). Coarse grain memory management in real-time systems. University of York, UK
- [2]. Apache Software Foundation (2013). Memory management with pools. ”[http://www.fmc-modeling.org/category/projects/apache/amp/3 3Extending Apache.html](http://www.fmc-modeling.org/category/projects/apache/amp/3%20Extending%20Apache.html)”.
- [3]. Ban’us, J. M., Arenas, A., and Labarta, J. (2002). An efficient scheme to allocate soft-a-periodic tasks in multiprocessor hard real-time systems. In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications - Volume 2, PDPTA ’02, (pp. 809–815). CSREA Press.
- [4]. Barrett, D. A. and Zorn, B. G. (1993). Using lifetime predictors to improve memory allocation performance. SIGPLAN Not., 28(6): (pp. 187–196).
- [5]. Bays, C. (1977). A comparison of next-fit, first-fit, and best-fit. Commun. ACM, 20(3): (pp. 191–192).
- [6]. Berger, E. D., McKinley, K. S., Blumofe, R. D., and Wilson, P. R. (2000). Hoard: a scalable memory allocator for multithreaded applications. SIGPLAN Not., 35(11): (pp. 117–128).
- [7]. Bohra, A. and Gabber, E. (2001). Are mallocs free of fragmentation? In Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference, (pp. 105–117), Berkeley, CA, USA. USENIX Association.
- [8]. Bolosky, W. J. and Scott, M. L. (1993). False sharing and its effect on shared memory performance. In In Proceedings of the USENIX Symposium on Experiences with Distributed and Multiprocessor Systems (SEDMS IV), (pp. 57–71).
- [9]. Brent, R. P. (1989). Efficient implementation of the first-fit strategy for dynamic storage allocation. ACM Trans. Program. Lang. Syst., 11(3): (pp. 388–403).
- [10]. Broquedis, F., Clet-Ortega, J., Moreaud, S., Furmento, N., Goglin, B., Mercier, G., Thibault, S., and Namyst, R. (2010). hwloc: A generic framework for managing hardware affinities in hpc applications. In Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on, (pp. 180 –186).
- [11]. Burns, A. and Wellings, A. J. (2001). Real-Time Systems and Programming Languages. Addison Wesley, 3rd edition.

- [12]. Changwoo Min, Inhyeok Kim, Taehyoung Kim, Young Ik Eom. (2012). VMMB: Virtual Machine Memory Balancing for Unmodified Operating Systems. Published in Journal of Grid Computing. 10(1): (pp. 69-84)
- [13]. Chen, S., Xu, J., Nakka, N., Kalbarczyk, Z., and Iyer, R. (2005). Defeating memory corruption attacks via pointer taintedness detection. In Proceedings of the 2005 International Conference on Dependable Systems and Networks, DSN '05, (pp. 378 – 387), Washington, DC, USA. IEEE Computer Society.
- [14]. Chowdhury, S. K. and Srimani, P. K. (1987). Worst case performance of weighted buddy systems. Acta Informatica, 24:555–564. 10.1007/BF00263294.
- [15]. Christian Del Rosso. (2005). Dynamic Memory Management for Software Product Family Architectures in Embedded Real-Time Systems. Fifth Working {IEEE} / {IFIP} Conference on Software Architecture (pp. 211-212)
- [16]. Christopher J. Rossbach, Owen S. Hofmann, Donald E. Porter, Hany E. Ramadan, Bhandari Aditya, Emmett Witchel. (2007). TxLinux: using and managing hardware transactional memory in an operating system. Published in Proceedings of the 21st {ACM} Symposium on Operating Systems Principles SOSP (pp. 87-102)
- [17]. Confessore, G., Dell’Olmo, P., and Giordani, S. (2001). An approximation result for a periodic allocation problem. Discrete Applied Mathematics, 112(1–3): (pp. 53–72).
- [18]. Cowan, C., Wagle, F., Pu, C., Beattie, S., and Walpole, J. (2000). Buffer overflows: attacks and defenses for the vulnerability of the decade. In DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings, volume 2, (pp. 119–129).
- [19]. Crespo, A., Ripoll, I., and Masmano, M. (2006). Dynamic memory management for embedded real-time systems. 225: (pp. 195–204).
- [20]. Daniel P. Bovet, M. C. (2005). Understanding the Linux Kernel, chapter Chapter 2. O'Reilly, 3rd edition.
- [21]. Dipti Diwase, Shraddha Shah, Tushar Diwase and Priya Rathod. (2012). Survey Report on Memory Allocation Strategies for Real-time Operating System in Context with Embedded Devices. International Journal of Engineering Research and Applications, Vol. 2, Issue 3, (pp.1151-1156).

- [22]. Dirk Vogt, Cristiano Giuffrida, Herbert Bos, Andrew S. Tanenbaum. (2014) Techniques for efficient in-memory checkpointing. Published in Operating Systems Review 48(1) (pp. 21-25)
- [23]. las, N. (2011). nedmalloc. <http://www.nedprod.com/programs/portable/nedmalloc>.
- [24]. Edge, J. (2009). Perfcounters added to the mainline. <http://lwn.net/Articles/336542/>.
- [25]. Felicia Ionescu. (2000). Application-Level Virtual Memory Management in Real-Time Multiprocessor Systems. Proceedings of the {ACM} Symposium on Applied Computing, Villa Olmo, Via Cantoni 1, 22100 Como, Italy. (pp. 610-614)
- [26]. Ferreira, T., Matias, R., Macedo, A., and Araujo, L. (2011). An experimental study on memory allocators in multicore and multithreaded applications. In Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2011 12th International Conference on, (pp. 92 –98).
- [27]. Fredkin, E. (1960). Trie memory. Commun. ACM, 3(9): (pp. 490–499).
- [28]. FSF, F. s. f. (2012a). Glibc, the gnu c library. <http://www.gnu.org/software/libc/libc.html>.
- [29]. FSF, F. s. f. (2012b). The gnu c++ library manual. <http://gcc.gnu.org/onlinedocs/libstdc++>.
- [30]. Garey, M. R. and Johnson, D. S. (1979). A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., San Francisco, CA.
- [31]. Gergov, J. (1996). Approximation algorithms for dynamic storage allocation. In Algorithms — ESA '96, volume 1136, pages 52–61. Springer Berlin / Heidelberg.
- [32]. Gergov, J. (1999). Algorithms for compile-time memory optimization. In Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms, SODA '99, (pp. 907–908), Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- [33]. Gloger, W. (2001). Dynamic memory allocator implementations in linux system libraries.
- [34]. Gloger, W. (2006). ptmalloc2. ”<http://www.malloc.de/en/>”.
- [35]. Grunwald, D., Zorn, B., and Henderson, R. (1993). Improving the cache locality of memory allocation. SIGPLAN Not., 28(6): (pp. 177–186).
- [36]. Hans-Georg Eßer. (2011) Combining memory management and filesystems in an operating systems course. Proceedings of the 16th Annual {SIGCSE} Conference on Innovation and Technology in Computer Science Education, Darmstadt, Germany.
- [37]. Hasan, Y. and Chang, M. (2005). A study of best-fit memory allocators. Computer Languages, Systems & Structures, 31(1): (pp. 35 – 48).

- [38]. Hasan, Y., Chen, W.-M., Chang, J. M., and Gharaibeh, B. M. (2010). Upper bounds for dynamic memory allocation. *IEEE Trans. Comput.*, 59(4): (pp. 468–477).
- [39]. Hirschberg, D. S. (1973). A class of dynamic memory allocation algorithms. *Commun. ACM*, 16(10): (pp. 615–618).
- [40]. Hyde, R. L. and Fleisch, B. D. (1996). An analysis of degenerate sharing and false coherence. *J. Parallel Distrib. Comput.*, 34(2): (pp. 183–195).
- [41]. Hyojun Kim, Sangeetha Seshadri, Clement L. Dickey, Lawrence Chiu. (2014). Phase change memory in enterprise storage systems: silver bullet or snake oil? Published in *Operating Systems Review* 48(1): (pp. 82-89)
- [42]. Ian McDonald. (1999). Distributed, Configurable Memory Management in an Operating System Supporting Quality of Service. *FTDCS 1999*: (pp. 191-196).
- [43]. Ian McDonald. (2001). Memory management in a distributed system of single address space operating systems supporting quality of service. University of Glasgow, UK
- [44]. Jane W. S. Liu. (2000). “Real-time System”, 1st Edition published by Person Education.
- [45]. Jeremiassen, T. E. and Eggers, S. J. (1995). Reducing false sharing on shared memory multiprocessors through compile time data transformations. *SIGPLAN Not.*, 30(8): (pp. 179–188).
- [46]. Johnstone, M. S. and Wilson, P. R. (1998). The memory fragmentation problem: solved? *SIGPLAN Not.*, 34(3): (pp. 26–36).
- [47]. Kaminski, P. (2009). Numa aware heap memory manager. ”[http://amddevcentral.com/Assets/NUMA aware heap memory manager article final.pdf](http://amddevcentral.com/Assets/NUMA%20aware%20heap%20memory%20manager%20article%20final.pdf)”.
- [48]. Kingsley, C. (1982). Description of a very fast storage allocator.
- [49]. Kiyohito Miyazaki, Yoshinari Nomura, Hideo Taniguchi. (2013). Memory Segmentation and Transfer in Mint Operating System. Published in 16th International Conference on Network-Based Information Systems NBIS 2013 (pp. 366-371)
- [50]. Kleen, A. (2005). A NUMA API for Linux. Novel Inc. accessed on September 2011.
- [51]. Knuth, D. (1997). The art of computer programming: Fundamental Algorithms, volume 1. addison-Wesley, 2 edition.
- [52]. Larson, P.-k. and Krishnan, M. (1998). Memory allocation for long running server applications. *SIGPLAN Not.*, 34(3): (pp. 176–185).

- [53]. Lea, D. (1996). A memory allocator. "<http://g.oswego.edu/dl/html/malloc.html>". Unix/Mail December, 1996.
- [54]. Lei Liu, Mengyao Xie, Hao Yang. (2017). Memos: Revisiting Hybrid Memory Management in Modern Operating System. CoRR abs/1703.07725
- [55]. Lei Liu, Yong Li, Chen Ding, Hao Yang, Chengyong Wu. (2016). Rethinking Memory Management in Modern Operating System: Horizontal, Vertical or Random? IEEE Trans. Computers 65(6): (pp. 1921-1935)
- [56]. Linus Torvalds, e. (2011). Source codes of linux kernel v3.0.4. "<http://lxr.linux.no/linux+v3.0.4/>".
- [57]. Liu, T. and Berger, E. D. (2011). Sheriff: precise detection and automatic mitigation of false sharing. SIGPLAN Not., 46(10): (pp. 3–18).
- [58]. Luby, M. G., Naor, J. S., and Orda, A. (1994). Tight bounds for dynamic storage allocation. In Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms, SODA '94, (pp. 724–732), Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- [59]. M. Masmano, I. Ripoll and A. Crespo. (2004). TLSF: A new dynamic memory allocator for real-time systems. Proceedings of 16th Euromicro Conference on Real-Time Systems, Catania, Italy, July 2004, (pp. 79-88).
- [60]. Majo, Z. and Gross, T. R. (2011). Memory system performance in a numa multicore multiprocessor. In Proceedings of the 4th Annual International Conference on Systems and Storage, SYSTOR '11, (pp. 1–10), New York, NY, USA. ACM.
- [61]. Manoj Kumar Putchala, Adam R. Bryant. (2016). Synchron-ITS: An Interactive Tutoring System to Teach Process Synchronization and Shared Memory Concepts in an Operating Systems Course. International Conference on Collaboration Technologies and Systems CTS 2016: (pp. 180-187)
- [62]. Marathe, J. and Mueller, F. (2006). Hardware profile-guided automatic page placement for ccnuma systems. In Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming, PPOPP '06, (pp. 90–99), New York, NY, USA. ACM.

- [63]. Marchand, A., Balbastre, P., Ripoll, I., Masmano, M., and Crespo, A. (2007). Memory resource management for real-time systems. In *Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference*, (pp. 201 –210).
- [64]. María-del-Mar Gallardo, Pedro Merino, David Sanán. (2009). Model Checking Dynamic Memory Allocation in Operating Systems. Published in *Journal of Automation Reasoning* 42(2-4): (pp. 229-264)
- [65]. Mark Oskin, Diana Keen, Justin Hensley, Lucian Vlad Lita, Frederic T. Chong. (2002). Operating Systems Techniques for Parallel Computation in Intelligent Memory. *Parallel Processing Letters* 12(3-4): (pp. 311-326)
- [66]. Masmano (2012). The latest version of TLSF source. <http://wks.gii.upv.es/tlsf/files/src/TLSF-2.4.6.tbz2>.
- [67]. Masmano, M., Ripoll, I., and Crespo, A. (2003). Dynamic storage allocation for real-time embedded systems. *Proc. of Real-Time System Symposium* WIP.
- [68]. Masmano, M., Ripoll, I., Balbastre, P., and Crespo, A. (2008a). A constant-time dynamic storage allocator for real-time systems. *Real-Time Systems*, 40(2): (pp. 149–179).
- [69]. Masmano, M., Ripoll, I., Real, J., Crespo, A., and Wellings, A. (2008b). Implementation of a constant-time dynamic storage allocator. *Software: Practice and Experience*, 38(10): (pp. 995–1026).
- [70]. McCamant, S. and Ernst, M. D. (2007). A simulation-based proof technique for dynamic information flow. In *Proceedings of the 2007 workshop on Programming languages and analysis for security, PLAS '07*, (pp. 41–46), New York, NY, USA. ACM.
- [71]. McCurdy, C. and Vetter, J. (2010). Memphis: Finding and fixing numa-related performance problems on multi-core platforms. In *Performance Analysis of Systems Software (ISPASS), 2010 IEEE International Symposium on*, (pp. 87–96).
- [72]. Meenakshi Sundaram Bhaskaran, Jian Xu, Steven Swanson. (2014). Bankshot: caching slow storage in fast non-volatile memory. *Operating Systems Review* 48(1): (pp. 73-81)
- [73]. Mehta, H., Owens, R., Irwin, M., Chen, R., and Ghosh, D. (1997). Techniques for low energy software. In *Low Power Electronics and Design, 1997. International Symposium*, (pp. 72–75).
- [74]. MicroQuill (2012). shbench benchmark tool. <http://www.microquill.com>.

- [75]. Mochel, P. (2005). The sysfs filesystem. In Linux Symposium, (pp. 313–326), Ottawa, Ontario, Canada.
- [76]. Molnar, I. (2009). Performance counters for linux, v8. <http://lwn.net/Articles/336542/>.
- [77]. MS Johnstone. (1998). Fragmentation Problem: Solved? Proceeding of the Int. Symposium on Memory Management (ISMM'98), Vancouver, Canada. ACM Press
- [78]. Nethercote, N. and Mycroft, A. (2002). The cache behaviour of large lazy functional programs on stock hardware. SIGPLAN Not., 38(2 supplement): (pp. 44–55).
- [79]. Nilsen, K. and Gao, H. (1995). The real-time behavior of dynamic memory management in c++. In Real-Time Technology and Applications Symposium, 1995. Proceedings, (pp. 142–153).
- [80]. Ogasawara, T. (1995). An algorithm with constant execution time for dynamic storage allocation. In RTCSA '95: Proceedings of the 2nd International Workshop on Real-Time Computing Systems and Applications, pages 21–25, Washington, DC, USA. IEEE Computer Society.
- [81]. Ogasawara, T. (2009). Numa-aware memory manager with dominant-thread based copying gc. SIGPLAN Not., 44(10): (pp. 377–390).
- [82]. Oracle Inc. (2013). Memory architecture. "<http://docs.oracle.com/cd/E1407201/server.112/e10713/memory.htm>".
- [83]. Page, I. and Hagins, J. (1986). Improving the performance of buddy systems. Computers, IEEE Transactions on, C-35(5): (pp. 441 –447).
- [84]. Pawel Pisarczyk. (2007). Experimental Dependability Evaluation of Memory Manager in the Real-time Operating System. IJCSA 4(1): (pp. 33-38)
- [85]. Paul Werstein, Hailing Situ, Zhiyi Huang. (2006). "Load Balancing in a Cluster Computer", Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06).
- [86]. Peng Wei, Lihua Yue, Zhanzhan Liu, Xiaoyan Xiang. (2008). Flash memory management based on predicted data expiry-time in embedded real-time systems. Proceedings of the 2008 {ACM} Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008 (pp. 1477-1481)
- [87]. Polishchuk, M., Liblit, B., and Schulze, C. W. (2007). Dynamic heap type inference for program understanding and debugging. SIGPLAN Not., 42(1): (pp. 39–46).

- [88]. Puaut, I. (2002). Real-Time Performance of Dynamic Memory Allocation Algorithms. In ECRTS '02: Proceedings of the 14th Euromicro Conference on Real-Time Systems, (pp. 41–49), Washington, DC, USA. IEEE Computer Society.
- [89]. Puaut, I. and Hardy, D. (2007). Predictable paging in real-time systems: A compiler approach. In Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference on, (pp. 169 –178).
- [90]. Randell, B. (1969). A note on storage fragmentation and program segmentation. Commun. ACM, 12(7).
- [91]. Reza Salkhordeh, Hossein Asadi. (2016) An Operating System level data migration scheme in hybrid DRAM-NVM memory architecture. Automation & Test in Europe Conference & Exhibition. (pp. 936-941)
- [92]. Robert L. Budzinski, Edward S. Davidson. (1981). A Comparison of Dynamic and Static Virtual Memory Allocation Algorithms” IEEE Transactions on software Engineering, Vol. SE-7, NO. 1, January 1981.
- [93]. Robertson, W., Kruegel, C., Mutz, D., and Valeur, F. (2003). Run-time detection of heap-based overflows. In Proceedings of the 17th USENIX conference on System administration, LISA '03, (pp. 51–60), Berkeley, CA, USA. USENIX Association.
- [94]. Robson, J. (1980). Storage allocation is np-hard. Information Processing Letters, 11(3): (pp. 119–125).
- [95]. Robson, J. M. (1971). An estimate of the store size necessary for dynamic storage allocation. J. ACM, 18(3): (pp. 416–423).
- [96]. Robson, J. M. (1977). Worst case fragmentation of first fit and best fit storage allocation strategies. The Computer Journal, 20(3): (pp. 242–244).
- [97]. S. Baskiyar, Ph.D. and N. Meghanathan. (2005). A Survey of Contemporary Real-time Operating Systems Informatica 29, (pp. 233–240)
- [98]. Sangsoo Park, Yonghee Lee, Heonshik Shin. (2004). An experimental analysis of the effect of the operating system on memory performance in embedded multimedia computing. EMSOFT 2004. (pp. 26-33)
- [99]. Sanjay Ghemawat, P. M. (2010). Tcmalloc: Thread-caching malloc. <http://goog-perftools.sourceforge.net/doc/tcmalloc.html>.

- [100]. Seyeon Kim. (2013). Node-oriented dynamic memory management for real-time systems on ccNUMA architecture systems. University of York, UK.
- [101]. SGI (2004). Standard template library programmer's guide: Allocators. "http://www.sgi.com/tech/stl/Allocators.html".
- [102]. Shen, K. K. and Peterson, J. L. (1974). A weighted buddy method for dynamic storage allocation. *Commun. ACM*, 17(10) (pp. 558–562).
- [103]. Shore, J. E. (1975). On the external storage fragmentation produced by first-fit and best-fit allocation strategies. *Commun. ACM*, 18(8): (pp. 433–440).
- [104]. Standish, T. A. (1980). *Data Structure Techniques*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [105]. Stephenson, C. J. (1983). New methods for dynamic storage allocation (fast fits). *SIGOPS Oper. Syst. Rev.*, 17(5): (pp. 30–32).
- [106]. Sybase Inc. (2013). Configuration parameters that affect memory allocation.
- [107]. Tao, J., Kunze, M., and Karl, W. (2008). Evaluating the cache architecture of multicore processors. In *Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference*, (pp. 12 –19).
- [108]. Tiancheng Liu, Ying Li, Andrew Schofield, Matt Hogstrom, Kewei Sun, Ying Chen. (2008). Partition-based heap memory management in an application server. *Operating Systems Review* 42(1): (pp. 98-103)
- [109]. Tikir, M. and Hollingsworth, J. (2008). Hardware monitors for dynamic page migration. *Journal of Parallel and Distributed Computing*, 68(9): (pp. 1186–1200).
- [110]. Timothy Wood, Gabriel Tarasuk-Levin, Prashant J. Shenoy, Peter Desnoyers, Emmanuel Cecchet, Mark D. Corner. (2009). Memory buddies: exploiting page sharing for smart colocation in virtualized data centers. Published in *Operating Systems Review* 43(3): (pp. 27-36)
- [111]. Vatsal Shah, Kanu Patel. (2012). Load Balancing algorithm by Process Migration in Distributed Operating System. *International Journal of Computer Science and Information Technology & Security (IJCSITS)*, ISSN: 2249-9555, Vol. 2, No.6.
- [112]. V Shah, A Shah. (2017). Critical Analysis for Memory Management Algorithm for NUMA based Real-time Operating System. *IEEE Xplore*.

- [113]. V Shah, A Shah. (2018). Proposed Memory Allocation Algorithm for NUMA based Soft Real-time Operating System. International Conference On Emerging Technologies In Data Mining And Information Security (IEMIS 2018)
- [114]. Vatsal Shah, Apurva Shah. (2016). An Analysis and Review on Memory Management Algorithms for Real-time Operating System. International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 5.
- [115]. Vee, V.-Y. and Hsu, W.-J. (1999). A scalable and efficient storage allocator on shared memory multiprocessors. In Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks, ISPAN '99, Washington, DC, USA. IEEE Computer Society.
- [116]. Wilson, P. R., Johnstone, M. S., Neely, M., and Boles, D. (1995b). Dynamic Storage Allocation: A Survey and Critical Review. In IWMM '95: Proceedings of the International Workshop on Memory Management, (pp. 1–116), London, UK. Springer-Verlag.
- [117]. Wilson, P., Johnstone, M., Neely, M., and Boles, D. (1995a). Memory allocation policies reconsidered. Technical report, Technical report, University of Texas at Austin Department of Computer Sciences.
- [118]. XiaoHui Sun, JinLin Wang, xiao chan. (2007). “An Improvement of TLSF Algorithm”.
- [119]. Youngki Chung, Ramakrishna M, Jisung Kim and Woohyong Lee. (2008). Smart Dynamic Memory Allocator for embedded systems. Proceedings of 23rd International Symposium on Computer and Information Sciences, ISCIS '08.
- [120]. Zhou, X. and Petrov, P. (2011). Towards virtual memory support in real-time and memory-constrained embedded applications: the interval page table. Computers Digital Techniques, IET, 5(4): (pp. 287 –295).