A New Scheme of Multiple Automated Guided Vehicle System For Collision and Deadlock Free

Shengjun Shao[1,2,3,4], Zeyang Xia[1,2,3*],Guodong Chen[5], Jun Zhang[1,2,3], Ying Hu[1,2,3*], Jianwei Zhang[6]
1.Center for Cognitive Technologies, Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences, Shenzhen, China
2.Guangdong Provincial Key Laboratory of Robotics and Intelligent System,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
3.The Chinese University of Hong Kong, Hong Kong, China
4.University of Chinese Academy of Sciences,Beijing,China
5.Robotics and Microsystem Center, Soochow University, Suzhou, China
6.TAMS, Department of Informatics, University of Hamburg, Hamburg, Germany
* Correspondence should be addressed to Profs. Xia and Hu
at zy.xia/ying.hu@siat.ac.cn;+86-755-8639-2181/2182

***Abstract*—Conflicts and deadlocks are key issues in a multiple automated guided vehicle system, especially in complicated and uncertain environments. They have been challenges for ensuring the reliability, security, and efficiency of the multiple automated guided vehicle system. In this paper, a traffic control model based on semaphores is proposed to resolve conflict and deadlock problems in application of multiple automated guided vehicle system. Prototype experiments have been conducted and validated the function and performance of the proposed model.**

***Index Terms*—Automated guided vehicle; semaphore; conflict; deadlock; traffic control model.**

I. Introduction

Automated guided vehicles (AGVs) have been widely used in industrial fields, such as in flexible manufacturing systems, logistics warehouse, and assembly line materials processing. Various methods for AGV routing and scheduling are currently in use. However, new approaches which take advantage of faster and computationally more efficient algorithms are still the subjects of intensive researches, especially the way to avoid the conflict and deadlock in multi-AGV system. Moreover, it must guarantee the efficiency of the transportation.

The purpose of scheduling and routing [1] is not only to generate a shortest path set [2], but also to consider dispatch tasks and how to avoid conflict and deadlock [3]. In [4], a colored resource-oriented Petri net modeling method was used to deal with conflict and deadlock arising in multi-AGV system. Distributive traffic control Scheme was used to routing and scheduling in multi-AGV system by Jing, et al. [5]. This method maintains a minimum headway between two adjacent vehicles, but it was supposed to be used for special cases only. Deadlock occurs more frequently while the number of AGVs increases. Thus the centralized traffic control scheme is preferred in a multi-AGV system. Vedran, et al. [6] utilized a conventional method based on the well-known Banker's algorithm to improve resource utilization and unavailability, as well as to resolve some potential deadlocks. However, it cannot be executed in the system when network model contains a large number of nodes.

Jcong-Hoon and Bum et al. [7] proposed a real-time traffic control scheme in which a k-shortest path search algorithm was employed to construct path set, thus the online motion planning operation was performed in real time. It seems a good solution to resolve conflict and deadlock issues. Nevertheless, the particularly high requirements on the real time illustrates that this scheme must rely on specific environment. Some researchers [8] [9] [10] tried to adopt a conflict-free bidirectional AGV routing and scheduling algorithm based on the time window approach. These methods can offer a flexible and optimal solution to the vehicle routing and scheduling problem. However, they failed to address the issue of conflict and deadlock in complicated and uncertain environments.

In summary, the problem of the above methods is that they cannot guarantee no conflict or deadlock in system, simultaneously ensure that the algorithm and scheme can perform normally in dynamic, complicated, and uncertain environments. In order to resolve this, in this paper we propose a new method in which a centralized control architecture is adopted. We also use two-stage traffic control scheme [11] for routing and scheduling of a multi-AGV system. Meanwhile, we propose a novel traffic control model based on semaphore to synchronize AGVs.

The remainder of this paper is organized as follows. Section II provides the basic system descriptions. Section III

*This research was partially supported by the National Science Foundation of China (No. 61210013 and 51305436) and Key Fundamental Research Program of Shenzhen (JC201005270375A)

presents the proposed semaphore-based traffic control model so as to avoid conflict and deadlock in complicated and uncertain environments. Section IV validates the reliability and feasibility of the proposed method. Section V summarizes our study.

II. SYSTEM OVERVIEW

In multi-AGV system, we adopt client-server architecture, such as the centralized control architecture is employed. Centralized control unit manages global information, accepts task request and queues all the tasks according to priority, manages path planning and traffic control. Each AGV is equipped with sensors including gyroscope, laser scanner, radio frequency identification (RFID) tag antenna, and encoder. AGV can not only detect obstacles, but also calculate their positions through encoder and gyroscope, and correct the absolute position through short magnetic stripe and RFID tag antenna. All AGVs interact with the server through wireless network. There is no interaction between any two AGVs. Each AGV sends its position to the central controller every 200 milliseconds. Central controller then utilizes the position data to update user interface (UI) and synchronize AGVs in case the conflict and deadlock occurs in the system. Meanwhile, central controller sends action commands (e.g. stop command, start command, and charging command) to a specific AGV.

A two-staged central controller was applied in our multi-AGV system in order to obtain optimal motion. Fig. 1 shows the architecture of the central controller. In the first stage, the path generator assigns moving path for each AGV offline. In the second stage, the traffic controller operates each moving AGV online. In additional, the UI provides the position and state of each AGV. Global information and key data about each AGV are stored in the database. The system inquires and stores data during operation. More specifically, central controller utilizes A* algorithm to construct optimal path set for AGV which has been scheduled task and performed it. Avoiding conflict and deadlock is essential when multiple AGVs are moving. The online semaphore-based traffic control model is used to avoid conflict and deadlock.

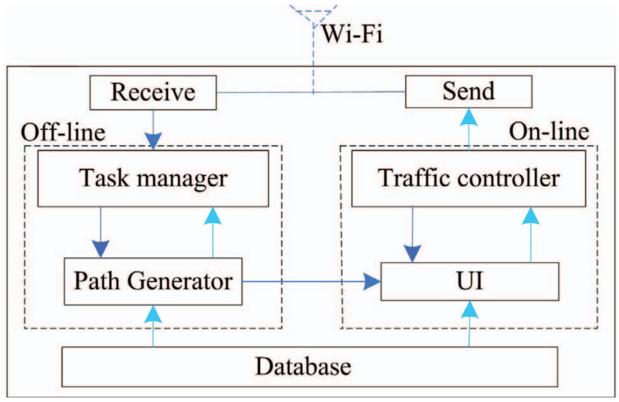


Fig. 1. Architecture of the central controller.

III. SEMAPHORE-BASED TRAFFIC CONTROL MODEL

Synchronization of concurrent activity using semaphore has been widely used in industrial systems. In this paper the theory of semaphore was used to create the traffic control model, which resolves the issue of conflict and deadlock in complicated and uncertain environments.

A. Description of the semaphore-based traffic control model

In semaphore-based traffic control model, two types of semaphores [12] are defined, binary semaphore (BS) and counting semaphore (CS). BS is used to control access to intersection, while CS is used control access to bidirectional lane. Both BS and CS can be hold by each AGV. In this model, BS and CS are described using data structure including a counter, a maximum resource count (C_{max}), a list of semaphore owners, and a list of semaphore waiters.

The C_{max} identifies the maximum number of resources that the semaphore can control. The counter indicates the number of these resources which are currently available. Owner queue with FIFO (First In First Out) property stores ID of AGV occupying resource controlled by the semaphore. Waiter queue with FIFO property stores ID of AGV waiting for the resource.

Initially, the of BS is set to 1 meaning only one AGV can own BS, while the C_{max} of CS is set to any integer determined by resource. Both owner queue and waiter queue are initially empty. Counters of both BS and CS are initially set to the because no AGV has owned any semaphore. Four operations, 'down', 'up', 'sleep', and 'wake', are defined to both BS and CS (Table I). With an 'up' operation on a semaphore the counter is incremented by 1 and AGV's ID is removed from owner queue, while with an 'down' operation the counter is decremented by 1 and AGV's ID is added to owner queue. With a 'sleep' operation AGV's ID is added to waiter queue, while with a 'wake' operation AGV's ID is removed from waiter queue.

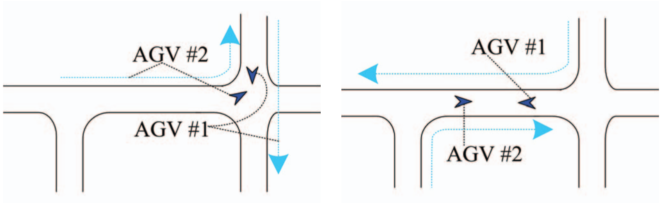
TABLE I
SEMAPHORE CHANGES WHEN OPERATION TO SEMAPHORE

operation	System changed Semaphore
up	counter = counter + 1
	remove AGV's ID from owner queue
down	counter = counter - 1
	add AGV's ID to owner queue
sleep	add AGV's ID to waiter queue
wake	remove AGV's ID from waiter queue

Two types of traffic light called binary semaphore light (BSL) and counting semaphore light (CSL) are defined respectively in order to indicate traffic conditions. BSL controls access to intersection, while CSL controls access to bidirectional lane. Each of BSL and CSL has two states, green or red. Green state denotes that the path is available for AGV, while red state denotes that the path has been occupied.

B. Solution to various conflicts

In real environment, there are usually two scenarios of conflicts in the AGV system. The first conflict occurs at an intersection (Fig. 2 (a)). The second conflict occurs when the path is bidirectional lane (Fig. 2 (b)). In these scenarios, there are two types of path segments, unidirectional lane and bidirectional lane.



(a) (b)

Fig. 2. Two scenarios of conflicts.(*Note that in Figs. 2-6, blue arrow denotes AGV, cyan and orange arrow lines denote AGV's path sets, red dot denotes that BSL is red, green dot denotes that BSL is green, red ellipse denotes that CSL is red, and green ellipse denotes that CSL is green)

To prevent a cross conflict, at the same time only one AGV has access to the intersection. BS whose counter initializes as 1 is used to address this issue, as mentioned in subsection A. Each intersection has a BS which is used to control intersection. Let AGV#1 and #2 approach an intersection at the same time. The one granted with access to the BS is determined based on its task priority. The AGV granted with access, say AVG#1, will continue move, then the system performs 'down' operation on BS.

As soon as AGV#2 attempts to enter the same intersection, the system will detect BS and find that the intersection has been occupied. Therefore, the system performs a 'sleep' operation on BS. As a result, the system prevents AGV#2 entering this intersection. When AGV#1 leaves the intersection, the system will perform an 'up' operation on BS. Then a 'wake' operation is performed on BS, and AGV#2 is allowed to move into the intersection. Afterword, a 'down' operation is performed on BS. Fig. 3 describes the process of synchronization mechanisms based on BS.

In case of a head-on conflict, we employ two CSs with same structures, p CS and q CS, to synchronize AGVs on one bidirectional lane. Furthermore, two CSLs denoted as p CSL and q CSL are arranged at both ends of the lane to indicate the state of the lane (Fig. 4).

Initially, system has to set the counter of p CS and q CS to the which denotes how many AGVs can travel in the same direction at the same time in one bidirectional lane. Two semaphores control the entrance of the lane at both ends respectively. p CS is the entrance's CS relative to AGV#2,

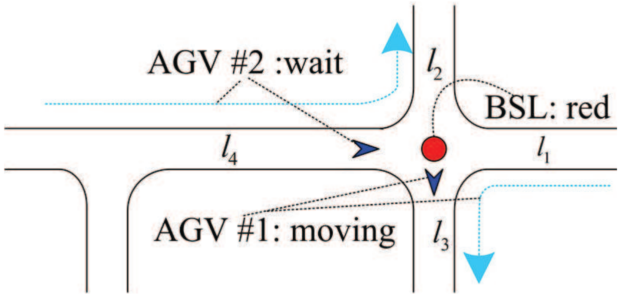


Fig. 3. Using BS to solve cross conflict.

while p CS is the exit's CS relative to AGV#1. The protocol to avoid head-on conflict, named as SMSL (Synchronize Multi-AGVs in a Segment Lane), is described below.

SMSL Protocol:

When AGV plans to enter a bidirectional lane, the system checks the entrance's CS. If the CS's counter is not equal to or the counter of exit's CS equals zero, the system performs a 'sleep' operation on exit's CS, otherwise a 'down' operation.

End of SMSL Protocol

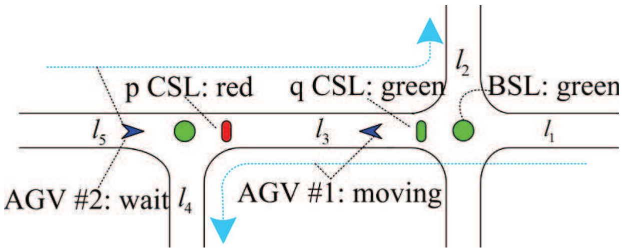


Fig. 4. Using CS to solve head-on conflict.

Now, we explain the details of how the SMSL protocol applies to the conflict between AGV#1 and #2 (Fig. 4). Assume AGV#1 and #2 moves along the path sets P1={...,l_1,l_3,l_4,...} and P2={...,l_5,l_3,l_2,...} respectively. Suppose that AGV#1 just drives out of l_1, and attempts to enter l_3 (a bidirectional lane). At this moment, the system applies SMSL protocol. Because the q CSL is green, AGV#1 moves. When AGV#2 attempts to enter l_3 from l_5 and AGV#1 is driving on l_3, the system applies SMSL protocol on l_3. Since the p CSL has turned to be red after first SMSL protocol is applied on l_3, the system will prevent AGV#2 getting into l_3. When AGV#1 drives out of l_3, the system performs an 'up' operation on p CS. Then AGV#2 moves as p CSL is green. Simultaneously the system performs a 'wake' operation and a 'down' operation on q CS.

Assume that AGV#3 or more AGVs arrive at intersection between l_1 and l_3, and plan to enter l_3 when AGV#1 still occupies l_3, the system applies SMSL protocol on l_3. As a result, AGV#3 or more AGVs enter l_3. AGV#2 must wait until l_3 is clear, meaning the counter of p CS is equal to the C_{max} of p CS (p CSL is green). Thus, head-on conflict has been avoided.

C. Solution to deadlock

The problem of deadlock occurs when vehicles are moving to a section of a bidirectional lane. Fig. 5 (a) describes the deadlock in a multi-AGV system. Assume that AGV#1 and #2 moves along the path sets P1={...,l_2,l_4,l_6,l_7,...} and P2={...,l_8,l_6,l_4,...} respectively. After AGV#1 entered l_6, AGV#2 enters l_6 before AGV#2 drives out of l_4. The system applies SMSL protocol on l_4 when AGV#1 enters l_4, and applies SMAL protocol to l_6 when AGV#2 enters l_6. Afterward, AGV#1 attempts to enter l_6 and AGV#2 attempts to enter l_4. However, they are prevented by the system because they occupy desired lane of each other. The above deadlock appears in multi-AGV system. To resolve this problem, a protocol named as SMCL (Synchronize Multi-AGVs in Continuous Lane) is proposed as described below.

SMCL Protocol:

When AGV attempts to enter a bidirectional lane, the system checks the exit's CS. If the system finds the AGV's ID in the CS's owner or waiter queue, it stops checking the next lane. Otherwise the system applies SMSL protocol to the current lane. The system will then check if the next lane is a bidirectional one. If yes, the system applies SMSL protocol to the next lane. If the next lane is a unidirectional one or does not exit, the system stops checking.

End of SMCL Protocol

Now, we use SMCL protocol to solve deadlock described in Fig .5 (a). The system applies SMCL protocol when AGV#1 attempts to enter l_4. As a result, CSLs are red (Fig. 5 (b)). Thus AGV#2 has to wait until AGV#1 drives out of l_6.

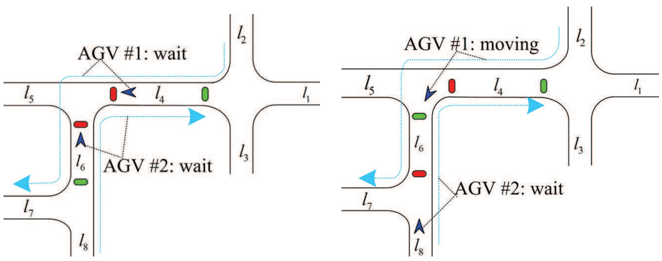


Fig. 5. Avoid deadlock by SMCL protocol.(BSL is not shown in the picture, and it exits in fact)

In very rare situations, the above process of applying the SMCL protocol may cause a new deadlock (Fig. 6). System applies SMCL when AGV#3 enters l_1. AGV#1 attempts to enters l_4 from l_5, but failed because AGV#3 has occupied l_4. AGV#2 enters l_4 from l_5 before AGV#3 enters l_4. As a result, deadlock between AGV#1 and #2 occurs. To resolve this problem, another protocol named as ESMCL (Enhanced SMCL) is proposed as described below.

ESMCL Protocol:

The system checks entrance's CS when an AGV attempts to drive into a bidirectional lane. If the counter of entrance's CS is equal to C_{max} and the waiter queue is empty, the system applies SMCL protocol and the AGV continues to move. If the waiter queue of CS is not empty, the system will check if there is any other AGV in the neighboring bidirectional lanes. If the result is negative, the system applies SMCL protocol and the AGV continues to move. Otherwise, the AGV stops and waits until the neighboring bidirectional lanes are clear.

End of ESMCL Protocol

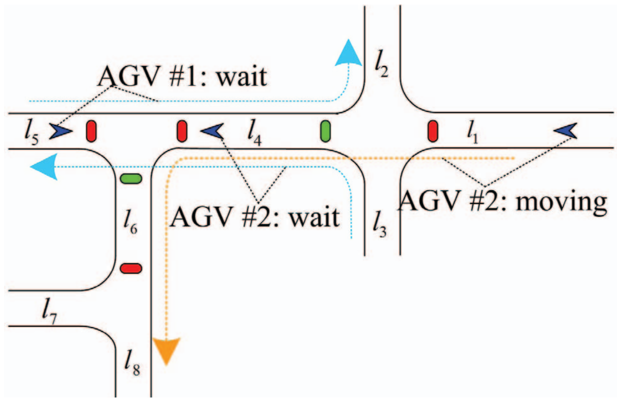


Fig. 6. The new deadlock in very rare situations.

Subsections B and C have described the solutions to conflict and deadlock for multi-AGV system respectively. The algorithm below shows how they are integrated to work.

Algorithm 1: Avoid conflict and deadlock

1 The system detects the type of semaphore which controls current intersection and entrance to the next lane.
2 ***if*** only BS exits, ***then***
3 System checks the counter of the BS.
4 ***if*** counter = , ***then***
5 System lets AGV move, and executes a 'down' operation on BS.
6 ***else***
AGV waits, and algorithm jumps to 3.
end if
7 ***else*** */* both BS and CS exist*/*
8 System checks the counter of the BS and CS.
9 ***if*** any one of the counters does not equal respective , ***then***
10 repeat 8.
11 ***else***
12 System performs a 'down' operation on BS, and applies ESMCL protocol to the rest of the path.
end if
end if
13 When AGV drives out each intersection, system performs an 'up' operation on BS which controls the intersection.

IV. EXPERIMENT AND RESULTS

The scheduler control system using semaphore-based traffic control model shown in Fig. 7 was developed using C++ and runs on Windows system. The map in the right of Fig. 7 shows a part of the testing field with a scale of 1:200. In this map, each AGV is marked as a red arrow with real time state information. The green lines are the AGV's virtual guiding paths. The large green and red dots are virtual BSLs, while the small green and red dots are virtual CSLs. The lanes dotted with CSLs are bidirectional, with those not dotted are unidirectional ones. There are 12 loading and unloading stations (see A1, B2, D1 in Fig.7) in the testing filed.

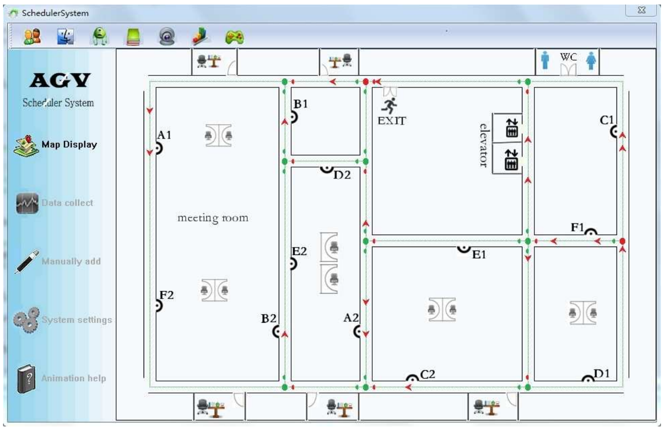


Fig. 7. The panel of the scheduler control system.

The experiments conducted were designed as described below. First a task package is added to the system (see Table II for task package details). Then set one AGV available for use. After several minutes, set more AGV available for use. We record the path AGV drives and the time system assigned each task to AGVs. The experiments were conducted in our testing field.

TABLE II
TASK PACKAGE

Task type	Amount	Priority	Loading station	Unloading station
P-cool	2	1	#F1	#F2
resistance	4	2	#A1	#A2
capacity	8	3	#C1	#C2
p-shell	5	4	#D1	#D2
p-chip	6	5	#E1	#E2
circuit-b	1	6	#B1	#B2

Scheduler control system assigned the P-cool task which owns highest priority to the first AGV. Then system assigned task to AGV according the priority of the task. Then AGV which is nearest to the task loading station will be firstly considered to be dispatched to the task.

In experiments, the times needed decreased obviously while the number of AGV increased. Even the number of AGVs to perform the task package was as high as 18, no situation of conflicts or deadlocks has occurred. The results validated that our proposed scheduler control systems can solve the conflict and deadlock problems successfully.

V. CONCLUSION AND FUTURE WORKS

In this paper, a new semaphore-based traffic control model has been proposed to resolve problems of conflicts and deadlocks in application of multi-AGV system. The effect of the scheduler control system using this model to resolve conflicts and deadlocks has been validated by experiments.

In future, we will apply the scheduler control system to factory with large number of AGVs, and further improve the system's efficiency.

REFERENCES

[1] Ling, Q., Wen-Jing, H., Shell-Ying Huang, Han Wang, "Scheduling and routing algorithms for AGVs: A survey", International Journal of Production Research, Vol. 40, Iss. 3, 2002.

[2] Rinehart, M., Dahleh, M.A., "The Value of Side Information in Shortest Path Optimization," Automatic Control, IEEE Transactions on , vol.56, no.9, pp.2038,2049, Sept. 2011.

[3] Reveliotis, S.A., Roszkowska, E., "On the Complexity of Maximally Permissive Deadlock Avoidance in Multi-Vehicle Traffic Systems," Automatic Control, IEEE Transactions on , vol.55, no.7, pp.1646,1651, July 2010.

[4] Naiqi, W., Mengchu, Zhou., "Shortest Routing of Bidirectional Automated Guided Vehicles Avoiding Deadlock and Blocking", IEEE/ASME Transactions on Mechatronics, vol.12, no.1, pp.63-72, Feb. 2007.

[5] Jing, W., "Fully distributed traffic control strategies for many-AGV systems," Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on , vol., no., pp.1199-1204 vol.3, 3-5 Nov 1991.

[6] Bobanac, V., Bogdan, S., "Routing and scheduling in Multi-AGV systems based on dynamic banker algorithm," Control and Automation, 2008 16th Mediterranean Conference on , vol., no., pp.1168-1173, 25-27 June 2008.

[7] Junghoon, L., Beomhee, L., Myoung Hwan Choi, "A real-time traffic control scheme of multiple AGV systems for collision free minimum time motion: a routing table approach," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol.28, no.3, pp.347-358, May 1998.

[8] Huang, J., Palekar, "A labeling algorithm forthe navigation of automated guided vehicles," Trans. ASMEJ. Eng.Ind., Vol.115, pp. 315-321, 1993.

[9] Sun, Q., Yang, Q., Yan, W., "On the Design of AGVs:Modeling, Path Planning and Localization," IEEE International Conference on Mechantronics and Automation, pp.1515-1520, Aug. 2011.

[10] Smolicrocak, N., Bogdan, S., Kovacic, Z., Petrovic, T., "Time Windows Based Dynamic Routing in Multi-AGV Systems," IEEE Transactions on Automation Science and Engineering, vol.7, no.1, pp.151-155, Jan. 2010.

[11] Peihuang, L., Xing, W., Jiarong, W., "Path Planning and Control for Multiple AGVs Based on Improved Two-Stage Traffic Scheduling", Int. J. of Automation Technology, vol. 3, no. 2, 2009.

[12] Andrew, S., "Modern Operating Systems", Prentice Hall; 3 edition (December, 2007).