9 лек х 1 бал = 9 балів

9 ЛР х 8 балів = 72

2 МКР х 8 балів = 16

Бонус 3 бали

Разом =100 балів

jup@ztu.edu.ua

**Література**

1. Алиев Т.М., Вигдоров Д.И., Кривошеев В.П. Системы отображения информациию М.: Высшая школа, 1988. 223 с.

2. Полякова Л.В., Лейн В.М. Отображение измерительной информациию Л.: Энергия, 1978. 144 с.

3. Безвесільна О.М., Таланчук П.М. Відлікові та реєструючи пристрої приладів. К.: ІСДО, 1993. 172 с.

4. Гук М. Процессоры Intel от 8086 до Pentium II. – СПб.: Питер, 1998.

5. Гук М. Энциклопедия аппаратных средств IBM PC.

6. Григорьев В.Л. Видеосистемы ПК фирмы IBM. – М.: Радио и связь, 1993.

7. Хант. МАТЛАБ с нуля

8. Дьяконов.

**Лабораторна робота № 1**

**ДОСЛІДЖЕННЯ ПРОЦЕСУ ВИВЕДЕННЯ ДАНИХ  
НА ПРИСТРІЙ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ**

**2.4.1. Мета роботи**

Порівняти різні варіанти організації виведення даних на пристрій відображення інформації (на екран монітору персонального комп'ютера).

Вивчити особливості програмування на мові Borland Pascal 7.0 операцій виведення даних на пристрій відображення інформації.

**2.4.2. Підготовка до роботи**

Вивчити різні варіанти організації виведення даних на пристрій відображення інформації [].

Виконати попередній аналіз початкових даних індивідуального варіанту (табл. 2.3).

Скласти програму на мові програмування Borland Pascal 7.0, що реалізує різні варіанти виведення даних на пристрій відображення інформації (п.п. 1 – 5 розділу “Виконання роботи”).

При складанні програми рекомендується використовувати додаток 5 даного методичного посібника.

**2.4.3. Виконання роботи**

1. Вивести на екран монітора персонального комп'ютера наступну графічну інформацію: прямокутник з координатами вершин і кольором заповнення у відповідності з індивідуальним варіантом завдання (табл. 2.3).

Таблиця 2.3

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант | Колір заповнення | Координати кутів прямокутника | |
|  |  | Лівий верхній | Правий нижній |
| 1 | Білий | 0.0 | 600.400 |
| 2 | Жовтий | 100.0 | 600.450 |
| 3 | Світло-сірий | 100.100 | 550.450 |
| 4 | Світло-зелений | 0.100 | 500.400 |
| 5 | Блакитний | 50.50 | 500.450 |
| 6 | Білий | 150.0 | 630.470 |
| 7 | Жовтий | 0.0 | 550.400 |
| 8 | Світло-сірий | 100.0 | 600.400 |
| 9 | Світло-зелений | 100.100 | 600.450 |
| 10 | Блакитний | 0.100 | 550.450 |
| 11 | Білий | 50.50 | 500.400 |
| 12 | Жовтий | 150.0 | 500.450 |
| 13 | Світло-сірий | 0.0 | 630.470 |
| 14 | Світло-зелений | 100.0 | 550.400 |
| 15 | Блакитний | 100.100 | 600.400 |
| 16 | Білий | 0.100 | 600.450 |
| 17 | Жовтий | 50.50 | 550.450 |
| 18 | Світло-сірий | 150.0 | 500.400 |
| 19 | Світло-зелений | 0.0 | 500.450 |
| 20 | Блакитний | 100.0 | 630.470 |

Закінчення таблиці 2.3

|  |  |  |  |
| --- | --- | --- | --- |
| Варіант | Колір заповнення | Координати кутів прямокутника | |
|  |  | Лівий верхній | Правий нижній |
| 21 | Білий | 100.100 | 550.400 |
| 22 | Жовтий | 0.100 | 600.400 |
| 23 | Світло-сірий | 50.50 | 600.450 |
| 24 | Світло-зелений | 150.0 | 550.450 |
| 25 | Блакитний | 0.0 | 500.400 |
| 26 | Білий | 100.0 | 500.450 |
| 27 | Жовтий | 100.100 | 630.470 |
| 28 | Світло-сірий | 0.100 | 550.400 |
| 29 | Світло-зелений | 50.50 | 630.400 |
| 30 | Блакитний | 150.0 | 500.470 |

2. Виведення графічної інформації виконати різними способами:

- використати стандартні процедури FillPoly або Bar мови Borland Pascal 7.0, що виконують заповнення замкнутого контуру;

- використати стандартну процедуру Line мови Borland Pascal 7.0 для заповнення прямокутника набором ліній;

- використати стандартну процедуру PutPixel мови Borland Pascal 7.0 для заповнення прямокутника набором точок;

- використати переривання Int 10h базової системи введення - виведення;

- використати безпосередній запис даних в відеопам’ять.

3. Порівняти час виведення графічної інформації на екран і складність програмування для всіх перерахованих вище способів виведення.

**2.4.4. Зміст звіту**

1. Найменування і мета роботи.

2. Програма, що реалізує різні варіанти виведення графічної інформації на екран монітору.

3. Результати визначення тимчасових характеристик виведення даних.

4. Висновки по роботі.

**2.4. 5. Контрольні запитання**

1. Поясніть, як впливає швидкість виведення графічної інформації на екран монітору на продуктивність ІОС.

2. Поясніть на прикладах взаємозв'язок структури відеосистемы персонального комп'ютера і принципів організації висновку інформації на екран монітору.

3. Назвіть можливі способи організації виведення графічної інформації на екран монітору.

4. Як можна прискорити виведення на екран групи сусідніх точок, що мають однаковий колір?

5. Як в програмі на мові програмування Borland Pascal 7.0 виконати виведення графічної інформації на екран монітору? Наведіть приклади.

**Лабораторна робота № 2**

**ДОСЛІДЖЕННЯ ПРОЦЕСУ ЗАПИСУ ДАНИХ  
З ОЗП НА ЖОРСТКИЙ ДИСК**

**2.3.1. Мета роботи**

Порівняти різні варіанти організації запису даних з ОЗП на жорсткий диск.

Вивчити особливості програмування на мові Borland Pascal 7.0 операцій запису даних на жорсткий диск ІОС.

**2.3.2. Підготовка до роботи**

Вивчити різні варіанти логічної організації даних, призначених для зберігання на зовнішніх запам’ятовуючих пристроях [ ].

Виконати попередній аналіз початкових даних індивідуального варіанту (табл. 2.2).

Скласти програму на мові програмування Borland Pascal 7.0, що реалізує різні варіанти запису даних на жорсткий диск ІОС (п.п. 1 – 3 розділу “Виконання роботи”).

При складанні програми рекомендується використовувати додаток 4 даного методичного посібника.

**2.3.3. Виконання роботи**

1. Виконати запис на жорсткий диск масиву даних, що зберігається в ОЗП у вигляді послідовності байтів (табл. 2.2). За одну операцію виведення даних виконувати запис на диск одного байта. Визначити час запису масиву даних.

Таблиця 2.2

|  |  |  |
| --- | --- | --- |
| Варіант | Довжина масиву в ОЗП, Мбайт | Довжина упорядкова­ної структури, Мбайт |
| 1 | 20 | 1 |
| 2 | 20 | 2 |
| 3 | 20 | 0.5 |
| 4 | 20 | 0.2 |
| 5 | 20 | 0.4 |
| 6 | 50 | 1 |
| 7 | 50 | 2 |
| 8 | 50 | 0.5 |
| 9 | 50 | 0.2 |
| 10 | 50 | 5 |
| 11 | 40 | 1 |
| 12 | 40 | 2 |
| 13 | 40 | 0.5 |

Закінчення таблиці 2.2

|  |  |  |
| --- | --- | --- |
| Варіант | Довжина масиву в ОЗП, Мбайт | Довжина упорядкова­ної структури, Мбайт |
| 14 | 40 | 0.2 |
| 15 | 40 | 5 |
| 16 | 30 | 1 |
| 17 | 30 | 2 |
| 18 | 30 | 0.5 |
| 19 | 30 | 0.2 |
| 20 | 30 | 5 |
| 21 | 35 | 1 |
| 22 | 35 | 2.5 |
| 23 | 35 | 0.5 |
| 24 | 35 | 0.2 |
| 25 | 35 | 5 |
| 26 | 60 | 1 |
| 27 | 60 | 2 |
| 28 | 60 | 0.5 |
| 29 | 60 | 0.2 |
| 30 | 60 | 5 |

2. Виконати запис на жорсткий диск масиву даних, що має в ОЗП вигляд упорядкованої структури (масив, елементи якого - дані типу запис). За одну операцію виведення даних виконувати запис на диск одного елемента масиву. Визначити час запису масиву даних.

3. Виконати запис на жорсткий диск масиву даних, що зберігається в ОЗП у вигляді послідовності байтів. Запис даних на диск організувати по блокам довжиною 512 байтів. Визначити час запису масиву даних.

4. Порівняти результати, отримані при виконанні  
п.п. 1 – 3.

**2.3.4. Зміст звіту**

1. Найменування і мета роботи.

2. Програма, що реалізує різні варіанти запису даних на жорсткий диск.

3. Результати визначення часу виконання запису даних.

4. Висновки по роботі.

**2.3.5. Контрольні запитання**

1. Пояснити, як впливає швидкість обміну інформацією з зовнішніми приладами на продуктивність ІОС.

2. Пояснити на прикладах взаємозв'язок особливостей підключення зовнішніх запам’ятовуючих приладів в персональному комп'ютері і принципів організації запису інформації на жорсткий диск.

3. Наведіть приклади можливої логічної структури даних, призначених для запису на жорсткий диск комп'ютера.

4. Якою повинна бути довжина блоку даних, що записуються на жорсткий диск за одну операцію виведення даних, для забезпечення максимальної швидкості обміну?

5. Як в програмі на мові програмування Borland Pascal 7.0 організувати запис даних на жорсткий диск? Наведіть приклади.

**Додаток 1**

**Процедури і функції мови програмування  
Borland Pascal 7.0, що рекомендуються до використання  
 в лабороторних роботах**

**1. Змінна:** vаr Test8087: Byte;

**Модуль:** System.

**Призначення:** зберігає результат автоматичної перевірки присутності співпроцесора в системі.

**Примітки:** Test8087 показує, будуть чи не бубудуть команди з плаваючою комою емулюються або справді виконуються. Наступні величини визначені для змінної Tesl8087: 0 - співпроцесора немає,1 - 8087, 2 - 80287, 3 - 80387 і вище.

В програмах, що не містять інструкцій співпроцесора, алгоритм розпізнання співпроцесора не включається в EXE - файл, і змінна Test8087 завжди містить нуль.

**2. Процедура:** procedure GetTime (var Hour, Minute, Second, Sec 100: Word);

**Модуль:** Dos, WinDos.

**Призначення:** вертає поточний час, встановлений в операційній системі.

**Параметри:** Hour – значення годин (0..23);

Minute – значення хвилин (0..59);

Second – значення секунд (0..59);

Sec100 – значення сотих часток секунди (0..99).

**3. Процедура:** procedure Line (Xl, Yl, X2, Y2: Integer);

**Модуль:** Graph.

**Призначення:** малює пряму лінію, що з'єднує дві вказані точки.

**Параметри:** (X1. Y1) і (X2. Y2) - координати двох точок, що необхідно з'єднати лінією.

**Примітка:** при зображенні прямої товщина лінії і її стиль задаються процедурою SetLineStyle, колір - процедурою SelColor.

**4. Процедура:** procedure LineTo (X, Y: Integer);

**Модуль:** Graph.

**Призначення:** малює пряму лінію з поточної позиції до вказаної точки.

**Параметри:** X, Y - координати точки.

**Примітка:**

MoveTo (100.100);

LineTo (200.200);

Еквівалентно

Line (100.100. 200.200);

Але перший метод вимагає більше часу і більшого обсягу пам'яті. Використовуйте процедуру LineTo тільки в тому випадку, коли поточний покажчик знаходиться в одному з кінців прямої, що потрібно намалювати.

**5. Процедура:** procedure MoveTo (X, Y: Integer);

**Призначення:** переміщує поточний покажчик в задану точку.

**Параметри:** X, Y - відносні (відносно поточної області перегляду) координати заданої точки (вказуються відносно поточного покажчика).

**Примітка:** Поточний покажчик аналогічний текстовому курсору, але не відображається на екрані.

**6. Процедура:** procedure PutPixel (X, Y: Integer; Pixel: Word);

**Модуль:** Graph.

**Призначення:** відображає на екрані точку означеного кольору.

**Параметри:** X, Y - координати точки; Pixel - її колір.

**7. Процедура:** procedure FillPoly (NumPoints: Word; var PolyPoints);

**Модуль:** Graph.

**Призначення:** малює на екрані многокутник і заповнює його, використовуючи поточний тип і колір заповнення.

**Параметри:** PolyPoints - нетипизований параметр, що містить координати всіх вершин многокутника;

NumPoints - кількість координат PolyPoints.

**8. Процедура:** procedure Bar (Xl, Yl, X2, Y2: Integer);

**Модуль:** Graph.

**Призначення:** малює прямокутник, використовуючи поточний тип і колір заповнення.

**Параметри:** XI, Y1. X2, Y2 - координати двох кутів прямокутника.

**Примітки:** використовуються поточний стиль і колір заповнення, задані за допомогою процедур SetFillStyle і SetFillPattem.

**9. Процедура:** procedure Write (F, VI [, V2,., VN]);

**Модуль:** System.

**Призначення:** записує змінну в елемент файлу.

**Параметри:** F - файлова змінна;

V1,., VN - змінні того же типу, що і елементи файлу.

**Примітки:** при кожному запису змінної поточна позиція в файлі переміщується до наступного елементу. Якщо поточна позиція файлу знаходиться в кінці файлу (тобто коли Eof (F) рівно True), то файл розширюється. При використанні директиви компілятора {$I-} функція lOResult буде вертати значення 0 в тому випадку, якщо операція завершилася успішно, і ненульовий код помилки в протилежному випадку.

**10. Процедура:** procedure BIockWrite (var F: file; var Buf, Count: Word [; var Result: Word]);

**Модуль:** System.

**Призначення:** записує в файл один або більше записів зі змінної.

**Параметри:** F – нетипизована файлова змінна, що описує файл, в який буде виконуватися запис;

Buf – будь-яка змінна; записи зчитуются з області пам'яті, що починається з першого байта, відведеного під Buf;

Count – кількість записів, які необхідно записати в файл;

Result – в цій змінній може вертатися число справді переданих у файл записів. Якщо цей параметр не заданий, то в випадку, коли число справді переданих в файл записів не співпадає з Count, то виникає помилка введення - виведення.

**Додаток 2**

**Приклад програми виведення даних на монітор**

{Програма виконує заповнення всього екрану точками білого кольрору}

program pixel;

uses crt,graph;

var i,j,k,planes:integer;

ch:char;

gd,gm:integer;

segadr,ofsadr:word;

begin

write('вариант -> ');

readln(k);

gd:= VGA; gm:= VGAHi;

initgraph(gd,gm,'d:\bp\bgi');

setbkcolor(black);

case k of

{Заповнення екрану точками з використанням стандартної процедури Putpixel мови Borland Pascal 7.0}

1: begin

for i:=0 to getmaxy do begin

for j:=0 to getmaxx do begin

putpixel(j,i,white);

end;

end;

end;

{Заповнення екрану точками з використанням функцій переривання Int 10h}

2: begin

setactivepage(0);

setvisualpage(0);

for i:=0 to getmaxy do begin

for j:=0 to getmaxx do begin

asm

mov ah,0ch

mov al,0fh

mov bh,0

mov cx,j

mov dx,i

int 10h

end;

end;

end;

end;

{ Заповнення екрану точками з використанням безпосереднього запису у відеопам‘ять}

3: begin

setactivepage(0);

setvisualpage(0);

segadr:=$a000;

ofsadr:=$0000;

for i:=0 to getmaxy do begin

for planes:=0 to 3 do begin

port[$3c4]:=$02;

case planes of

0:port[$3c5]:=$01;

1:port[$3c5]:=$02;

2:port[$3c5]:=$04;

3:port[$3c5]:=$08;

end;

for j:=0 to 79 do begin

ofsadr:=i\*80+j;

case planes of

0:mem[segadr:ofsadr]:=$ff;

1:mem[segadr:ofsadr]:=$ff;

2:mem[segadr:ofsadr]:=$ff;

3:mem[segadr:ofsadr]:=$ff;

end;

end;

end;

end;

end;

end; {case}

ch:=readkey;

closegraph;

end.

**Додаток 3**

**Приклад програми виведення даних на жорсткий диск**

{Програма виконує запис на жорсткий диск 50000 байтів з ОЗП}

program DMA;

uses dos,crt;

type

user\_rec=record

item1:array [1..1000] of byte;

end;

var ar1:array [1..50000] of byte;

ar2:user\_rec;

i,j,k:word;

f1:file of byte;

f2:file of user\_rec;

f3:file;

f\_name:PathStr;

count:word;

Hour1, Minute1, Second1, Sec100\_1: Word;

Hour2, Minute2, Second2, Sec100\_2: Word;

begin

write('имя файла ->');

readln(f\_name);

write('вариант ->');

readln(k);

for i:=1 to 50000 do ar1[i]:=7;

for j:=1 to 1000 do begin

ar2.item1[j]:=j;

end;

case k of

{За одну операцію виведення виконується запис 1 байту, операція виведення повторюється 50000 разів}

1:begin

assign(f1,f\_name);

rewrite(f1);

gettime(Hour1, Minute1,   
 Second1, Sec100\_1);

for i:=1 to 50000 do begin

write(f1,ar1[i]);

end;

gettime(Hour2, Minute2,   
 Second2, Sec100\_2);

writeln('вариант = ',k:1);

writeln(' начало -> ',

hour1:2,':',minute1:2,':',

second1:2,'.',sec100\_1:3,

' конец -> ',

hour2:2,':',minute2:2,':',

second2:2,'.',sec100\_2:3);

end;

{За одну операцію виведення виконується запис 1000 байтів, операція виведення повторюється 50 разів }

2:begin

assign(f2,f\_name);

rewrite(f2);

gettime(Hour1, Minute1,   
 Second1, Sec100\_1);

for i:=1 to 50 do begin

write(f2,ar2);

end;

gettime(Hour2, Minute2,   
 Second2, Sec100\_2);

writeln('вариант = ',k:1);

writeln(' начало -> ',

hour1:2,':',minute1:2,':',

second1:2,'.',sec100\_1:3,

' конец -> ',

hour2:2,':',minute2:2,':',

second2:2,'.',sec100\_2:3);

end;

{Виведення виконується на низькому рівні по блокам розміром 512 байтів, що дорівнює розміру сектора на жорсткому диску}

3:begin

assign(f3,f\_name);

rewrite(f3,512);

gettime(Hour1, Minute1,   
 Second1, Sec100\_1);

blockwrite(f3,ar1,50000,count);

gettime(Hour2, Minute2,   
 Second2, Sec100\_2);

writeln('вариант = ',k:1);

writeln(' начало -> ',

hour1:2,':',minute1:2,':',

second1:2,'.',sec100\_1:3,

' конец -> ',

hour2:2,':',minute2:2,':',

second2:2,'.',sec100\_2:3);

end;

end; {case}

end.

**ЛІТЕРАТУРА**

1. Гук М. Процессоры Intel от 8086 до Pentium II. – СПб.: Питер, 1998.

2. П. Брамм, Д. Брамм. Микропроцессор 80386 и его программирование. М.: Мир, 1990.

3. Григорьев В.Л. Микропроцессор i486. Архитектура и программирование (в 4-х книгах). М.: ГРАНАЛ, 1993.

4. Колодницький М.М. Технічне та програмне забезпечення комп’ютерних інформаційних технологій. Навчальний посібник. – Житомир: ЖІТІ, 1995.

5. Ларионов А.М., Майоров С.А., Новиков Г.И. Вычислительные комплексы, системы и сети. Учебник для вузов. – Л.: Энергоатомиздат, 1987.

6. Лю Ю - Ч., Гибсон Г. Микропроцессоры семейства 8086/8088. – М.: Радио и связь, 1987.

7. Нортон П. Программно - аппаратная организация IBM PC. – М.: Радио и связь, 1991.

8. Нортон П. Персональный компьютер фирмы IBM и операционная система MS DOS. – М.: Радио и связь, 1991.

9. Нортон П. Справочное руководство по MS DOS. – М.: Радио и связь, 1992.

10. Брябрин В.М. Программное обеспечение персональных ЭВМ. – М.: Наука, 1989.

11. Григорьев В.Л. Видеосистемы ПК фирмы IBM. – М.: Радио и связь, 1993.

12. Абель П. Язык Ассемблера для IBM PC и программирования. М.: Высшая школа, 1992.

13. Скэнлон Л. Персональные ЭВМ IBM PC XT. Программирование на языке ассемблера. – М.: Радио и связь, 1989.

14. Зуев Е.А. Язык прграммирования Turbo Pascal 6.0. – М.: “Унитех”, 1992.

15. Федоров А. Особенности программирования на Borland Pascal. – К.: Диалектика, 1994.

16. Федоров А., Рогаткин Дм. Borland Pascal в среде Windows. – К.: Диалектика,1994.