

УДК 004.451

О.С. САВЕНКО, С.В. МОСТОВИЙ

Хмельницький національний університет, Україна

МОДЕЛЬ ПРОГНОЗУВАННЯ СТАНУ ПРОЦЕСІВ В КОМП'ЮТЕРНІЙ СИСТЕМІ

В роботі запропоновано перелік параметрів процесу, які враховуються при побудові його сигнатури. На основі визначених параметрів процесу запропонована структура сигнатури процесу та алгоритм її побудови в комп'ютерній системі. На основі сигнатури запропоновано модель та укрупнений алгоритм прогнозування стану процесу в комп'ютерній системі.

процес, взаємоблокування, параметри процесу, сигнатура процесу, прогнозування стану процесу

Вступ

Широке застосування персональних комп'ютерів (ПК) для вирішення масштабних і об'ємних задач вимагає безвідмовної роботи процесів, що виконуються на них [1, 2]. Оскільки відомі на сьогодні методи та засоби не повністю вирішують таку задачу [2, 3], постає необхідність у розробці нових підходів до її розв'язання.

Постановка задачі. -Аналіз відомих методів та засобів вирішення проблеми взаємоблокування процесів в комп'ютерній системі (КС) показав, що дані методи мають ряд недоліків: блокування роботи операційної системи (ОС), наявність циклів активного очікування, складність програмної реалізації для багатьох процесів, необхідність використання спеціалізованої команди процесора [2]. Тому необхідно розробити модель прогнозування стану процесів в КС, яка базується на використанні сигнатур процесів і дає можливість виділити підмножину сигнатур процесів, що наближаються до стану взаємоблокування.

1. Параметри процесу

Визначимо параметри процесу, що є необхідними для побудови його сигнатури. Для цього розглянемо множину характеристик процесу, якими він володіє у сучасних операційних системах.

Операційна система створює процес, коли кори-

стувач запускає програму на виконання. Процес і програма, що виконується в даному процесі, не отожднюються. Програмою є машинний код і початкові дані, що містяться у виконуваному файлі на диску або скопійовані ОС для виконання в оперативну пам'ять. Для кожного процесу ОС створює додатковий набір даних, що називається середовищем виконання процесу або контекстом процесу. Дані із цього набору називаються атрибутами процесу [4, 5] – ознаками, що відрізняють один процес від іншого. Вони можуть бути як сталими, так і змінюватись протягом життєвого циклу процесу (наприклад, ідентифікатор процесу – сталий параметр, пріоритет процесу – динамічний параметр).

До найважливіших атрибутів процесу віднесемо наступні:

- ідентифікатор процесу (PID – Process Identifier). Ідентифікатор процесу є цілим додатнім числом. Кожний процес в системі має унікальний ідентифікатор. ОС керує процесами, використовуючи ідентифікатор процесу;
- ідентифікатор батьківського процесу (PPID). Цей атрибут процес отримує під час свого запуску і може використовувати його для отримання інформації про стан батьківського процесу або для передачі йому сигналу;
- реальний і ефективний ідентифікатори користувача і групи (UID, GID, EUID і EGID). Реальні

ідентифікатори співпадають з ідентифікаторами користувача, що запустив процес, і групи, до якої він належить. Реальні ідентифікатори наслідуються дочірнім процесом і не можуть бути змінені. Права доступу процесу до ресурсів ЕОМ визначаються ефективними ідентифікаторами;

- відкриті файли. Крім стандартних файлів вводу, виводу і помилок процес може відкрити інші файли. Дочірній процес наслідує всі файли, відкриті батьківським процесом;

- пріоритет і відносний пріоритет. Пріоритет процесу – змінна величина, яка обраховується ОС в момент вибору процесу для виконання. Він змінюється в залежності від наступних факторів: відносний пріоритет процесу, час очікування запуску, поточний стан процесу та ін.;

- поточний каталог. Кожний процес має свій поточний та робочий каталоги;

- час виконання. Для процесу визначений користувацький, системний і реальний час виконання процесу. Користувацький час – це час, витрачений ЦП на виконання коду програми. Системний час – це час, витрачений ЦП на обробку системних викликів, операцій вводу/виводу, пересилки даних. Реальний час – це час, що пройшов від моменту запуску програми;

- змінні оточення;

- розмір програми – об'єм пам'яті, яку займає процес. Кожний процес характеризується значеннями віртуального розміру і розміру резидентної частини;

Перераховані параметри є спільними для різних операційних систем. Проте цей перелік може бути доповнений деякими параметрами, що є важливими для процесу у конкретній ОС.

2. Сигнатура процесу

Для вирішення задачі прогнозування стану процесу представимо його набором характеристик (сигнатурою).

Визначення 1. Сигнатурою процесу назвемо сукупність його характеристик, яка однозначно ідентифікує поведінку процесу у комп'ютерній системі.

До характеристик процесу, що формують сигнатуру, віднесемо ідентифікатор процесу, ідентифікатор батьківського процесу, ідентифікатор користувача якому належить процес, пріоритет процесу, квоти процесу (кількість пам'яті і процесорний час доступні процесу), дескриптори відкритих процесом файлів [5, 6].

Подамо сигнатуру процесу у наступному вигляді (рис. 1):

- 1) ідентифікатор процесу (x_1 біт);
- 2) ідентифікатор батьківського процесу (x_2-x_1 біт);
- 3) ідентифікатор користувача (x_3-x_2 біт);
- 4) пріоритет процесу (x_4-x_3 біт);
- 5) кількість дескрипторів файлів, що використовуються процесом (x_5-x_4 біт);
- 6) обсяг віртуальної пам'яті, що використовує процес (x_6-x_5 біт);
- 7) час виконання процесу (x_7-x_6 біт);
- 8) додаткові параметри процесу (x_n-x_7 біт).

Довжина сигнатури буде залежати від кількості додаткових параметрів, які будуть включені в неї, та від архітектурних особливостей певної ОС.

Для відображення процесів та відповідних їм сигнатур подамо їх у наступній формі: (I, S, T) , де I – ім'я процесу, S – побудована сигнатура для даного процесу, T – час побудови або останньої модифікації сигнатури для даного процесу.

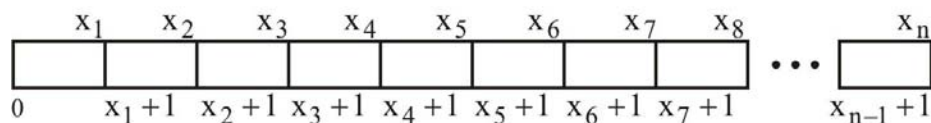


Рис. 1. Представлення сигнатури у машинному форматі

3. Алгоритм побудови сигнатури процесу

Представимо процес побудови сигнатури процесу наступним алгоритмом (рис. 2):

Крок 1. Перевірка, чи опрацьована уся множина процесів. Якщо так, то перехід до кроку 8, інакше – перехід до кроку 2.

Крок 2. Перевірка, чи має процес раніше сформовану сигнатуру. Якщо так, то перехід до кроку 6, інакше перехід до кроку 3.

Крок 3. Отримання інформації про процес, яка необхідна для формування його сигнатури.

Крок 4. Формування сигнатури процесу. На цьому кроці відбувається перетворення значень необхідних параметрів процесу до заданого вигляду і утворення його сигнатури.

Крок 5. Додавання створеної сигнатури до мно-

жини сигнатур процесів. Перехід до кроку 1.

Крок 6. Перевірка, чи змінилися значення складових сигнатури процесу. Якщо так, то перехід до кроку 7, інакше перехід до кроку 1.

Крок 7. Модифікація сигнатури процесу згідно поточних значень складових та перехід до кроку 1.

Крок 8. Збереження поточної множини сигнатур процесів.

4. Приклад побудови сигнатури процесу

Розглянемо на прикладі реальні процеси в ОС сімейства Windows (рис. 3) та Linux (рис. 4).

Включимо до складу сигнатури параметри [8, 9], якими володіє процес у даній ОС (табл. 1):

Результати побудови сигнатур для двох процесів в ОС Windows та для двох процесів в ОС Linux представимо у табл. 2.

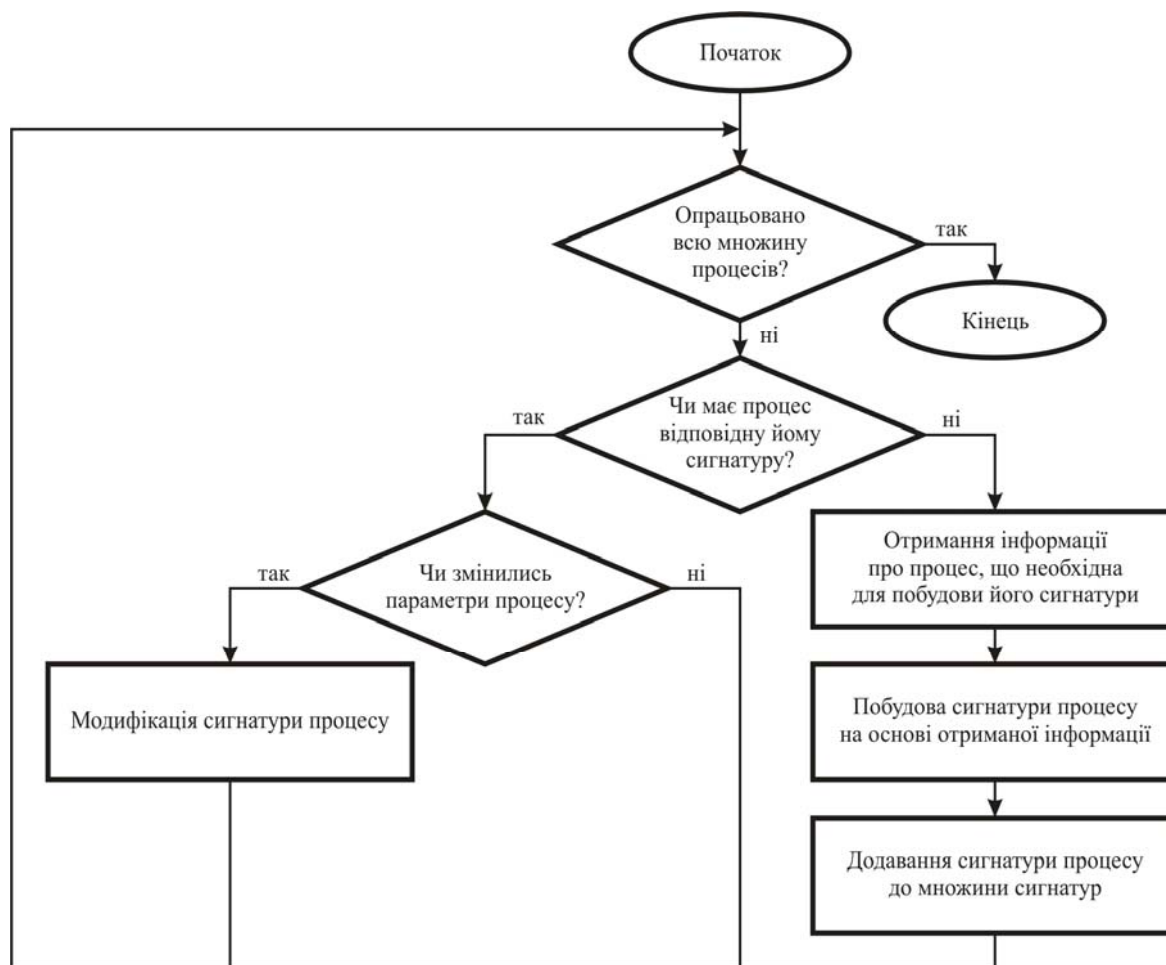


Рис. 2. Алгоритм побудови сигнатури процесу

Process	PID	CPU	User Name	Priority	Start Time	CPU Time	Virtual Size
System Idle Process	0	93.40	NT AUTHORITY...	0	n/a	3:07:57.506	0 K
Interrupts	n/a	n/a	n/a	0	n/a	0:00:53.827	0 K
DPCs	n/a	n/a	n/a	0	n/a	0:00:41.890	0 K
System	4		NT AUTHORITY...	8	n/a	0:00:28.140	1 880 K
smss.exe	496		NT AUTHORITY...	11	7:29:38 29.01.2008	0:00:00.050	3 804 K
csrss.exe	556	0.94	NT AUTHORITY...	13	7:29:44 29.01.2008	0:00:30.093	23 552 K
winlogon.exe	580		NT AUTHORITY...	13	7:29:45 29.01.2008	0:00:02.103	54 724 K
services.exe	624	0.94	NT AUTHORITY...	9	7:29:45 29.01.2008	0:01:04.733	20 876 K
svchost.exe	788		NT AUTHORITY...	8	7:29:47 29.01.2008	0:00:00.440	62 884 K
svchost.exe	832		NT AUTHORITY...	8	7:29:47 29.01.2008	0:00:00.620	40 804 K
svchost.exe	920		NT AUTHORITY...	8	7:29:47 29.01.2008	0:00:03.064	135 824 K
svchost.exe	964		NT AUTHORITY...	8	7:29:48 29.01.2008	0:00:00.090	31 544 K
svchost.exe	1012		NT AUTHORITY...	8	7:29:48 29.01.2008	0:00:01.682	40 180 K
spoolsv.exe	1192		NT AUTHORITY...	8	7:29:49 29.01.2008	0:00:01.381	53 316 K
CNAB4RP...	2152		NT AUTHORITY...	8	7:30:24 29.01.2008	0:00:00.070	22 820 K
nod32km.exe	1572		NT AUTHORITY...	8	7:29:52 29.01.2008	0:00:05.267	69 444 K
mysvc32.exe	1604		NT AUTHORITY...	8	7:29:52 29.01.2008	0:00:00.660	23 352 K
LEXBCE5.EXE	2796		NT AUTHORITY...	8	8:17:59 29.01.2008	0:00:00.150	42 376 K
LEXFPS.EXE	3004		NT AUTHORITY...	8	8:17:59 29.01.2008	0:00:00.130	46 216 K
lsass.exe	636		NT AUTHORITY...	9	7:29:46 29.01.2008	0:00:02.433	43 708 K
explorer.exe	1396		SAGITTARIUS...	8	7:29:50 29.01.2008	0:00:42.110	76 912 K
soundman.exe	1936		SAGITTARIUS...	8	7:29:56 29.01.2008	0:00:00.110	32 756 K
nod32kui.exe	136		SAGITTARIUS...	8	7:29:57 29.01.2008	0:00:02.082	34 924 K
DRIVES~1.EXE	216		SAGITTARIUS...	8	7:29:58 29.01.2008	0:00:00.220	32 484 K
ctfmon.exe	344		SAGITTARIUS...	8	7:30:00 29.01.2008	0:00:00.370	30 052 K
mandl.exe	408		SAGITTARIUS...	8	7:30:00 29.01.2008	0:00:03.645	35 320 K
ServUTray.exe	200		SAGITTARIUS...	8	7:30:01 29.01.2008	0:00:00.130	29 320 K
CNAB4LAK.EXE	1036		SAGITTARIUS...	8	7:30:05 29.01.2008	0:00:00.060	25 460 K
C7XRCli.exe	1132		SAGITTARIUS...	8	7:30:05 29.01.2008	0:00:00.891	34 660 K
TOTALCMD.EXE	2648		SAGITTARIUS...	8	7:31:02 29.01.2008	0:00:23.764	62 404 K
proccp.exe	1128	4.72	SAGITTARIUS...	13	9:22:59 29.01.2008	0:04:41.835	94 980 K
WINWORD.EXE	3884		SAGITTARIUS...	8	8:19:25 29.01.2008	0:01:34.586	182 376 K
calc.exe	2448		SAGITTARIUS...	8	9:17:07 29.01.2008	0:00:00.650	25 112 K

Рис. 3. Активні процеси в ОС сімейства Windows

PID	USER	PR	NI	VRT	RES	SHR	S	PPID	UID	P	SWAP	TIME	COMMAND
3805	beaglein	39	19	39144	17m	8308	R	3804	106	0	20m	4:05	beagle-build-in
3718	Saggitar	16	0	65204	15m	12m	S	1	1000	0	47m	0:34	ksysguard
2635	root	15	0	29796	20m	4668	S	2572	0	0	8432	0:09	Xorg
3719	Saggitar	15	0	4680	1728	1100	S	3718	1000	0	2952	0:10	ksysguardd
3848	Saggitar	15	0	31496	14m	12m	S	3293	1000	0	16m	0:00	ksnapshot
427	root	10	-5	0	0	0	S	2	0	0	0	0:00	scsi_ah_1
3447	Saggitar	15	0	37424	16m	12m	S	1	1000	0	20m	0:03	kicker
3742	Saggitar	15	0	33060	13m	10m	R	3293	1000	0	18m	0:02	konsole
1	root	15	0	744	288	240	S	0	0	0	456	0:00	init
2	root	11	-5	0	0	0	S	0	0	0	0	0:00	kthreadd
3	root	RT	-5	0	0	0	S	2	0	0	0	0:00	migration/0
4	root	34	19	0	0	0	S	2	0	0	0	0:00	ksoftirqd/0
5	root	10	-5	0	0	0	S	2	0	0	0	0:00	events/0
6	root	18	-5	0	0	0	S	2	0	0	0	0:00	khelper
25	root	10	-5	0	0	0	S	2	0	0	0	0:00	kblockd/0
26	root	20	-5	0	0	0	S	2	0	0	0	0:00	kacpid
27	root	10	-5	0	0	0	S	2	0	0	0	0:00	kacpid_notify
110	root	20	-5	0	0	0	S	2	0	0	0	0:00	cqueue/0
111	root	10	-5	0	0	0	S	2	0	0	0	0:00	kseriod
128	root	25	0	0	0	0	S	2	0	0	0	0:00	pdflush
129	root	15	0	0	0	0	S	2	0	0	0	0:00	pdflush
130	root	10	-5	0	0	0	S	2	0	0	0	0:00	kswapd0
131	root	20	-5	0	0	0	S	2	0	0	0	0:00	aio/0
362	root	15	-5	0	0	0	S	2	0	0	0	0:00	kpsmoused
372	root	11	-5	0	0	0	S	2	0	0	0	0:00	kondemand/0
419	root	10	-5	0	0	0	S	2	0	0	0	0:00	ata/0
420	root	13	-5	0	0	0	S	2	0	0	0	0:00	ata_aux
426	root	12	-5	0	0	0	S	2	0	0	0	0:00	scsi_ah_0
612	root	13	-5	0	0	0	S	2	0	0	0	0:00	ksuspend_usbd
613	root	11	-5	0	0	0	S	2	0	0	0	0:00	khubb
1010	root	10	-5	0	0	0	S	2	0	0	0	0:00	kjournald
1072	root	21	-4	2240	1000	376	S	1	0	0	1240	0:00	udev
1312	root	19	-5	0	0	0	S	2	0	0	0	0:00	scsi_ah_2
1314	root	10	-5	0	0	0	S	2	0	0	0	0:00	usb-storage
1317	root	19	-5	0	0	0	S	2	0	0	0	0:00	kgameportd
1814	root	18	0	3652	940	552	S	1	0	0	2712	0:00	mount.ntfs-3g

Рис. 4. Активні процеси в ОС сімейства Linux

Таблиця 1

Параметри процесу, що входять до складу сигнатури

№ п/п	Параметр	ОС Windows		ОС Linux	
		Використання	Кількість бітів	Використання	Кількість бітів
1	ідентифікатор процесу	+	14	+	14
2	ідентифікатор батьківського процесу	+	14	+	14
3	ідентифікатор користувача	+	4	+	10
4	пріоритет процесу	+	4	+	8
5	кількість дескрипторів файлів, що використовуються процесом	+	8	+	8
6	обсяг віртуальної пам'яті, що використовує процес	+	30	+	32
7	час виконання процесу	+	30	+	32
8	додатковий пріоритет процесу	+	4	+	8
9	стан процесу	-	-	+	2

Таблиця 2

Процеси та відповідні їм побудовані сигнатури

Ім'я процесу	ОС	Система числення	Сигнатура процесу	Час створення
nod32kui.exe (антивірусна система NOD32)	Windows XP	двійкова	00000010001000 00010101110100 0011 1000 01000110 00000000000000001000100001101100 0000000000000000000010000100010 1001	14:15:07
		шістнадцяткова	022057438460000886C000008229	
TOTALCMD.EXE (файловий менеджер)	Windows XP	двійкова	00101001011000 00010101110100 0011 1000 10101000 00000000000000001111001111000100 000000000000000000101110011010100 1000	14:15:07
		шістнадцяткова	296057438A80000F3C400005CD48	
ksysguard (менеджер процесів)	Linux SUSE 10.3	двійкова	00111010000110 00000000000001 1111101000 00010000 00000001 0000000000000000111111010110100 0000000000000000101110011010100 00000000 01	15:06:45
		шістнадцяткова	3A18001FA040040003FAD00001735001	
Xorg (графічна система)	Linux SUSE 10.3	двійкова	00101001001011 00101000001100 0000000000 00001111 00000101 0000000000000000111010001100100 00000000000000000000100000100010 00000001 01	15:06:45
		шістнадцяткова	292CA0C0003C180001D1900000208805	

5. Модель процесу прогнозування стану процесу в комп'ютерній системі

Подамо процес прогнозування стану процесів в КС наступною моделлю (2):

$$M = \langle A, S, D, P, R \rangle \quad (2)$$

де A - множина сигнатур процесів, що виконуються в КС у даний момент; S – впорядкована послідовність [7] характеристик комп'ютерної системи (загальний обсяг оперативної пам'яті, зовнішньої пам'яті, обсяг вільної в даний момент пам'яті, кількість периферійних пристроїв); D – підмножина сигнатур процесів, що знаходяться у стані, наближеному до стану взаємоблокування; P – множина правил, на основі яких проводиться розбиття множини сигнатур працюючих процесів на дві підмножини: підмножину сигнатур процесів, що знаходяться у стані, наближеному до стану взаємоблокування, та підмножину сигнатур процесів, які не досягли цього стану; R – вектор ймовірностей переходу у стан блокування процесів із підмножини D .

Для забезпечення процесу прогнозування стану

процесів в КС включимо до моделі наступні структурні частини (рис. 5):

- підсистема виявлення зміни стану процесу – призначена для контролю за поточними параметрами процесів, що вже присутні в КС, та запуском нових процесів;
- підсистема визначення характеристик КС – призначена для контролю за зміною основних характеристик КС, що суттєво впливають на наближення процесів до стану взаємоблокування;
- підсистема побудови сигнатури процесу – призначена для побудови сигнатури процесу, що виконує перехід у стан готовності і ще не має сигнатури, та модифікації сигнатури процесів, що змінили значення своїх параметрів та мають відповідні їм сигнатури;
- множина сигнатур процесів, що присутні у КС – призначена для зберігання сигнатур процесів, які в даний момент виконуються на КС;
- множина правил взаємодії сигнатур процесів містить правила, що визначають поведінку процесів за їх сигнатурами;

– підсистема аналізу та логічного висновку – за допомогою множини правил взаємодії сигнатур процесів виділяє із множини сигнатур всіх процесів підмножину сигнатур процесів, що наближаються до стану взаємоблокування, і робить висновок про можливість продовження роботи цих процесів.

Алгоритм визначення стану процесів в КС за та-

кою моделлю буде таким (рис. 6): на вхід системи поступає процес, що претендує на виконання в даній КС. Підсистема побудови сигнатури процесу отримує та аналізує дані про процес і виконує побудову його сигнатури (або модифікацію сигнатури, якщо процес вже мав сигнатуру). Побудована сигнатура додається до множини сигнатур процесів.



Рис. 5. Модель прогнозування стану процесів в комп'ютерній системі

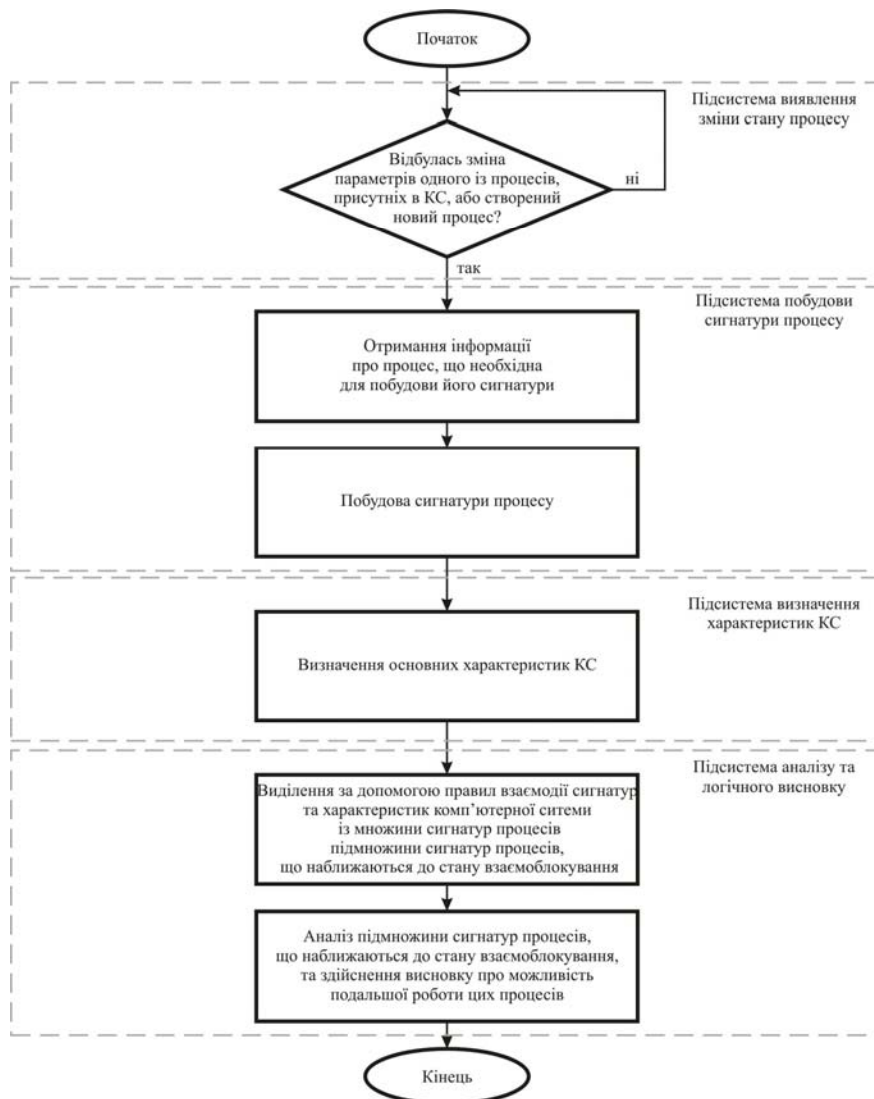


Рис. 6. Укрупнений алгоритм визначення стану процесів в КС

Підсистема аналізу та логічного висновку на основі множини правил взаємодії сигнатур процесів проводить розбиття множини сигнатур процесів, присутніх в КС на дві підмножини: підмножину сигнатур процесів, що знаходяться у стані, наближеному до стану взаємоблокування, та підмножину сигнатур процесів, які не досягли цього стану. Далі дана підсистема проводить аналіз підмножини сигнатур процесів, що наближаються до стану взаємоблокування та визначає ймовірність виникнення ситуації взаємоблокування. Після цього формується висновок про можливість подальшого виконання таких процесів.

Висновки

В роботі визначено параметри процесу, які включені в структуру сигнатури процесу, розроблено алгоритм її побудови. Запропоновано модель прогнозування стану процесів в КС, яка дає можливість виділити підмножину сигнатур процесів, що наближаються до стану взаємоблокування, та зробити висновок про можливість подальшої роботи цих процесів. На відміну від відомих підходів запропонована модель дозволяє уникнути наявності циклів активного очікування, блокування роботи ОС та необхідності використання спеціалізованої команди процесора. Це необхідно для розробки методу та засобів, які дозволять попереджувати про можливість взаємоблокування процесів в комп'ютерній системі.

Література

1. Савенко О.С., Мостовий С.В. Порівняльний аналіз програмних засобів тестування системних

ресурсів персональних комп'ютерів в локальній обчислювальній мережі // Матеріали II науково-практичної конференції "Образование и наука без границ-2006". – Днепропетровск: Наука и образование, 2006. – Т. 8. – С.115-118.

2. Савенко О.С., Кльоц Ю.П., Мостовий С.В. Дослідження та аналіз блокування процесів в комп'ютерній системі // Вісник ХНУ. – Хмельницький: ХНУ, 2007. – № 3, Т. 1. – С. 248-251.

3. Савенко О.С., Мостовий С.В. Модель побудови сигнатури процесу в комп'ютерній системі // Прогресивні інформаційні технології в науці та освіті. Збірник наукових праць. – Вінниця: Вінницький соціально-економічний інститут університету "Україна", 2007. – С. 58-63.

4. Хоар Ч. Взаимодействующие последовательные процессы: Пер. с англ. – М.: Мир, 1989. – 264 с.

5. Столлинге В. Операционные системы. – СПб: Вильямс, 2002. – 848 с.

6. Дейтел Х.М., Дейтел П.Дж., Чофнес Д.Р. Операционные системы. Т. 1: Основы и принципы: Пер. с англ. – М: Бином, 2006. – 753 с.

7. Сигорский В.П. Математический аппарат инженера. – К.: Техніка, 1975. – 768 с.

8. Роберт Лав. Разработка ядра Linux. – СПб.: Вильямс, 2006. – 448 с.

9. Стахнов А. Linux. – СПб.: BHV-Санкт-Петербург, 2005. – 944 с.

Надійшла до редакції 15.01.2008

Рецензент: д-р техн. наук, проф. О.М. Романкевич, Національний технічний університет України «КПІ», Київ.