А.Г. Габович

В.М. Єфимець

С.В. Зибін

Операційні системи

Методичні вказівки

Частина 1

Київ – 2006

Міністерство освіти і науки України
Державний університет інформаційно-комунікаційних технологій

А.Г. Габович

В.М. Єфимець

С.В. Зибін

Операційні системи

Методичні вказівки

до виконання курсового проекту студентами

Інституту заочного та дистанційного навчання спеціальності 7.091501

«Комп'ютерні системи та мережі»

Частина 1

Київ – 2006

УДК 681.3.066
ББК 32.973-018.2
Г-60

Затверджено на засіданні вченої ради навчально-наукового інституту захисту інформації (протокол № 3 від 16 жовтня 2006 р.)

Рецензент: кандидат технічних наук, доцент, Богуш В.М.

Операційні системи: Методичні вказівки до виконання курсового проекту / Автори: Габович А.Г., Єфимець В.М., Зибін С.В., - К.: ДУІКТ, 2006. 24 с.

Методичні вказівки містять вимоги до змісту та визначають порядок виконання курсового проекту.

Для студентів четвертого курсу ІЗДН та можуть бути використані студентами денної форми навчання.

1. Основні положення курсового проектування

Курсовий проект (КП) з дисципліни «Операційні системи» є самостійною роботою студента і виконується за індивідуальним завданням. Він має закріпити, узагальнити знання, які отримані студентами в період навчання. Крім того, в процесі курсового проектування студент має набути навиків з користування довідковим матеріалом та освоїти процес створення проектно-конструкторської документації відповідно до вимог стандартів.

Можна відмітити такі етапи проектування спеціалізованої операційної системи – розробка архітектури апаратного ядра та операційної системи (ОС).

На першому етапі розробляється архітектура апаратного ядра, визначається структура пакета завдань.

Другий етап пов'язаний з розробкою архітектури ОС.

Третій етап визначає алгоритми та програмування модулів ОС.

2. Тематика курсового проектування

Проектується архітектура спеціалізованої ОС для пакетної обробки завдань, завдань реального часу і т.д.

3. Структура курсового проекту

В КП виділяються аналітична та графічна частини.

3.1. Аналітична частина

В цій частині розроблюються такі етапи КП: архітектурний; алгоритмічний; програмний.

Архітектурний етап визначає:

- архітектуру апаратного ядра;
- архітектуру ОС.

Алгоритмічний етап визначає:

- алгоритми функціонування модулів ОС;
- схеми алгоритмів модулів ОС.

Програмний етап.

На цьому етапі розроблюються модулі ОС.

3.2. Графічна частина

В графічній частині приводяться ілюстровані матеріали, що супроводжують етапи аналітичної частини.

КП оформлюється у вигляді пояснювальної записки та супроводжуючих креслень.

3.3. Структура пояснювальної записки та перелік графічних робіт

Пояснювальна записка має об’єм 20-25 сторінок формату А4 та має таку структуру.

1. Титульний аркуш.
2. Технічне завдання.

3. Зміст.
4. Вступ. Аналітична частина. Список використаних джерел.

Графічні матеріали реалізуються на окремих аркушах формату А4.

У додатку 1 наведений приклад реалізації КП.

Додаток 1
Приклад титульного аркушу

Інститут заочного та дистанційного навчання

Спеціальність 7.091501 «Комп’ютерні системи та мережі»

Кафедра комплексних систем захисту інформації

КУРСОВИЙ ПРОЕКТ (РОБОТА)

з дисципліни «Операційні системи»

Тема: Спеціалізована операційна система

Виконав (ла)

Керівник:

Оцінка

Київ 2006

Технічне завдання
на курсовий проект по дисципліні «Операційні системи»
Термін виконання:_____

1. Спроектувати спеціалізовану ОС (фрагмент).
2. Задачі пакету: $F_1 = sh\frac{z_1}{z_2}; F_2 = (z_1 + z_2)\cdot z_3$.
3. Апаратне ядро:
   - адресність команди АК=2;
   - ємність ОП $N_{RAM} = 4$КСЛ;
   - команди та дані мають однакову довжину «n»;
   - пристрої вводу/виводу: клавіатура, принтер, маніпулятор типа «миша»;
   - допоміжна пам'ять - магнітний диск.
4. Операційна система:
   - пакетна багатозадачна;
   - стани ОС: блокований, виконання, чекання;
   - приорітети завдань: $F_1 = 1, F_2 = 2$.
5. Необхідно спроектувати:
   - схеми алгоритмів задач;
   - систему команд;
   - формат команди;
   - структуру апаратного ядра;
   - керуючі таблиці ОС;
   - загальні структури процесів;
   - дескриптори процесів.

ЗМІСТ

Вступ…….4

1. Система команд апаратного ядра……5

1.1. Схеми алгоритмів задач……..6

1.2. Програми задач F_1, F_2 у змістовній формі……..7

2. Структура апаратного ядра……..8

2.1. Визначення формату команд……..9

2.2. Структурна схема апаратного ядра……..10

3. Визначення часу рішення задач F_1, F_2 ……11

4. Формати завдань F_1, F_2 ……12

5. Визначення системних процесів……13

6. Розробка образу процесу……14

6.1. Структура керуючих таблиць…..……..15

6.2. Процеси користувачів…….……..16

Список використаних джерел……..17

Вступ

У вступі приводяться:

- стисла характеристика об’єкта розробки;
- функціональне призначення;
- особливість завдання проекту.

1. Система команд апаратного ядра

1.1. Схеми-алгоритми рішення задач

1.1.1. Задача $F_1 = sh(z_1 / z_2)$

Задача F_1 відноситься до класу науково-технічних. Для обчислення функції *sh* використовується степеневий ряд (див. дод.2):

$$sh(z) = z - \frac{z^3}{3!} + \frac{z^5}{5!} - ... = U_0 + \sum_{k=1}^{n} U_k \text{ , } (k=1,2,\ldots,n);$$

$$U_0 = z;\ U_k = U_{k-1} \cdot \left(\frac{z^2}{2k \cdot (2k+1)} \right).$$

Схема алгоритму рішення задачі F_1 приведена на рис. 1 та реалізована до арифметичних операцій (операторів присвоювання).

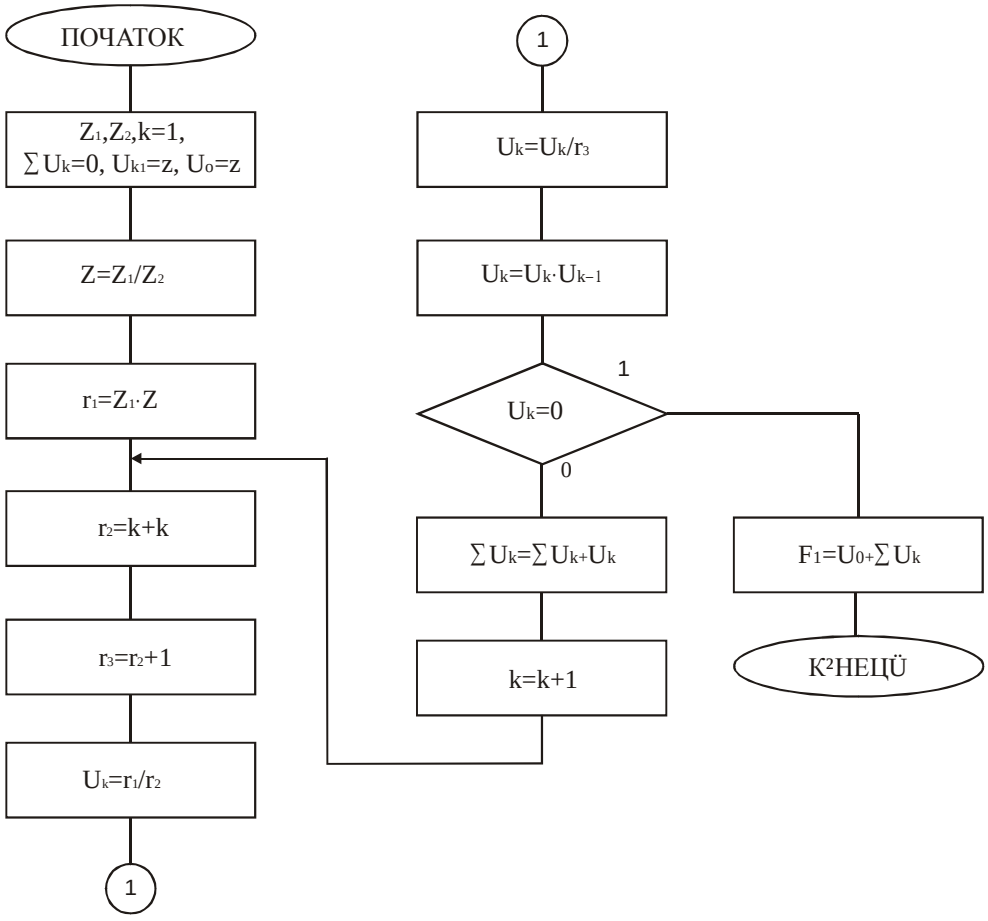


Рис.1. Схема алгоритму рішення задачі F_1

Із схеми алгоритму випливає наступний набір арифметичних операцій (перетворення) {+, *,/}.

1.1.2. Задача $F_2 = (z_1 + z_2) \cdot z_3$

Задача F_2 також відноситься до класу науково-технічних. Схема алгоритму рішення задачі F_2 представлена на рис. 2.

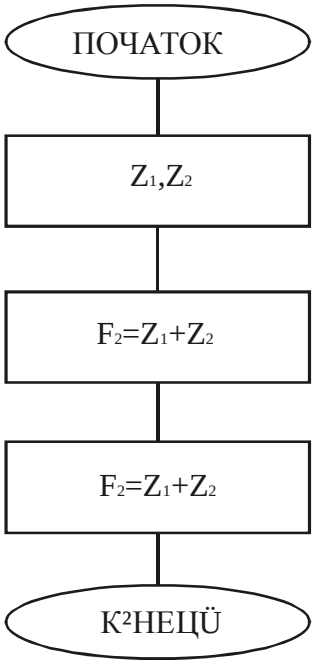


Рис. 2. Схема алгоритму рішення задачі F_2

Набір операцій перетворення {+,*}.

Об’єднаємо обидва набори. Отримаємо набір операцій перетворення апаратного ядра: {+,*,/}U{+,*}={+,*,/}.

1.2. Програми задач F_1 та F_2 у змістовній формі

Ці програми подібні до асемблерних та представлені в табл. 1 та 2. При написанні програм враховується адресність команди. В нашому прикладі АК=2. Вважаємо, що цикли команд перетворення не мають такту запису. Реалізація зберігання (запису) даних в пам’яті забезпечує команда ST. Задача F_1 (табл. 1) має циклічну частину. Вхід в цикл (оператор «S», вихід із циклу (оператор «15»). Кількість циклів приймаємо рівним п’яти. Програми в таблицях 1 та 2 відповідають схемам алгоритмів рис. 1 та рис. 2.

Програми та данні F_1, F_2 зберігаються у віртуальній пам'яті (ВП). ВП має сторінкову адресацію. Розмір сторінки – 0,5 Кслів.

Символьний запис команди: КОП, А1, А2, де КОП – код операції, А1, А2 – віртуальні адреси операндів.

Таблиця 1

Задача F_1

ВА	КОП	А1	А2	Пояснення
0	IN	D_0	-	Введення даних.
1	/	Z_1	Z_2	$R := Z_1 / Z_2$
2	ST	Z	-	$Z := R$
3	*	Z	Z	$R := Z * Z$
4	ST	Z		$Z := R$
5	+	K	K	$R := K + K$
6	ST	r_2		$r_2 := R$
7	+	r_2	<1>	$R := r_2 + <1>$
8	ST	r_3	-	$r_3 := R$
9	/	r_1	r_2	$R := r_1 / r_2$
10	ST	U_k	-	$U_k := R$
11	/	U_k	r_3	$R := U_k / r_3$
12	ST	U_k	-	$U_k := R$
13	*	U_k	U_{k-1}	$R := U_k * U_{k-1}$
14	ST	U_k	-	$U_k := R$
15	JZ	21	-	Умовний перехід по ознаці «нуль»
16	+	$\sum U_k$	U_k	$R := \sum U_k + U_k$
17	ST	$\sum U_k$	-	$\sum U_k := R$
18	+	K	<1>	$R := K + <1>$
19	ST	-	-	$K := R$
20	JMP	5	-	Безумовний перехід
21	+	U_0	$\sum U_k$	$R := U_0 + \sum U_k$
22	ST	F_1	-	$F_1 := R$
23	OU	D_1	-	Друк F_1
24	HLT	-	-	З
25	-	-	-	Z_1 - операнд
26	-	-	-	Z_2 - операнд
27	-	-	-	F_1 - операнд-результат
28	-	-	-	<1> - константа «1»
29	-	-	-	r_1
30	-	-	-	r_2
31	-	-	-	r_3
32	-	-	-	$\sum U_k$
33	-	-	-	Z
34	-	-	-	K

Поле ВА в табл. 1 рядки відображають команди та дані. Числа 0,1,.. є віртуальними адресами команд та даних.

В полі КОП представляються назви операцій: IN – операція вводу; ST – операція збереження; JZ – умовний перехід при нульовому значенні результату; JMP – безумовний перехід; OU – операція виводу; HLT – операція зупинки програми.

Складові частини задачі F_1 : програма; операнди { z_1, z_2 }; операнд-результат F_1; проміжні операнди $z, r_1, r_2, r_3, \sum U_k, U_0$; константи 0,1.

Змінній *R* відповідає вмісту регістра акумулятора АЛБ.

Список операцій необхідний для рішення задачі F_1

{+,*,/,ST,JZ,JMP,HLT}.

Ці операції виконуються в режимі користувача.

Для вводу даних з клавіатури та друку результату F_1 використовуються відповідно команди IN- ввід, OU – вивід, які є привілегійованими.

Таблиця 2

Задача F_2

ВА	КОП	А1	А2	Пояснення
1	+	Z_1	Z_2	$R := Z_1 + Z_2$
2	ST	F_2		$F_2 := R$
3	*	F_2	Z_3	$R := F_2 * Z_3$
4	ST	F_2	-	$F_2 := R$
5	OU	-	-	Друк F_2
6	HLT	-	-	Зупин
7	-	-	-	Z_1 - операнд
8	-	-	-	Z_2 - операнд
9	-	-	-	F_2 - операнд-результат

Складові частини задачі F_2 : програма (команди); операнди { z_1, z_2 }; операнд-результат F_2.

Список операцій необхідний для рішення задачі F_2:

{+,*,/,ST,HLT}.

Ці операції виконуються в режимі користувача.

Об’єднує списки операцій, отримуємо систему команд апаратного ядра {+,*,/,ST,JZ,JMP,HLT,JN,OU}. Кількість машинних операцій M=9.

2. Структура апаратного ядра

2.1. Визначення формату команди

Відповідно ТЗ команда має структуру:

КОП	А1	А2

2.1.1. Довжина поля КОП

При мінімальному кодуванні обчислюється за формулою:

$$n_{КОП} =]\log_2 M[=]\log_2 9[= 4 .$$

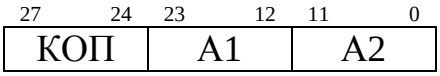
2.1.2. Довжина адресного поля A_i

$$n_{КОП} =]\log_2 N_{RAM}[=]\log_2 4K[=]\log_2 2 \cdot 2[= 12 .$$

2.1.3. Довжина коду команди

$$n = n_{КОП} + 2 \cdot n_{A_i} = 4 + 2 \cdot 12 = 28 \text{ біт}.$$

2.1.4. Формат команди



2.2. Структурна схема апаратного ядра

На рис. 3 представлена структурна схема центрального процесора CPU, основна пам'ять, пристрої вводу/виводу (ПВВ), а також зовнішня пам'ять – жорсткий диск. ПВВ та зовнішня пам'ять з'єднуються з CPU, через контролер, завдяки яким стандартні команди та сигнали від CPU перетворюються в індивідуальні для кожного ПВВ і навпаки.

У випадку використання віртуальної пам'яті для перетворення ВА у фізичні адреси (ФА) служить перетворювач ВА $\rightarrow$ ФА.

Основна пам'ять (ОП) має ємкість 4к28-розрядних слів та служить для зберігання програм та даних користувачів (пакетних завдань). Адресний простір (0÷4К-1). При сторінковій адресації вона поділяється на 40 сторінок (блоків) по 100 слів кожна.

Зовнішня пам'ять реалізована на магнітному диску.

Пристрій вводу – клавіатура. Служить для вводу завдань.

Пристрій виводу – принтер. Служить для друку результатів завдань.

Визначимо призначення регістрів CPU.

R(27:0) – регістр-акумулятор. Служить для короткочасного зберігання операнда, а також результата-операнда;

ПЛ(11:0) – програмний лічильник. Призначений для генерації адрес команд програм. Програмний код може розміщуватись в ОП довільно, тому що його розрядність буде

$$n_{ПЛ} = n_{A_i} = 12 біт .$$

RK(27:0) – регістр команд. Призначений для зберігання коду команди, яка вибрана з ОП.

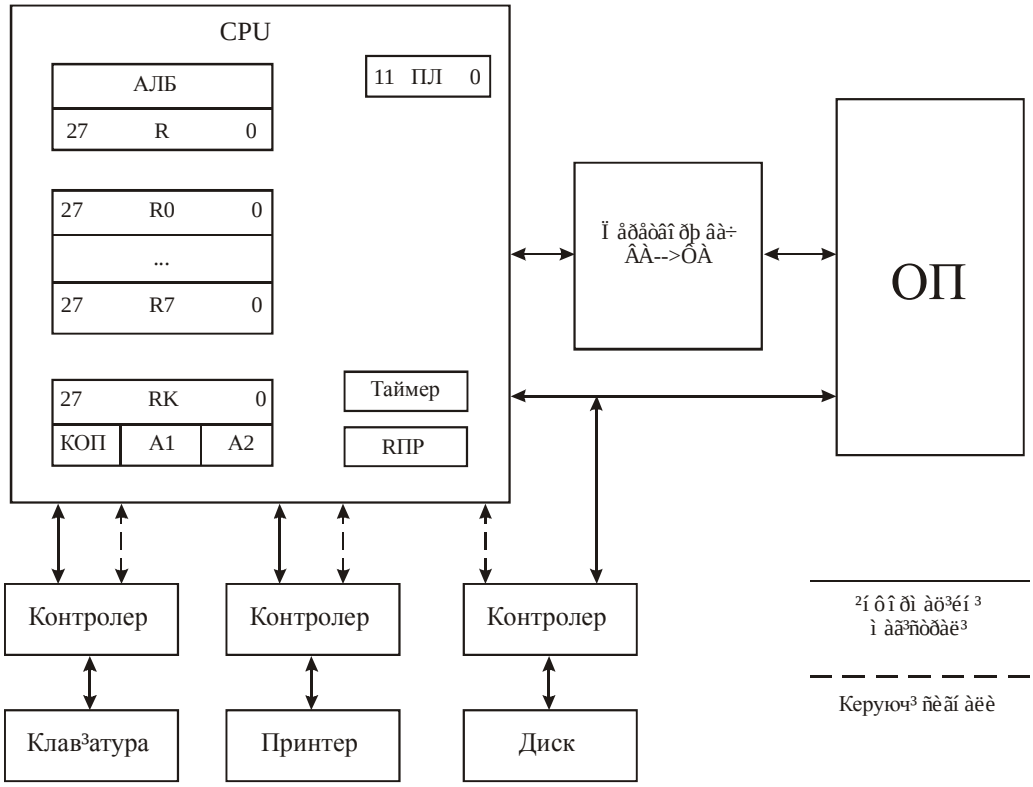


Рис. 3. Структура апаратного ядра

R0-R7 – регістри загального призначення.
RPSW – регістр стану процесора (програми). Має наступну структуру:

	27					21	20	0
RPSW	P	I	N	C	Z	V	Не використовується	

N,C,Z,V – біти ознак операцій. N – ознака від'ємного результату операції; C – біт переносу із старшого розряду; Z – ознака нульового результату; V – ознака переповнення розрядної сітки.

T – біт «пастка». Використовується для покомандного відлагодження програми.

$$T = \begin{cases} 0 - \textit{команди виконуються безперервно}; \\ 1 - \textit{після виконання кожної команди виконується переривання програми}. \end{cases}$$

I – біт маски переривання

$$I = \begin{cases} 0 - \textit{переривання заборонено}; \\ 1 - \textit{переривання дозволено}. \end{cases}$$

P – біт стану процесора

$$P = \begin{cases} 0 - \textit{режим користувача}; \\ 1 - \textit{режим привілегійований (ядра ОС)}. \end{cases}$$

РПР – регістр переривання. Фіксує запити переривань – внутрішні та зовнішні. Внутрішні запити породжуються особливими випадками, які виникають в програмі користувача, зовнішні запити – від ПВВ.

	21			4	3	2	1	0
РПР					V	Кл	Принт	Д

РПР (0)=Д – запит переривання дискової пам’яті; РПР(1)=ПРИНТ – запит переривання принтера; РПР(2)=Кл – запит переривання клавіатури, де є зовнішні переривання; РПР(3)=V – внутрішній запит переривання (переповнення розрядної сітки). Запити переривання «Кл», «ПРИНТ», «Д», якщо дозволені, викликають зміну стану CPU на привілегійований.

RS – регістр номера сторінки;

SP – вказівник стеку.

Таймер – лічильник часу. Функціонує як від’ємний лічильник. Початкове значення таймера, що встановлюється при ініціалізації процесу, кожним тактовим сигналом зменшується на одиницю. При досягненні нульового значення, виникає переривання від таймера. Опитування та встановлення таймеру виконується в привілегійованому режимі.

Перевірка регістра РПР з виділенням приорітетного запиту виконується апаратно.

3. Визначення часу реалізації завдань F_1, F_2

Час реалізації програми визначається за формулою:

$$t_p = \tau \cdot \sum_{i=1}^{k} m_i \cdot p_i + 1{,}5 \cdot \tau \cdot m_{пер} + \tau_{RAM} \left(\sum_{i=1}^{k} m_i \cdot m_{RAM_i} + m_{пер} \cdot m_{RAM_{пер}} \right), \text{ де}$$

τ – тривалість такту процесора;

τ_{RAM} - тривалість циклу основної пам’яті;

p_i - середнє число мікрокоманд тактів в *i*-й операції перетворення;

m_i - кількість разів виконання *i*-ї операції перетворення;

$m_{пер}$ - кількість разів виконання *i*-ї операції пересилки (LD – завантаження, ST – зберігання);

$1{,}5 \cdot \tau$ – тривалість операції пересилки;

m_{RAM_i} - число звернень до ОП *i*-ї команди перетворення;

$m_{RAM_{пер}}$ - число звернень до ОП команди пересилки (LD, ST).

Можливі значення p_i для операцій перетворення представлені в табл. 3.

Таблиця 3

Можливі значення p_i

Операція	+	*	/
p_i	1	10	13

Впливом тривалостей операцій JZ, JMP,HLT,JN,OU проігноруємо.

При визначенні параметру m_{RAM_i} необхідно враховувати адресність команди та її тип.

Для операцій перетворення (+,*,/) цикл команди із тактів: $T_{виб}$ - вибірка команди; $T_{ОП1}$ - вибірка 1-го операнду; $T_{ОП2}$ - вибірка 2-го операнду; $T_{опр}$ - виконання операції. Число звернень до ОП

$$m_{RAM_i}=3.$$

Цикл операції пересилки ST складається з тактів вибірки команди $T_{виб}$ та виконанні операції T_{ST}. Число звернень до ОП

$$m_{RAM_{пер}}=2.$$

Параметри $m_i, m_{пер}$ визначаються із табл. 1 та 2, і представлені в табл. 4.

Таблиця 4

Значення параметрів m_i, $m_{пер}$

Операція	F_1		F_2		m_{RAM}
	m_i	$m_{пер}$	m_i	$m_{пер}$	
+	21		1		3
*	6		1		3
/	11				3
ST		38		2	2

Час реалізації задач F_1, F_2.

$$t_{F_1}=\tau(2\cdot1+6\cdot10+11\cdot13)+1{,}5\tau\cdot38+\tau_{RAM}(2\cdot3+6\cdot3+11\cdot3+38\cdot2)=262\tau+133\tau_{RAM}$$
$$t_{F_2}=\tau(1\cdot1+1\cdot10)+1{,}5\tau\cdot2+\tau_{RAM}(1\cdot3+10\cdot3+2\cdot2)=14\tau+37\tau_{RAM}$$

Нехай τ=10 нс, τ_{RAM}=100 нс. Тоді

$$t_{F_1}=262\cdot10+133\cdot100\cong16000нс=16мкс.$$
$$t_{F_2}=14\cdot10+37\cdot100\cong4000нс=4мкс.$$

4. Формат завдань F_1, F_2

4.1. Структура завдання

В проекті виконується пакетна мультипрограмна обробка завдань користувача. Пакет складається із двох завдань F_1 та F_2. Завдання має структуру:

<Завдання>::=<блок керування>,<програма>,<END Id>.

<Блок керування>::= M |Id|T_{F_i}|N_{F_i}, де

- M – мультипрограмне завдання.
- Id – символьний ідентифікатор завдання.

- T_{F_i} – орієнтований час виконання.
- N_{F_i} – орієнтований об’єм виводу даних в рядках.

<Програма>. На початку блоку розміщується адреса віртуальної пам’яті користувача. Далі – послідовність команд програми та даних.
<END Id>. Кінець завдання.

4.2. Формати завдань користувачів

4.2.1. Завдання F_1

4.2.2. Завдання F_2

$$\left.\begin{matrix} M \\ F_1 \\ 16\cdot 10^{-6}c \\ 1 \end{matrix}\right\} < \textit{Блок кекрування} > \left\{\begin{matrix} M \\ F_2 \\ 4\cdot 10^{-6}c \\ 1 \end{matrix}\right.$$

$$\left.\begin{matrix} 0 \\ \textit{Таблиця}1 \end{matrix}\right\} < \quad \textit{Програма} \quad > \left\{\begin{matrix} 1 \\ \textit{Таблиця}2 \end{matrix}\right.$$

$$END_F_1\} < \textit{Кінець завдання} > \{END_F_2$$

5. Визначення системних процесів

Мета ОС – це обробка потоку завдань користувачів, які представлені у вигляді пакету. Це досягається багатозадачністю системних процесів та процесів користувача.

Кожне завдання проходить послідовно через наступні фази:

1. Ввідний спулінг. Завдання вводиться з клавіатури та пересилається на магнітний диск (МД).
2. Основна обробка. Програмний блок завдань F_i завантажується з МД в пам'ять користувача. Завдання F_i готове до виконання та перетворюється в процес «*i*». До свого повного завершення його стан буде багаторазово змінюватись:
   - готовність. Чекає призначення CPU;
   - виконання. Розвиток на CPU;
   - блокування. Очікування задовільнення запиту на введення/вивід.
3. Вивідний спулінг. Вивід результатів завдання.

У фазі основної обробки будуть одночасно знаходитися два завдання F_1 та F_2.

В ОС повинні бути передумовлені наступні процеси:

- JN_клав – введення з клавіатури в буфер вводу;
- JOB_J – створення дескриптора завдання та пересилка завдання на МД;
- LOADER_J – завантаження завдання користувача в пам'ять;
- JN_OU – обробка команд вводу/виводу апаратного ядра;
- READ_DISK – читання з МД рядка результатів в буфер виводу;

- PRINT – друк рядка.

Активізація ОС досягається за допомогою переривань в режимі користувача. Виклик програм переривання забезпечує планувальник процесів (розподільник CPU).

Основна задача ОС, яка проектується, є керування апаратними та програмними ресурсами: пам’яттю користувача; пам’яттю МД; програмними буферами; дескрипторами завдань та CPU.

На ОС покладається також завдання по збору статистичних даних по використанню ресурсів:

- апаратних – часу заняття CPU, пам’яті, МД;
- програмних – часу виконання завдання; об’єму пам’яті користувача і т.і.

Ця статистична інформація друкується після прогону пакета.

6. Розробка образу процесу

6.1. Структура керуючих таблиць ОС

Керуючі таблиці забезпечують ОС інформацією про стан ресурсів та процесів. Використовують чотири типи таблиць, які підтримуються ОС: пам'ять, пристрої вводу/виводу, файлів та процесів.

Структура керуючих таблиць представлена на рис. 4.

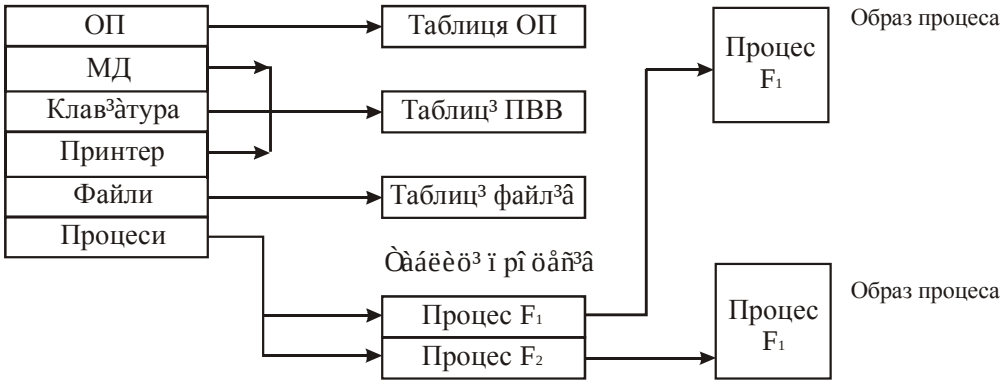


Рис. 4. Керуючі таблиці ОС

6.2. Процеси користувачів

Структура образів процесів у віртуальній пам'яті наведена на рис. 5.

Процес F_1		Процес F_2
Ідентифікатор	*Дескриптор процесу*	Ідентифікатор
Стан CPU		Стан CPU
Керуюча інформація процесу		Керуюча інформація процесу
Стек користувача		Стек користувача
АП користувача (табл. 1)		АП користувача (табл. 2

Рис. 5. Образи процесів F_1 та F_2

6.2.1. Дескриптор процесу F_1

Дескриптор процесу F_1 має наступну структуру параметрів.
Ідентифікація процесу.
Внутрішній ідентифікатор i=1.
Зовнішній ідентифікатор Id(i)= F_1.

6.2.2.Стан CPU

Загальні регістри.
R(0),…,R(7).
Адресні регістри:

- PC – програмний лічильник;
- RS – сторінковий регістр;
- SP – вказівник стеку користувача.

Системні регістри:

- RPSW – регістр стану процесу (програми);
- RПР – регістр переривань.

6.2.3. *Стан процесу*

Інформація для планування.

Статус процесу

$$\text{Ñòàòóñ[i]} = \begin{cases} \text{виконується;} \\ \text{готовий;} \\ \text{блокований.} \end{cases}$$

Пріоритет процесу користувача
Пріоритет[i]=1.
Час обороту процесу (початок/завершення)

$t[i] = t_{F_1} + \sum_{j=0}^{n} t_j \approx k \cdot \tau$, k=1,2,…,n=3,

де t_{F_1} - час прогону задачі F_1, t_j – процесорний такт, n – число використаних ресурсів.

Ресурси.

Визначається список ресурсів, які використовуються процесом F_1.

Процес[i]=A[i],

Де A[i] – вказівник списку ресурсів процесу F_1.

Елемент списку складається з двох полів: назви ресурсу та прапору ресурсу. Список ресурсів[i] представлений на рис. 6.

Клавіатура	U
Принтер	U
МД	U
CPU	U

$$U = \begin{cases} 0 - \textit{розподілени ресурс}; \\ 1 - \textit{монопольний ресурс}. \end{cases}$$

Рис. 6. Списки ресурсів [i]

Зв’язки процесів.

Вибір структури зв’язку процесів.

Можливі наступні структури: черга (FCFS – перший прийшов, перший обслугований); кільце (RR). Дисципліна FCFS реалізовується без витиснення процесу, дисципліна RR – з витисненням процесу. Витиснення базується на квантуванні часу, який виділяється процесу.

Список процесів структурно пов’язаних з процесом *i*.

Процес[i]=B[i],

де B[i] – вказівник списку взаємопов’язаних процесів. Список процесів представлений на рис. 7.

0	Процес[I]
1	JN
2	JOB[I]
3	JOB[J]
4	LOADER I
5	LOADER J
6	IN_OU
7	READ DISK
8	PRINT

Рис. 7

6.2.4. Керування пам’яттю

Визначається простір віртуальних адрес процесу. Вказуються верхня та нижня границі пам’яті. Можливе використання одного з двох способів розподілу пам'яті – розділами або сторінками. Необхідно лаконічно описати суть вибраного способу. Розмір пам'яті визначається з табл. 1.

6.3.1. Дескриптор процесу F_2

Дескриптор F_2 формується подібно до дескриптору F_1.

Список літератури.

1. Шеховцов В.А. Операційні системи. – К.: Видавнича група BHV, 2005. – 576 с.: іл.

2. Гордеев А.В. Операционные системы: Учебник для вузов. 2-е издание. – СпБ.: Питер, 2005. – 416 с.: ил.

3. Шоу А. Логическое проектирование операционных систем. – М.: Мир, 1981. – 360 с.

4. Столлингс В. Операционные системы, 4-е издание. М.: Вильямз, 2002. – 848 с.

5. Танненбаум Э. Современные операционные системы, 2-е изд. СПб: Питер, 2004. - 1024 с.

Додаток 2

1. $e^z; e^z = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + ... = U_0 + \sum_{k=1}^{n} U_k$, *(k=1,2,…,n),*

$U_0 = 1, U_k = U_{k-1} \cdot \frac{z}{k}$.

2. $\sin z; \sin z = z - \frac{z^3}{3!} + \frac{z^5}{5!} - ... = U_0 + \sum_{k=1}^{n} U_k$, *(k=1,2,…,n),*

$U_0 = z, U_k = U_{k-1} \cdot \left(- \frac{z^2}{2k(2k+1)} \right)$.

3. $\cos z; \cos z = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - ... = U_0 + \sum_{k=1}^{n} U_k$, *(k=1,2,…,n),*

$U_0 = 1, U_k = U_{k-1} \cdot \left(- \frac{z^2}{2k(2k-1)} \right)$.

4. $sh(z); sh(z) = z + \frac{z^3}{3!} + \frac{z^5}{5!} - ... = U_0 + \sum_{k=1}^{n} U_k$, *(k=1,2,…,n),*

$U_0 = z, U_k = U_{k-1} \cdot \frac{z^2}{2k(2k+1)}$.

5. $ch(z); ch(z) = 1 + \frac{z^2}{2!} + \frac{z^4}{4!} - ... = U_0 + \sum_{k=1}^{n} U_k$, *(k=1,2,…,n),*

$U_0 = 1, U_k = U_{k-1} \cdot \frac{z^2}{2k(2k-1)}$.

Варіанти завдань на КП по дисципліні «ОС»

№ варіанта	Задачі	Апаратне ядро				Операційна система		
		АК	N_{RAM}	ПВВ	МД	Режим	Пріоритет	Стан процесу
1	$F_1 = ch(z_1); F_2 = \frac{z_1}{z_2};$ $F_3 = (z_1 + z_2)$	1	1Ксл	Клавіатура, монітор	1	Пакетний	$F_1 > F_2 > F_3$	3
2	$F_1 = \frac{1}{ch(z_3)};$ $F_2 = (z_1 + \frac{z_2}{z_3})$	2	2Ксл	Клавіатура, монітор	2	Пакетний	$F_1 < F_2$	2
3	$F_1 = 1 + sh(z_2);$ $F_2 = \frac{z_1}{z_3}$	3	4 Ксл	Клавіатура, монітор	1	Пакетний	$F_1 > F_2$	5
4	$F_1 = \frac{1}{\sin(z_1)} + z_3;$ $F_2 = (z_2 + z_3)$	1	1 Ксл	Клавіатура, монітор, «миша»	2	Пакетний	$F_1 > F_2$	2
5	$F_1 = (z_1 + \frac{z_3}{z_2});$ $F_2 = \exp(z_2 / z_3)$	2	2 Ксл	Клавіатура, монітор, «миша»	1	Пакетний	$F_1 > F_2$	3
6	$F_1 = \sin(\frac{z_1}{z_3});$ $F_2 = \frac{1}{z_2 + z_3}$	3	4 Ксл	Клавіатура, монітор, «миша»	2	Пакетний	$F_1 > F_2$	5
7	$F_1 = z_1 \cdot z_3;$ $F_2 = \exp(z_1 + z_2)$	1	1 Ксл	Клавіатура, монітор, «миша»	1	Пакетний	$F_1 < F_2$	2
8	$F_1 = \frac{z_2}{z_3}; F_2 = \cos(z_3)$	2	2 Ксл	Клавіатура, «миша»	2	багато користувальний	$F_1 > F_2$	3
9	$F_1 = z_2 \cdot z_3;$ $F_2 = \frac{1}{sh(z_3)}$	3	4 Ксл	Клавіатура, «миша»	1	багато користувальний	$F_1 > F_2$	5
10	$F_1 = \frac{1}{\cos(z_1)};$ $F_2 = z_2 + z_3$	1	1 Ксл	Клавіатура, «миша»	2	багато користувальний	$F_1 > F_2$	2
11	$F_1 = (z_2 + \frac{z_1}{z_3});$ $F_2 = \frac{1}{sh(z_3)}$	2	2 Ксл	Клавіатура, «миша»	1	Пакетний	$F_1 < F_2$	3
12	$F_1 = z_1 \cdot z_3;$ $F_2 = \cos(\frac{z_1}{z_2})$	3	4 Ксл	Монітор, «миша»	2	Пакетний	$F_1 < F_2$	5

Продовження табл.

13	$F_1 = ch(z_2);$ $F_2 = (z_1 + \frac{z_3}{z_2})$	1	1 Ксл	Монітор, «миша»	1	Пакетний	$F_1 < F_2$	2
14	$F_1 = \frac{1}{sh(z_2)};$ $F_2 = z_1 + z_3$	2	2 Ксл	Монітор, «миша»	2	Пакетний	$F_1 < F_2$	3
15	$F_1 = \frac{1}{z_1 + z_2};$ $F_2 = \frac{1}{\sin(z_3)};$	3	4 Ксл	Монітор, «миша»	1	Пакетний	$F_1 > F_2$	5
16	$F_1 = \frac{1}{z_1 \cdot z_3};$ $F_2 = \sin(z_1 + z_2)$	1	1 Ксл	Клавіатура, монітор	2	Реаль-ного часу	$F_1 > F_2$	2
17	$F_1 = z_1 + z_2;$ $F_2 = \frac{1}{\sin(z_3)};$	2	2 Ксл	Клавіатура, монітор	1	Реаль-ного часу	$F_1 > F_2$	3
18	$F_1 = \frac{1}{z_1};$ $F_2 = sh(z_2 + z_3)$	3	4 Ксл	Клавіатура, монітор	2	Реаль-ного часу	$F_1 > F_2$	5
19	$F_1 = \frac{1}{(z_1 + z_2)};$ $F_2 = \exp(z_2 + z_3)$	1	8 Ксл	Клавіатура, монітор	1	Реаль-ного часу	$F_1 < F_2$	2
20	$F_1 = \frac{1}{z_3};$ $F_2 = \frac{1}{\exp(z_1)}$	2	1 Ксл	Монітор, «миша»	2	Пакетний	$F_1 < F_2$	3
21	$F_1 = \frac{1}{(z_1 + z_3)};$ $F_2 = \cos(z_2)$	3	2 Ксл	Монітор, «миша»	1		$F_1 < F_2$	5
22	$F_1 = z_3 + z_2;$ $F_2 = \frac{1}{\sin(z_1)};$	1	4 Ксл	Монітор, «миша»	2		$F_1 < F_2$	2
23	$F_1 = \frac{1}{(z_1 + z_3)};$ $F_2 = \frac{1}{\exp(z_1 / z_2)}$	2	8 Ксл	Монітор, «миша»	1		$F_1 > F_2$	3
24	$F_1 = \frac{z_1}{(z_1 + z_3)};$ $F_2 = \sin(\frac{z_1}{z_2})$	3	1 Ксл	Клавіатура, монітор	2		$F_1 > F_2$	5

Продовження табл.

25	$F_1 = z_3 \cdot z_2;$ $F_2 = sh(\frac{z_1}{z_2})$	1	2 Ксл	Клавіатура, монітор	1		$F_1 > F_2$	2

Навчально-методичне видання

Габович Артур Григорович

Єфимець Валентин Микитович

Зибін Сергій Вікторович

Операційні системи

Методичні вказівки

до виконання курсового проекту студентами

ІЗДН спеціальності 7.091501 «Комп'ютерні системи та мережі» та можуть буди корисні для студентів денної форми навчання спеціальності: 7.160102 Захист інформації з обмеженим доступом та автоматизація її обробки; 7.160103 Системи захисту від несанкціонованого доступу; 7.160104 Адміністративний менеджмент у сфері захисту інформації з обмеженим доступом; 7.160105 Захист інформації в комп'ютерних системах та мережах.

Частина 1

Технічний редактор Чирков Д.В.

Коректор Капустян М.В.

Підписано до друку 07.12.2006 р. Формат 64x84/16, папір офсетний
Друк офсетний
Умовн. друк. арк. 1,5. Обл. вид. арк. 1,2
Наклад 300 прим. Замовлення №258/11

Видавництво ДУІКТ
03110, Київ, вул. Солом'янська, 7.
Надруковано видавництвом ДУІКТ.
03110, Київ, вул.. Солом'янська, 7.
Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
Серія ДК №2539 від 26.06.2006 р.