UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan Pendidikan Berterusan (UTMSPACE)

FINAL EXAMINATION / *PEPERIKSAAN AKHIR*
SEMESTER 2 – SESSION 2016 / 2017
PROGRAM KERJASAMA

COURSE CODE / *KOD KURSUS*	:	DDC 2223 / DDPC 2223
COURSE NAME / *NAMA KURSUS*	:	SYSTEM SOFTWARE *PERISIAN SISTEM*
YEAR / PROGRAMME / *TAHUN / PROGRAM*	:	2 DDC / DDPC
DURATION / *TEMPOH*	:	2 HOUR 30 MINUTES
DATE / *TARIKH*	:	MARCH/APRIL 2017

INSTRUCTION / *ARAHAN* :

1. ANSWER ALL QUESTIONS IN SECTION A, B AND C IN THIS QUESTION PAPER. *JAWAB SEMUA SOALAN DI BAHAGIAN A, B DAN C DALAM KERTAS SOALAN INI.*

(You are required to write your name and your lecturer's name on your answer script)
(Pelajar dikehendaki tuliskan nama anda dan nama pensyarah pada skrip jawapan)

NAME / *NAMA*	:	.....
I.C NO. / *NO. K/PENGENALAN*	:	.....
YEAR / COURSE *TAHUN / KURSUS*	:	.....
COLLEGE NAME *NAMA KOLEJ*	:	.....
LECTURER'S NAME *NAMA PENSYARAH*	:	.....

This examination paper consists of ...19.. pages including the cover
Kertas soalan ini mengandungi ..19.. muka surat termasuk kulit hadapan

SECTION A TRUE/FALSE QUESTIONS(10 Marks)

Instructions: Answer all questions on page 15.

BAHAGIAN A SOALAN BENAR/SALAH (10 Markah)

Arahan: Jawab semua soalan pada mukasurat 15.

1. TRAP#2 is an assembler directive.
   TRAP#2 ialah satu direktif penghimpun.

2. System software acts as the interfaces between the user, the application and the computer.
   Perisian sistem beraksi sebagai antara muka antara pengguna, aplikasi dan komputer.

3. Windows CE.NET and Linux OS are examples of embedded operating system.
   Windows CE.NET dan Linux OS adalah contoh sistem pengoperasian terbenam.

4. Running program with more than one processor is called multiprogramming.
   Melaksana program dengan lebih dari satu pemproses dipanggil berbilang pengaturcaraan.

5. Windows 9.0 is an example of Operating System using GUI interface.
   Windows 9.0 adalah satu contoh Sistem Pengoperasian mengguna antara muka GUI.

6. The instruction MOVE 5,D1 will assign the first least significant word of register D1 with value 5.
   Suruhan MOVE 5,D1 akan mengumpuk perkataan pertama yang kurang bererti bagi daftar D1 dengan nilai 5.

7. The assembler does not translates assembler directives into machine codes.
   Penghimpun tidak menterjemah direktif penghimpun ke dalam kod mesin.

8. Symbol Table is a table that stores address of all labels or symbols used in a source program.
   Jadual Simbol adalah satu jadual yang menyimpan alamat semua label atau simbol dalam satu program sumber.

9. XXX DC.B %00001111 cause the label XXX to be assigned with value $00000F.
   XXX DC.B %00001111 menyebabkan label XXX diberi nilai $00000F.

10. MOVE.L #@778,A1 has an error.
    MOVE.L #@778,A1 mempunyai ralat.

SECTION B OBJECTIVE QUESTIONS (15 Marks)

Instructions: Choose the most appropriate answer.

Answer all questions on page 15.

BAHAGIAN B SOALAN OBJEKTIF (15 Markah)

Arahan: Pilih Jawapan yang paling tepat.

Jawab semua soalan pada muka surat 15.

1. **Parsing** is also known as?

   Imbasan *juga dikenali sebagai?*

   A. Lexical Analysis / *Analisis Leksikal*

   B. Semantic Analysis / *Analisis Semantik*

   C. Syntax Analysis / *Analisis Sintak*

   D. Code Generation / *Penjanaan kod*

2. The **Linker** _____.

   Pemaut _____.

   A. is the same as a loader / *adalah sama dengan pemuat*

   B. is required to create a load module / *diperlukan untuk membentuk satu modul muat*

   C. is the same as an editor / *adalah sama dengan penyunting*

   D. is used before program is executed / *digunakan sebelum program*

3. In a **two pass assembler** the symbol table generation occurs during
*Dalam satu **penghimpun dua laluan** penjanaan jadual simbol berlaku semasa*

A. the second pass / *laluan kedua*

B. the first pass / *laluan pertama*

C. the third pass / *laluan ketiga*

D. None of the above / *Tiada di atas*

4. Which of the following cannot be done by an **assembler**?
*Manakah antara berikut tidak boleh dilakukan oleh satu **penghimpun**?*

A. Convert Assembler to Machine Code / *Tukar Penghimpun kepada Kod Mesin*

B. Generate Symbol Table / *Jana Jadual Simbol*

C. Generate Program Listing / *Jana Senarai Program*

D. All of the above / *Semua di atas*

5. Which of the following is **not** a part of the phases in **compiling process**?
*Manakah antara berikut **bukan** sebahagian fasa dalam **proses pengkompilan**?*

A. Lexical Analysis / *Analisis Leksikal*

B. Semantic Analysis / *Analisis Semantik*

C. Compile Analysis / *Analisis Kompil*

D. *None* of the above / *Tiada di atas*

6. GUI (Graphical User Interface) is used in all computer system below **EXCEPT**
GUI (Antara muka Pengguna Grafik) digunakan dalam semua sistem komputer di bawah ***KECUALI***

A. LINUX
C. Windows 7.0
B. Macintosh System 7.5
D. None of the above

7. _____ instruction is a valid **MC68000 Assembly Language Instruction**.
Arahan _____adalah satu ***Arahan Bahasa Himpunan MC68000*** *yang sah.*

A. CLR.W A8
C. MOVE D2,D4
B. CLR.L (A5)
D. MOVE.W #$ADEF,D2

8. The difference between copying a file and moving a file in **operating system** is_____.
Perbezaan antara menyalin fail dan menggerakkan fail dalam ***sistem pengoperasian*** *adalah* _____.

A. No difference, both are the same.
Tiada perbezaan, kedua-duanya adalah sama.

B. Copying a file will delete the original file while moving a file will not delete any file.
Menyalin fail akan memadamkan fail asal manakala mengerakkan fail tidak akan memadamkan sebarang fail.

C. Copying a file will create another copy of the file and moving a file will not create another copy of file.
Menyalin satu fail akan mewujudkan satu salinan fail yang lain dan mengerakkan fail tidak akan mewujudkan salinan fail yang lain.

D. None of the above
Tiada di atas

9. The following Operating System is embedded **EXCEPT**
*Sistem Pengoperasian berikut adalah terbenam **KECUALI***

A. Windows CE.NET
C. Windows Mobile 2003
B. Palm OS
D. Windows XP

10. The following UNIX command are correctly used **EXCEPT**
*Perintah UNIX berikut digunakan dengan betul **KECUALI***

A. $whoaml
C. $man man
B. $cd ..
D. $chmod 333 txt.c

SECTION C STRUCTURED QUESTIONS (75 Marks)

Instructions: **Answer all questions in the space provided.**

BAHAGIAN C SOALAN BERSTRUKTUR (75 Markah)

Arahan: Jawab semua soalan pada ruang yang disediakan.

1. a) Give **three(3) Commercial Operating System Software.** [3 M]

*Berikan **tiga(3)** Perisian Sistem Pengoperasian Komersial.*

b) Name **three(3)** method of Interface that you can implement on an **Operating System.** [3 M]

*Namakan **tiga(3)** kaedah antara muka yang anda boleh laksana pada **Sistem Pengoperasian.***

c) State **four(4)** functions of **Operating System.** [4 M]

*Nyatakan **empat(4)** fungsi **Sistem Pengoperasian.***

2. With reference to the following data registers, address registers and main memory below, show the content of the register or main memory location that has changed, after the execution of the following instruction. [10M]

Merujuk kepada daftar data, daftar alamat dan ingatan utama di bawah, tunjukkan kandungan daftar atau lokasi ingatan utama yang telah bertukar, setelah perlaksanaan arahan berikut.

DATA REGISTER

D0	1111 1111
D1	2222 2222
D2	0001 0010
D3	4477 FFF5

ADDRESS REGISTER

A0	0022 0300
A1	0022 0302
A2	0022 0304
A3	0022 0306

MAIN MEMORY

$220300	AA	88
$220302	BB	99
$220304	FF	AD
$220306	33	BD

a) CLR.W (A1)

D0	1111 1111
D1	2222 2222
D2	0001 0010
D3	4477 FFF5

A0	0022 0300
A1	0022 0302
A2	0022 0304
A3	0022 0306

$220300		
$220302		
$220304		
$220306		

b) SUB.B D2,(A2)+

D0	1111 1111
D1	2222 2222
D2	0001 0010
D3	4477 FFF5

A0	0022 0300
A1	0022 0302
A2	
A3	0022 0306

$220300		
$220302		
$220304		
$220306		

c) ADD.W #7,2(A0)

D0	1111 1111
D1	2222 2222
D2	0001 0010
D3	4477 FFF5

A0	0022 0300
A1	0022 0302
A2	0022 0304
A3	0022 0306

$220300		
$220302		
$220304		
$220306		

d) MULS D3,D2

D0	1111 1111
D1	2222 2222
D2	
D3	4477 FFF5

A0	0022 0300
A1	0022 0302
A2	0022 0304
A3	0022 0306

$220300		
$220302		
$220304		
$220306		

e) DIVS #4,D2

D0	1111 1111
D1	2222 2222
D2	
D3	4477 FFF5

A0	0022 0300
A1	0022 0302
A2	0022 0304
A3	0022 0306

$220300		
$220302		
$220304		
$220306		

3. a) Why is **Machine code** used? [2M]

*Mengapakah **Kod mesin** digunakan?*

b) Give **five(5)** examples of **Branch Instruction** for **Motorola 68000 microprocessor.** [5M]

*Berikan **five(5)** contoh **Arahan Cabang** bagi **mikro pemproses Motorola 68000.***

c) Draw the diagram how **Editor, Assembler, Compiler, Linker** and **Loader** functions with respect to **Source Code, Object Code** and **Executable Code.** [5M]

*Lukiskan gambar rajah bagaimana **Penyunting, Penghimpun, Pengkompil, Pemaut** dan **Pemuat** berfungsi berdasarkan kepada **Kod Sumber, Kod Objek** dan **Kod BolehLaksana.***

d) What is the advantage of **Two Pass Assembler**? [2M]

*Apakah kebaikan **Penghimpun Dua Laluan**?*

4. With reference to **APPENDIX B**, assemble the source code below manually and built it's machine code. [16M]

*Merujuk kepada **APPENDIX B**, himpunkan kod sumber di bawah secara manual dan bina kod mesinnya.*

ADDRESS *ALAMAT*	MACHINE CODE *KOD MESIN*	SOURCE CODE *KOD SUMBER*
00001000		1 ORG $1000
00001000		2 START:
00001000		3 ADD.B X,D0
00001006		4 ADD.L Y,D1
0000100C		5 ADD D5,D7
0000100E		6 ADD.B Y,D0
00001014		7 ADD.L #255,D0
00001018		8 ADD D7,D6
0000101A		9 ADD.B D0,Z
00000600		10 ORG $600
00000600	=	11 Y: DC.B 10
00000601	=	12 X: DC.B 100
00000602		13 Z: DS.B 1

5. Given the PASCAL statement **X := A DIV C + B * 60**, state the function of each stage and its generated product during compilation. Refer to **APPENDIX A**. [15M]

*Diberi pernyataan PASCAL **X := A DIV C + B * 60**, nyatakan fungsi setiap tahap dan hasil yang dijana semasa pengkompilan. Rujuk pada **APPENDIX A**.*

a) Lexical Analysis / *Analisis Leksikal*

b) Sintax Analysis / *Analisis Sintak*

c) Code Generation / *Penjanaan Kod*

6. a) What are the functions of **LOCATION COUNTER**, **OPERATION CODE TABLE** and **SYMBOL TABLE** in a **Two Pass Assembling** process? [6 M]

*Apakah fungsi **PEMBILANG LOKASI**, **JADUAL KOD OPERASI** dan **JADUAL SIMBOL** dalam satu proses **Penghimpunan Dua Laluan?***

b) An assembly language source code is processed by a **Two Pass Assembler** to form object code. What happen in **Pass-1** process and **Pass-2** process stages? [4 M]

*Satu kod sumber bahasa himpunan diproses oleh sebuah **Penghimpun Dua Laluan** untuk membentuk kod objek. Apakah berlaku dalam peringkat proses **Laluan-1** dan proses **Laluan-2**?*

ANSWER SHEET FOR SECTION A AND SECTION B

RUANG JAWAPAN BAHAGIAN A DAN BAHAGIAN B

SECTION A (10 Marks)

BAHAGIAN A (10 Markah)

1		6	
2		7	
3		8	
4		9	
5		10	

SECTION B (15 Marks)

BAHAGIAN B (15 Markah)

1		6	
2		7	
3		8	
4		9	
5		10	

APPENDIX A

PASCAL LANGUAGE SYNTAX / *SINTAK BAHASA PASCAL*

1.	<prog>	::= PROGRAM <prog-name> VAR <dec-list> BEGIN <stmt-list> END
2.	<prog-name>	::= Id
3.	<dec-list>	::= <dec> | <dec-list> ; <dec>
4.	<dec>	::= <id-list>:<type>
5.	<type>	::= INTEGER
6.	<id-list>	::= Id | <id-list> , Id
7.	<stmt-list>	::= <stmt> | <stmt-list> ; <stmt>
8.	<stmt>	::= <assign> | <read> | <write> | <for>
9.	<assign>	::= Id::= <exp>
10.	<exp>	::= <term> | <exp> + <term> | <exp> - <term>
11.	<term>	::= <factor> | <term> * <factor> | <term> DIV <factor>
12.	<factor>	::= id | int | (<exp>)
13.	<read>	::= READ (<id-list>)
14.	<write>	::= WRITE (<id-list>)
15.	<for>	::= FOR <index-exp> DO <body>
16.	<index-exp>	::= id:=<exp> TO <exp>
17.	<body>	::= <stmt> | BEGIN <stmt-list> END

APPENDIX B

MC60000 INSTRUCTION FORMAT / *FORMAT ARAHAN MC60000*

Integer Instructions

ADD

Add
(M68000 Family)

ADD

Operation: Source + Destination → Destination

Assembler Syntax:
ADD < ea > ,Dn
ADD Dn, < ea >

Attributes: Size = (Byte, Word, Long)

Description: Adds the source operand to the destination operand using binary addition and stores the result in the destination location. The size of the operation may be specified as byte, word, or long. The mode of the instruction indicates which operand is the source and which is the destination, as well as the operand size.

Condition Codes:

X	N	Z	V	C
*	*	*	*	*

X — Set the same as the carry bit.
N — Set if the result is negative; cleared otherwise.
Z — Set if the result is zero; cleared otherwise.
V — Set if an overflow is generated; cleared otherwise.
C — Set if a carry is generated; cleared otherwise.

Instruction Format:

15	14	13	12	11 10 9	8 7 6	5 4 3	2 1 0
1	1	0	1	REGISTER	OPMODE	EFFECTIVE ADDRESS MODE	EFFECTIVE ADDRESS REGISTER

APPENDIX B

MC60000 INSTRUCTION FORMAT / *FORMAT ARAHAN MC60000*

ADD

Add
(M68000 Family)

ADD

Instruction Fields:

Register field—Specifies any of the eight data registers.

Opmode field

Byte	Word	Long	Operation
000	001	010	< ea > + Dn → Dn
100	101	110	Dn + < ea > → < ea >

Effective Address field—Determines addressing mode.

a. If the location specified is a source operand, all addressing modes can be used as listed in the following tables:

Addressing Mode	Mode	Register
Dn	000	reg. number:Dn
An*	001	reg. number:An
(An)	010	reg. number:An
(An) +	011	reg. number:An
– (An)	100	reg. number:An
(d_{16},An)	101	reg. number:An
(d_8,An,Xn)	110	reg. number:An

Addressing Mode	Mode	Register
(xxx).W	111	000
(xxx).L	111	001
#<data>	111	100
(d_{16},PC)	111	010
(d_8,PC,Xn)	111	011

MC68020, MC68030, and MC68040 only

(bd,An,Xn)**	110	reg. number:An	(bd,PC,Xn)†	111	011
([bd,An,Xn],od)	110	reg. number:An	([bd,PC,Xn],od)	111	011
([bd,An],Xn,od)	110	reg. number:An	([bd,PC],Xn,od)	111	011

*Word and long only
**Can be used with CPU32.

APPENDIX B
MC60000 INSTRUCTION FORMAT / *FORMAT ARAHAN MC60000*

Integer Instructions

ADD — **Add** (M68000 Family) — **ADD**

b. If the location specified is a destination operand, only memory alterable addressing modes can be used as listed in the following tables:

Addressing Mode	Mode	Register
Dn	—	—
An	—	—
(An)	010	reg. number:An
(An) +	011	reg. number:An
– (An)	100	reg. number:An
(d_{16},An)	101	reg. number:An
(d_8,An,Xn)	110	reg. number:An

Addressing Mode	Mode	Register
(xxx).W	111	000
(xxx).L	111	001
#<data>	—	—
(d_{16},PC)	—	—
(d_8,PC,Xn)	—	—

MC68020, MC68030, and MC68040 only

(bd,An,Xn)*	110	reg. number:An
([bd,An,Xn],od)	110	reg. number:An
([bd,An],Xn,od)	110	reg. number:An

(bd,PC,Xn)*	—	—
([bd,PC,Xn],od)	—	—
([bd,PC],Xn,od)	—	—

*Can be used with CPU32

NOTE

The Dn mode is used when the destination is a data register; the destination < ea > mode is invalid for a data register.

ADDA is used when the destination is an address register. ADDI and ADDQ are used when the source is immediate data. Most assemblers automatically make this distinction.