

## เอกสารอ้างอิง

### ภาษาไทย

อิน กุ๊วรวรรณ, ชัยยงค์ วงศ์ชัยสวัสดิ์, ดร. และไพศาล สงวนหม่ม, ดร.. เทคโนโลยีไมโครคอมพิวเตอร์  
16 บิต. พิมพ์ครั้งที่ 1. กรุงเทพฯ:บริษัท ซีเอ็ดยูเคชั่น จำกัด., 2530.  
วสิน เพิ่มทวีชัย. คู่มือ MS-DOS PC-DOS. พิมพ์ครั้งที่ 3. กรุงเทพฯ: VS Enterprise., 2532.

### ภาษาอังกฤษ

Andleigh, Prabhat K. UNIX System Architecture. Prentice-Hall Inc.. Copyright  
1990.

Leffler, Samuel J., McKusick, Karshall Kirk, Karels, Michael J. and Quarterman,  
John S.. The Design and Implementation of the 4.3BSD UNIX Operating  
System, Addison-Wesley Publishing Company, Inc.. 1989.

Bach, Maurice J. The Design of The UNIX Operating System. Prentice-Hall Inc..  
Copyright 1986.

Hayes, John P. Computer architecture and organization. international edition.  
1st Printing. McGraw-Hill Company. 1988.

Wilkinson, Barry. Computer architecture: design and performance. 1st printing.  
Prentice Hall International (UK) Ltd. 1991.

Bartee, Thomas C. Digital computer fundamentals. international student edition.  
1st printing. McGraw-Hill Book Company. 1985.

Deitel, H. M. An Introduction to Operating Systems. Second Edition.  
Addison-Wesley Publishing Company. Copyright 1990.

Wear, Larry L., Pinkert, James R., Wear, Larry C. and Lane, William G. Computer: An  
introduction to hardware and software design. International Edition.  
McGraw-Hill, Inc. 1991.

---



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

```

_TEXT          SEGMENT      byte public 'CODE'
              ASSUME      CS:_TEXT,DS:_DATA
_TEXT          ENDS

_DATA         SEGMENT      word public 'DATA'

_PSP_Segment  PUBLIC      _PSP_Segment,_Line,_Block
_PSP_Segment  DW          ?
_Line         DW          ?
_Block       DW          ?

_Head         PUBLIC      _Head,_Not_Used
_Head         DW          ?
_Not_Used    DW          ?

              PUBLIC      _Track_Number,_Sector_Number,_Head_Number
              PUBLIC      _Drive_Number,_Number_of_Sectors
              PUBLIC      _Function_Number
_Track_Number DW          ?
_Sector_Number DB        ?
_Head_Number  DB        ?
_Drive_Number DB        ?
_Number_of_Sectors DB    ?
_Function_Number DB      ?

_Buffer_Offset PUBLIC    _Buffer_Offset,_Buffer_Segment
_Buffer_Offset DW        ?
_Buffer_Segment DW        ?

_Sector_Count PUBLIC    _Sector_Count
_Sector_Count DB        ?

_Disk_C       PUBLIC    _Disk_C,_Disk_D
_Disk_C       DB        ?
              ;Type
_Disk_C       DB        ?
              ;Number of Head
_Disk_C       DB        ?
              ;Number of Sector
_Disk_D       PUBLIC    _Disk_D
_Disk_D       DB        ?
              ;Type
_Disk_D       DB        ?
              ;Number of Head
_Disk_D       DB        ?
              ;Number of Sector

              PUBLIC    _Register_Flag,_Register_CS,_Register_IP
              PUBLIC    _Register_AL,_Register_AH
              PUBLIC    _Register_BL,_Register_BH
              PUBLIC    _Register_CL,_Register_CH
              PUBLIC    _Register_DL,_Register_DH
              PUBLIC    _Register_BP,_Register_DI
              PUBLIC    _Register_SI,_Register_DS,_Register_ES
_Register_Flag DW        ?
_Register_CS   DW        ?
_Register_IP   DW        ?

_Register_AL   DB        ?
_Register_AH   DB        ?
_Register_BL   DB        ?

```

```

_Register_BH      DB  ?
_Register_CL      DB  ?
_Register_CH      DB  ?
_Register_DL      DB  ?
_Register_DH      DB  ?

_Register_BP      DW  ?

_Register_DI      DW  ?
_Register_SI      DW  ?

_Register_DS      DW  ?
_Register_ES      DW  ?

_Entry_Point     PUBLIC  _Entry_Point
_Entry_Point     DW  ?
_Entry_Point     DW  ?

_Timer           PUBLIC  _Timer, _Lock, _Old_Timer
_Timer           DW  ?
_Lock           DW  ?
_Old_Timer      DW  ?
_Old_Timer      DW  ?

_Buffer_Area     PUBLIC  _Buffer_Area
_Buffer_Area     DB  ?

_DATA           ENDS
               END

;               *** Block Structure ***
;
;               1 Forward Pointer           2 bytes
;               2 Track Number            2 bytes
;               3 Sector Number           1 byte
;               4 Head Number             1 byte
;               5 Drive Number            1 byte
;               6 Status Word             1 byte
;               7 Buffer Area              512 bytes
;

```

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



```
_TEXT          SEGMENT byte public 'CODE'  
              ASSUME  CS:_TEXT,DS:_DATA  
  
              EXTRN  _Time_Routine:FAR  
              EXTRN  _Cache:FAR
```

```
_Initialize    PROC FAR
```

```
-----  
:              _Initialize Routine  
:-----
```

```
MOV  AX,_DATA  
MOV  DS,AX  
MOV  AX,ES  
MOV  _PSP_Segment,AX  
MOV  AX,_DATA  
ADD  AX,1000h  
MOV  SS,AX  
MOV  SP,0FFFFh
```

```
-----  
:              Get Command Line Parameters  
:-----
```

```
MOV  BX,080h  
XOR  AX,AX  
XOR  CX,CX  
MOV  AL,ES:[BX]  
INC  BX  
ADD  AX,BX  
MOV  _Line,AX  
CMP  AX,BX  
JZ   _Label003
```

```
_Label001:  MOV  AL,ES:[BX] ;Character '/'  
            CMP  AL,02Fh  
            JZ   _Label004  
_Label002:  INC  BX  
            CMP  BX,_Line  
            JNZ  _Label001  
_Label003:  JMP  _Label018 ;Jump To Install  
_Label004:  INC  BX  
            CMP  BX,_Line  
            JZ   _Label003  
            MOV  AL,ES:[BX] ;Character 'R'  
            CMP  AL,052h  
            JNZ  _Label005  
            JMP  _Label015 ;Jump To Remove  
_Label005:  CMP  AL,072h ;Character 'r'  
            JNZ  _Label006  
            JMP  _Label015 ;Jump To Remove  
_Label006:  CMP  AL,042h ;Character 'B'  
            JNZ  _Label007  
            JMP  _Label009 ;Get Parameter
```

```

_Label007:      CMP AL,062h      ;Character 'b'
                JNZ _Label008
                JMP _Label009      ;Get Parameter
_Label008:      JMP _Label002      ;Get New Character
_Label009:      INC BX
                CMP BX,_Line
                JNZ _Label010
                JMP _Label003
_Label010:      MOV AL,ES:[BX]
                CMP AL,03Dh      ;Character '='
                JZ _Label011
                JMP _Label002
_Label011:      INC BX
                CMP BX,_Line
                JNZ _Label012
                JMP _Label003
_Label012:      MOV AL,ES:[BX]
                CMP AL,030h
                JGE _Label013
                JMP _Label002
_Label013:      CMP AL,039h
                JLE _Label014
                JMP _Label002
_Label014:      AND AX,0Fh
                ADD CX,CX
                MOV DX,CX
                ADD CX,CX
                ADD CX,CX
                ADD CX,DX
                ADD CX,AX
                JMP _Label011
_Label015:      MOV AH,0A0h
                INT 13h
                JNC _Label016
                JMP _Label020
_Label016:      MOV AH,0A1h
                INT 13h      ;Clear Buffer
                JNC _Label017
                JMP _Label020
_Label017:      MOV DS,AX
                MOV BX,002CH
                MOV AX,[BX]
                MOV ES,AX
                MOV AH,049H
                INT 21H
                MOV AX,DS
                MOV ES,AX
                MOV AH,049H
                INT 21H
                JMP _Label020
_Label018:      AND CX,CX
                JNZ _Label019
                MOV CX,32      ;Default 32 Blocks
_Label019:      MOV _Block,CX

```

```

MOV AH,0A0h
INT 13h
JC _Label021
_Label020: MOV AX,04C00h
INT 21h
_Label021:
;-----;
;
; GET DRIVE TYPE AND DRIVE PARAMETER
;
;-----;
MOV AH,015h
MOV DL,080h
INT 13h
JNC _Label022
_Label022: MOV AH,00h
MOV BYTE PTR _Disk_C,AH
CMP AH,03h
JNE _Label023
MOV AH,08h
MOV DL,080h
INT 13h
MOV BYTE PTR _Disk_C+2,DH ;Head
AND CL,03Fh
MOV BYTE PTR _Disk_C+1,CL ;Sector
_Label023: MOV AH,015h
MOV DL,081h
INT 13h
JNC _Label024
_Label024: MOV AH,00h
MOV BYTE PTR _Disk_D,AH
CMP AH,03h
JNE _Label025
MOV AH,08h
MOV DL,081h
INT 13h
MOV BYTE PTR _Disk_D+2,DH ;Head
AND CL,03Fh
MOV BYTE PTR _Disk_D+1,CL ;Sector
_Label025:
;-----;
; GET ENTRY POINT OF INT 13H ROUTINE
;
;-----;
MOV DX,ES
MOV AH,035h
MOV AL,013h
INT 21h
MOV WORD PTR _Entry_Point,bx ;Offset
MOV BX,ES
MOV WORD PTR _Entry_Point+2,Bx ;Segment
;-----;

```

```

;
; GET ENTRY POINT OF INT 1Ch ROUTINE
;

```

77

```

MOV DX,ES
MOV AH,035h
MOV AL,01Ch
INT 21h
MOV WORD PTR _Old_Timer,bx ;Offset
MOV BX,ES
MOV WORD PTR _Old_Timer+2,Bx ;Segment

```

```

;
; SET NEW ENTRY POINT OF INT 13H ROUTINE
;

```

```

MOV AX,CS
MOV DS,AX ;Segment
MOV DX,offset _TEXT:_Cache ;Offset
MOV AH,025h
MOV AL,013h ;Interrupt
INT 21h

```

```

;
; SET NEW ENTRY POINT OF INT 1Ch ROUTINE
;

```

```

MOV AX,CS
MOV DS,AX ;Segment
MOV DX,offset _TEXT:_Time_Routine
MOV AH,025h
MOV AL,01Ch ;Interrupt
INT 21h

```

```

;
; PREPARE BUFFER AREA
;

```

```

MOV AX,_DATA
MOV DS,AX
MOV CX,_Block ;Number Of Block
MOV DX,0
MOV _Head,DX
MOV BX,OFFSET Buffer_Area
JMP _Label027
_Label026:
MOV DX,BX
ADD BX,520
_Label027:
MOV [BX],DX

MOV AX,0
MOV [BX+2],AX ;Track Number
MOV [BX+4],AL ;Sector Number
MOV [BX+5],AL ;Head Number

```



```
MOV [BX+6],AL ;Drive Number
MOV [BX+7],AL ;Status Word
```

78

```
LOOPNZ _Label026
```

```
MOV _Not_Used,BX
ADD BX,520
```

```
-----
;
; Keep Cache Routine
;
-----
```

```
MOV AX,BX
AND AX,0FFF0h
CMP AX,BX
JZ _Label028
ADD AX,010h
_Label028: MOV CX,4
SHR AX,CL
MOV DX,AX
MOV AX,_DATA
ADD DX,AX
MOV AX,_PSP_Segment
SUB DX,AX ;Paragraph available
_Label029: MOV AX,03100H
INT 21H
```

```
_Initialize
```

```
ENDP
```

```
_TEXT
```

```
ENDS
```

```
_DATA
```

```
SEGMENT word public 'DATA'
```

```
EXTRN _PSP_Segment:WORD
```

```
EXTRN _Line:WORD
```

```
EXTRN _Block:WORD
```

```
EXTRN _Head:WORD
```

```
EXTRN _Not_Used:WORD
```

```
EXTRN _Disk_C:BYTE
```

```
EXTRN _Disk_D:BYTE
```

```
EXTRN _Old_Timer:WORD
```

```
EXTRN _Entry_Point:WORD
```

```
EXTRN _Buffer_Area:BYTE
```

```
_DATA
```

```
ENDS
```

```
END
```

```
*** Block Structure ***
```

```
;
;
; 1 Forward Pointer 2 bytes
; 2 Track Number 2 bytes
```

; ; ; ; ; ; ;

3	Sector Number	1 byte
4	Head Number	1 byte
5	Drive Number	1 byte
6	Status Word	1 byte
7	Buffer Area	512 bytes



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

\_TEXT

SEGMENT byte public 'CODE'  
ASSUME CS:\_TEXT,DS:\_DATA

80

EXTRN \_Read\_Function:FAR  
EXTRN \_Write\_Function:FAR  
EXTRN \_Verify\_Function:FAR  
EXTRN \_Format\_Function:FAR  
EXTRN \_Check\_Cache:FAR  
EXTRN \_Get\_PSP:FAR  
EXTRN \_To\_BIOS:FAR  
EXTRN \_To\_DOS:FAR

PUBLIC \_Cache  
PROC FAR

\_Cache

Get Data From Registers

PUSH DS  
PUSH AX  
MOV AX,\_DATA  
MOV DS,AX  
POP AX

MOV WORD PTR \_Register\_AL,AX  
MOV WORD PTR \_Register\_BL,BX  
MOV WORD PTR \_Register\_CL,CX  
MOV WORD PTR \_Register\_DL,DX

MOV \_Register\_BP,BP  
MOV \_Register\_DI,DI  
MOV \_Register\_SI,SI

MOV BP,SP  
MOV AX,[BP+4]  
MOV \_Register\_CS,AX  
MOV AX,[BP]  
MOV \_Register\_DS,AX  
MOV AX,ES  
MOV \_Register\_ES,AX

MOV AX,[BP+2]  
MOV \_Register\_IP,AX  
MOV AX,[BP+6]  
MOV \_Register\_Flag,AX  
POP AX

POP AX ;POP IP Register  
POP AX ;POP CS Register  
POP AX ;POP Flag Register

Get Parameters From Data

```

MOV AX, _Register_ES
MOV ES, AX
MOV AX, WORD PTR _Register_AL
MOV BX, WORD PTR _Register_BL
MOV CX, WORD PTR _Register_CL
MOV DX, WORD PTR _Register_DL
MOV _Function_Number, AH
MOV _Number_of_Sectors, AL
MOV AL, CL
AND CL, 03Fh
MOV _Sector_Number, CL
AND AX, 0C0h
MOV CL, 2
SHL AX, CL
MOV AL, CH
MOV _Track_Number, AX
MOV _Head_Number, DH
MOV _Drive_Number, DL
MOV AX, ES
MOV _Buffer_Segment, AX
MOV _Buffer_Offset, BX

```

```

Set Lock

```

```

MOV AX, 01h
MOV _Lock, AX
MOV AL, _Drive_Number
AND AL, 080h
JNZ _Label000
MOV AX, 19
MOV _Timer, AX

```

Label000:

- ```

1.) Check Function Number
2.) Jump To Function

```

```

XOR AX, AX
MOV AH, _Function_Number
CMP AH, 02h
JNZ _Label001
JMP _Read_Function ;Read Sectors
_Label001:
CMP AH, 03h
JNZ _Label002
JMP _Write_Function ;Write Sectors
_Label002:
CMP AH, 04h
JNZ _Label003
JMP _Verify_Function ;Verify Sectors
_Label003:
CMP AH, 05h

```

```

                JNZ  _Label004
_Label004:      JMP  _Format_function ;Format Track,Cylinder 82
                CMP  AH,0A0h
                JNZ  _Label005
_Label005:      JMP  _Check_Cache ;Check Cache Installed
                CMP  AH,0A1h
                JNZ  _Label006
_Label006:      JMP  _Get_PSP ;Clear Buffer
                JMP  _To_BIOS ;Other Function Number

_Cache         ENDP

_TEXT          ENDS

_DATA         SEGMENT word public 'DATA'

                EXTRN _Track_Number:WORD
                EXTRN _Sector_Number:BYTE
                EXTRN _Head_Number:BYTE
                EXTRN _Drive_Number:BYTE
                EXTRN _Number_of_Sectors:BYTE
                EXTRN _Function_Number:BYTE

                EXTRN _Buffer_Offset:WORD
                EXTRN _Buffer_Segment:WORD

                EXTRN _Register_Flag:WORD
                EXTRN _Register_CS:WORD
                EXTRN _Register_IP:WORD

                EXTRN _Register_AL:BYTE
                EXTRN _Register_AH:BYTE
                EXTRN _Register_BL:BYTE
                EXTRN _Register_BH:BYTE
                EXTRN _Register_CL:BYTE
                EXTRN _Register_CH:BYTE
                EXTRN _Register_DL:BYTE
                EXTRN _Register_DH:BYTE

                EXTRN _Register_BP:WORD

                EXTRN _Register_DI:WORD
                EXTRN _Register_SI:WORD

                EXTRN _Register_DS:WORD
                EXTRN _Register_ES:WORD

                EXTRN _Lock:WORD
                EXTRN _Timer:WORD

_DATA         ENDS
                END

```

```
;
```

```
*** Block Structure ***
```

|   |                 |           |
|---|-----------------|-----------|
| 1 | Forward Pointer | 2 bytes   |
| 2 | Track Number    | 2 bytes   |
| 3 | Sector Number   | 1 byte    |
| 4 | Head Number     | 1 byte    |
| 5 | Drive Number    | 1 byte    |
| 6 | Status Word     | 1 byte    |
| 7 | Buffer Area     | 512 bytes |



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

```

_TEXT          SEGMENT  byte public 'CODE'
               ASSUME   CS:_TEXT,DS:_DATA

               EXTRN    _To_DOS:FAR

               PUBLIC   _Read_Function
               PROC FAR

```

84

```

_Read_Function

```

```

;-----;
;               Function Number 02h (Read Sectors)
;-----;

```

```

_label001:    XOR  AL,AL
              MOV  _Sector_Count,AL
_label002:    MOV  AL,_Number_of_Sectors
              AND  AL,AL
              JNZ  _label003
              JMP  _label021      ;End of Read

```

```

;-----;
;               Find Sector In Buffer Cache
;-----;

```

```

_label003:    MOV  BX,_Head
              MOV  DX,0h
              CMP  BX,0h
_label004:    JZ   _labelx08      ;Jump is Tail Block
              MOV  AL,[BX+6]      ;Drive Number
              CMP  AL,_Drive_Number
              JNZ  _label006
              MOV  AL,[BX+5]      ;Head Number
              CMP  AL,_Head_Number
              JNZ  _label006
              MOV  AL,[BX+4]      ;Sector Number
              CMP  AL,_Sector_Number
              JNZ  _label006
              MOV  AX,[BX+2]      ;Track Number
              CMP  AX,_Track_Number
              JNZ  _label006
              CMP  DX,0h
              JZ   _label005
              MOV  CX,[BX]
              XCHG BX,DX
              MOV  [BX],CX
              XCHG BX,DX
              MOV  DX,_Head
              MOV  [BX],DX
              MOV  _Head,BX
_label005:    JMP  _label012
_label006:    MOV  CX,DX
              MOV  DX,BX
              MOV  BX,[BX]
              CMP  BX,0h

```

```

JNZ _label004
MOV BX,DX
MOV DX,CX
MOV CX,_Not_Used
CMP CX,0h
JNZ _Labelx08

CMP DX,0h
JNZ _Labelx06
MOV _Head,CX
JMP _Labelx07
_Labelx06: XCHG BX,DX
MOV [BX],CX
XCHG BX,DX
_Labelx07: MOV _Not_Used,BX

_Labelx08: MOV BX,_Not_Used
MOV CX,[BX]
MOV _Not_Used,CX
MOV CX,_Head
MOV _Head,BX
MOV [BX],CX

MOV AL,_Drive_Number
MOV [BX+6],AL
MOV AL,_Head_Number
MOV [BX+5],AL
MOV AL,_Sector_Number
MOV [BX+4],AL
MOV AX,_Track_Number
MOV [BX+2],AX

;-----;
;
; Read One Sector from Disk
;
;-----;
_label008: PUSHF
MOV AX,CS
PUSH AX
MOV AX,OFFSET _TEXT:_label009
PUSH AX
MOV AX,_Entry_Point + 2
PUSH AX
MOV AX,_Entry_Point
PUSH AX
MOV AX,_DATA
PUSH AX
MOV AX,0201h
PUSH AX
MOV AX,_DATA
MOV ES,AX
MOV BX,_Head
MOV AX,[BX+2] ;Track Number
XOR AL,AL ;Clear AL

```



```

MOV CL,02
SHR AX,CL
MOV CL,AL
MOV AX,[BX+2] ;Track Number
MOV CH,AL
MOV AL,[BX+4] ;Sector Number
AND AL,03Fh
OR AL,CL
MOV CL,AL
MOV DH,[BX+5] ;Head Number
MOV DL,[BX+6] ;Drive Number
ADD BX,8
POP AX
POP DS
RET

_label009: JNC _label012 ;No Error
MOV BX,AX
MOV AX,_DATA
MOV DS,AX
MOV BL,_Sector_Count
MOV WORD PTR _Register_AL,BX
MOV AX,_Register_Flag
OR AX,01h
MOV _Register_Flag,AX

MOV BX,_Head
MOV CX,[BX]
MOV _Head,CX

MOV CX,_Not_Used
MOV _Not_Used,BX
MOV [BX],CX

JMP _To_DOS

```

```

:-----:
: Copy Data From Buffer Block To Memory Area :
:-----:

```

```

_label012: MOV AX,_DATA
MOV DS,AX
MOV BX,_Head
ADD BX,8
MOV SI,BX
MOV AX,0FFFFh
SUB AX,_Buffer_Offset
CMP AX,512
JNC _label013
MOV AX,_Register_Flag
OR AX,01h
MOV _Register_Flag,AX
MOV AH,09h
MOV AL,_Sector_Count

```

```
MOV WORD PTR _Register_AL,AX
JMP _To_DOS
```

87

```
_label013: MOV AX,_Buffer_Segment
MOV ES,AX
MOV DI,_Buffer_Offset
MOV CX,512
REP MOVSB
MOV _Buffer_Offset,DI
MOV AL,_Number_of_Sectors
DEC AL
MOV _Number_of_Sectors,AL
MOV AL,_Sector_Count
INC AL
MOV _Sector_Count,AL
MOV AL,_Drive_Number
CMP AL,00h
JZ _label016
CMP AL,01h
JZ _label016
CMP AL,80h
JNZ _label014
JMP _label017
_label014: CMP AL,81h
JNZ _label015
JMP _label019
_label015: JMP _label015
```

```
-----
: Calculate Next Sector for Floppy Disk
:-----
```

```
_label016: MOV AL,_Sector_Number
INC AL
MOV _Sector_Number,AL
JMP _label002
```

```
-----
: Cacculate Next Sector for Hard Disk
:-----
```

```
_label017: XOR AX,AX
MOV AL,_Sector_Number
INC AL
MOV _Sector_Number,AL
CMP AL,BYTE PTR _Disk_C + 1
JNG _label018
MOV AL,01h
MOV _Sector_Number,AL
MOV AL,_Head_Number
INC AL
MOV _Head_Number,AL
CMP AL,BYTE PTR _Disk_D + 2
JNG _label018
```

```

XOR AX,AX
MOV _Head_Number,AL
MOV AX,_Track_Number
INC AX
MOV _Track_Number,AX
_label018: JMP _label002
_label019: XOR AX,AX
MOV AL,_Sector_Number
INC AL
MOV _Sector_Number,AL
CMP AL,BYTE PTR _Disk_D + 1
JNG _label020
MOV AL,01h
MOV _Sector_Number,AL
MOV AL,_Head_Number
INC AL
MOV _Head_Number,AL
CMP AL,BYTE PTR _Disk_D + 2
JNG _label020
XOR AX,AX
MOV _Head_Number,AL
MOV AX,_Track_Number
INC AX
MOV _Track_Number,AX
_label020: JMP _label002
;-----;
; Return to DOS ;
;-----;
_label021: MOV AX,_Register_Flag
AND AX,OFFFEh
MOV _Register_Flag,AX
XOR AX,AX
MOV AL,_Sector_Count
MOV _Register_AL,AL
MOV _Register_AH,AH
JMP _To_DOS

_Read_Function ENDP
_TEXT ENDS
DATA SEGMENT word public 'DATA'

EXTRN _Head:WORD
EXTRN _Not_Used:Word

EXTRN _Track_Number:WORD
EXTRN _Sector_Number:BYTE
EXTRN _Head_Number:BYTE
EXTRN _Drive_Number:BYTE

```



```

_TEXT          SEGMENT  byte public 'CODE'
               ASSUME   CS:_TEXT,DS:_DATA

               EXTRN    _To_DOS:FAR

               PUBLIC   _Write_Function
_Write_Function PROC FAR

```

90

```

-----
;
;           Function Number 03h (Write Sectors)
;
-----

```

```

_Label001:    XOR    AL,AL
               MOV    _Sector_Count,AL
_Label002:    MOV    AL,_Number_of_Sectors
               AND    AL,AL
               JNZ    _Label003
               JMP    _Label021          ;End of Write

```

```

-----
;
;           Find Sector In Buffer Cache
;
-----

```

```

_label003:    MOV    BX,_Head
               MOV    DX,0h
               CMP    BX,0h
               JZ     _labelx08          ;Jump is Tail Block
_label004:    MOV    AL,[BX+6]           ;Drive Number
               CMP    AL,_Drive_Number
               JNZ    _label006
               MOV    AL,[BX+5]         ;Head Number
               CMP    AL,_Head_Number
               JNZ    _label006
               MOV    AL,[BX+4]         ;Sector Number
               CMP    AL,_Sector_Number
               JNZ    _label006
               MOV    AX,[BX+2]         ;Track Number
               CMP    AX,_Track_Number
               JNZ    _label006
               CMP    DX,0h
               JZ     _label005
               MOV    CX,[BX]
               XCHG  BX,DX
               MOV    [BX],CX
               XCHG  BX,DX
               MOV    DX,_Head
               MOV    [BX],DX
               MOV    _Head,BX
_label005:    JMP    _label008
_label006:    MOV    CX,DX
               MOV    DX,BX
               MOV    BX,[BX]
               CMP    BX,0h

```



```

MOV AX, _Buffer_Segment
MOV DS, AX
MOV CX, 512
REP MOVSB
MOV AX, _DATA
MOV DS, AX
MOV _Buffer_Offset, SI

```

```

-----
Write One Sector to Disk
-----

```

```

PUSHF
MOV AX, CS
PUSH AX
MOV AX, OFFSET _TEXT: _Label010
PUSH AX
MOV AX, _Entry_Point + 2
PUSH AX
MOV AX, _Entry_Point
PUSH AX
MOV AX, _DATA
PUSH AX
MOV AX, 0301h
PUSH AX
MOV AX, _DATA
MOV ES, AX
MOV BX, _Head
MOV AX, [BX+2] ;Track Number
XOR AL, AL ;Clear AL
MOV CL, 02
SHR AX, CL
MOV CL, AL
MOV AX, [BX+2] ;Track Number
MOV CH, AL
MOV AL, [BX+4] ;Sector Number
AND AL, 03Fh
OR AL, CL
MOV CL, AL
MOV DH, [BX+5] ;Head Number
MOV DL, [BX+6] ;Drive Number
ADD BX, 8
POP AX
POP DS
RET

```

```

_Label010: JNC _Label013 ;No Error
MOV BX, AX
MOV AX, _DATA
MOV DS, AX
MOV BL, _Sector_Count
MOV WORD PTR _Register_AL, BX
MOV AX, _Register_Flag
OR AX, 01h

```

```

MOV  _Register_Flag,AX

MOV  BX,_Head
MOV  CX,[BX]
MOV  _Head,CX

MOV  CX,_Not_Used
MOV  _Not_Used,BX
MOV  [BX],CX

JMP  _To_DOS

```

```

-----
Count Number of Sector
-----

```

```

_Label013:  MOV  AX,_DATA
            MOV  DS,AX
            MOV  AL,_Number_of_Sectors
            DEC  AL
            MOV  _Number_of_Sectors,AL
            MOV  AL,_Sector_Count
            INC  AL
            MOV  _Sector_Count,AL
            MOV  AL,_Drive_Number
            CMP  AL,00h
            JZ   _Label016
            CMP  AL,01h
            JZ   _Label016
            CMP  AL,80h
            JNZ  _Label014
            JMP  _Label017

_Label014:  CMP  AL,81h
            JNZ  _Label015
            JMP  _Label019

_Label015:  JMP  _Label015

```

```

-----
Calculate Next Sector for Floppy Disk
-----

```

```

_Label016:  MOV  AL,_Sector_Number
            INC  AL
            MOV  _Sector_Number,AL
            JMP  _Label002

```

```

-----
Cacculate Next Sector for Hard Disk
-----

```

```

_Label017:  XOR  AX,AX
            MOV  AL,_Sector_Number
            INC  AL
            MOV  _Sector_Number,AL

```



```

CMP AL,BYTE PTR _Disk_C + 1
JNG _Label018
MOV AL,01h
MOV _Sector_Number,AL
MOV AL,_Head_Number
INC AL
MOV _Head_Number,AL
CMP AL,BYTE PTR _Disk_D + 2
JNG _Label018
XOR AX,AX
MOV _Head_Number,AL
MOV AX,_Track_Number
INC AX
MOV _Track_Number,AX
_Label018: JMP _Label002
_Label019: XOR AX,AX
MOV AL,_Sector_Number
INC AL
MOV _Sector_Number,AL
CMP AL,BYTE PTR _Disk_D + 1
JNG _Label020
MOV AL,01h
MOV _Sector_Number,AL
MOV AL,_Head_Number
INC AL
MOV _Head_Number,AL
CMP AL,BYTE PTR _Disk_D + 2
JNG _Label020
XOR AX,AX
MOV _Head_Number,AL
MOV AX,_Track_Number
INC AX
MOV _Track_Number,AX
_Label020: JMP _Label002
;-----;
; Return to DOS ;
;-----;
_Label021: MOV AX,_Register_Flag
AND AX,OFFFEh
MOV _Register_Flag,AX
XOR AX,AX
MOV AL,_Sector_Count
MOV _Register_AL,AL
MOV _Register_AH,AH
JMP _To_DOS

_Write_Function ENDP
_TEXT ENDS
_DATA SEGMENT word public 'DATA'

```

```

EXTRN  _Head:WORD
EXTRN  _Not_Used:WORD

EXTRN  _Track_Number:WORD
EXTRN  _Sector_Number:BYTE
EXTRN  _Head_Number:BYTE
EXTRN  _Drive_Number:BYTE
EXTRN  _Number_of_Sectors:BYTE
EXTRN  _Function_Number:BYTE

EXTRN  _Buffer_Offset:WORD
EXTRN  _Buffer_Segment:WORD

EXTRN  _Sector_Count:BYTE

EXTRN  _Disk_C:BYTE
EXTRN  _Disk_D:BYTE

EXTRN  _Register_Flag:WORD
EXTRN  _Register_AL:BYTE
EXTRN  _Register_AH:BYTE

EXTRN  _Entry_Point:WORD

_DATA  ENDS
END

;          *** Block Structure ***
;
;          1 Forward Pointer          2 bytes
;          2 Track Number             2 bytes
;          3 Sector Number             1 byte
;          4 Head Number               1 byte
;          5 Drive Number              1 byte
;          6 Status Word               1 byte
;          7 Buffer Area                512 bytes
;

```

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

```

_TEXT          SEGMENT      byte public 'CODE'
               ASSUME      CS:_TEXT,DS:_DATA

               EXTRN       _To_DOS:FAR

               PUBLIC      _Verify_Function
               PUBLIC      _Format_Function
               PUBLIC      _Check_Cache
               PUBLIC      _Get_PSP
               Public      _To_BIOS

```

96

```

;-----;
;               Verify Function
;-----;

```

```

_Verify_Function  PROC FAR
                  jmp  _Label010
_Verify_Function  ENDP

```

```

;-----;
;               Format Function
;-----;

```

```

_Format_Function  PROC FAR
                  jmp  _Label003
_Format_Function  ENDP

```

```

;-----;
;               Check Cache Function
;-----;

```

```

_Check_Cache      PROC FAR
_Label001:        MOV  AX,_Register_Flag
                  AND  AX,OFFFEh
                  MOV  _Register_Flag,AX
                  MOV  AX,_PSP_Segment
                  MOV  WORD PTR _Register_AL,AX
                  JMP  _To_DOS
_Check_Cache      ENDP

```

```

;-----;
;               Get PSP Segment
;-----;

```

```

_Get_PSP          PROC FAR
_Label002:        MOV  DX,_Old_Timer
                  MOV  AX,_Old_Timer + 2
                  MOV  DS,AX
                  MOV  AH,025h
                  MOV  AL,01Ch
                  INT  21h
                  MOV  AX,_DATA
                  MOV  DS,AX
                  MOV  DX,_Entry_Point

```

```

MOV AX, _Entry_Point + 2
MOV DS, AX
MOV AH, 025h
MOV AL, 013h
INT 21h
MOV AX, _DATA
MOV DS, AX
MOV AX, _Register_Flag
AND AX, 0FFFEh
MOV _Register_Flag, AX
MOV AX, _PSP_Segment
MOV WORD PTR _Register_AL, AX
JMP _To_DOS
_Get_PSP ENDP

_To_BIOS PROC FAR
;-----;
; Free Buffer Block Before Do Other Function ;
;-----;
_Label003: MOV DX, 0h
MOV BX, _Head
_Label004: CMP BX, 0h
JZ _Label010
MOV AL, [BX+6]
CMP AL, _Drive_Number
JZ _Label005
MOV DX, BX
MOV BX, [BX]
JMP _Label004
_Label005: MOV CX, [BX]
CMP DX, 0h
JZ _Label006
XCHG BX, DX
MOV [BX], CX
XCHG BX, DX
JMP _Label007
_Label006: MOV _Head, CX
_Label007: MOV CX, _Not_Used
MOV [BX], CX
MOV _Not_Used, BX

CMP DX, 0h
JZ _Label008
MOV BX, DX
MOV BX, [BX]
JMP _Label009
_Label008: MOV BX, _Head
_Label009: JMP _Label004
;-----;
; Set Register Data ;
;-----;

```

```
-----;
_Label010:      MOV AX,_Register_Flag
                PUSH AX
                MOV AX,CS
                PUSH AX
                MOV AX,OFFSET _TEXT:_Label011
                PUSH AX
                MOV AX,_Entry_Point + 2
                PUSH AX
                MOV AX,_Entry_Point
                PUSH AX
                MOV AX,_Register_DS
                PUSH AX
                MOV AX,WORD PTR _Register_AL
                PUSH AX
                MOV AX,_Register_ES
                MOV ES,AX

                MOV BX,WORD PTR _Register_BL
                MOV CX,WORD PTR _Register_CL
                MOV DX,WORD PTR _Register_DL

                MOV BP,_Register_BP
                MOV DI,_Register_DI
                MOV SI,_Register_SI

                POP AX
                POP DS
                RET

_Label011:      PUSHF
                PUSH DS
                PUSH AX
                MOV AX,_DATA
                MOV DS,AX
                POP AX

                MOV WORD PTR _Register_AL,AX
                MOV WORD PTR _Register_BL,BX
                MOV WORD PTR _Register_CL,CX
                MOV WORD PTR _Register_DL,DX

                MOV _Register_BP,BP
                MOV _Register_DI,DI
                MOV _Register_SI,SI

                POP AX
                MOV _Register_DS,AX
                MOV AX,ES
                MOV _Register_ES,AX
                POP AX
                MOV _Register_Flag,AX
                JMP _To_DOS

_To_BIOS      ENDP
```

```

_TEXT          ENDS

_DATA          SEGMENT word public 'DATA'

              EXTRN   _Head:WORD
              EXTRN   _Not_Used:WORD

              EXTRN   _Drive_Number:BYTE

              EXTRN   _PSP_Segment:WORD

              EXTRN   _Register_Flag:WORD
              EXTRN   _Register_CS:WORD
              EXTRN   _Register_IP:WORD

              EXTRN   _Register_AL:BYTE
              EXTRN   _Register_AH:BYTE
              EXTRN   _Register_BL:BYTE
              EXTRN   _Register_BH:BYTE
              EXTRN   _Register_CL:BYTE
              EXTRN   _Register_CH:BYTE
              EXTRN   _Register_DL:BYTE
              EXTRN   _Register_DH:BYTE

              EXTRN   _Register_BP:WORD

              EXTRN   _Register_DI:WORD
              EXTRN   _Register_SI:WORD

              EXTRN   _Register_DS:WORD
              EXTRN   _Register_ES:WORD

              EXTRN   _Old_Timer:WORD
              EXTRN   _Entry_Point:WORD

_DATA          ENDS
              END

;              *** Block Structure ***
;
;              1 Forward Pointer      2 bytes
;              2 Track Number        2 bytes
;              3 Sector Number       1 byte
;              4 Head Number          1 byte
;              5 Drive Number         1 byte
;              6 Status Word          1 byte
;              7 Buffer Area           512 bytes
;

```

\_TEXT SEGMENT byte public 'CODE'  
ASSUME CS:\_TEXT,DS:\_DATA

100

To\_DOS PUBLIC \_To\_DOS  
PROC FAR

```
-----  
:                                     :  
:                               Reset Lock                               :  
:                                     :  
-----
```

MOV AX,0h  
MOV \_Lock,AX

```
-----  
:                                     :  
:                               RETURN                               :  
:                                     :  
-----
```

MOV AX,\_Register\_Flag  
PUSH AX  
MOV AX,\_Register\_CS  
PUSH AX  
MOV AX,\_Register\_IP  
PUSH AX  
MOV AX,\_Register\_DS  
PUSH AX  
MOV AX,WORD PTR \_Register\_AL  
PUSH AX  
MOV AX,\_Register\_ES  
MOV ES,AX

MOV BX,WORD PTR \_Register\_BL  
MOV CX,WORD PTR \_Register\_CL  
MOV DX,WORD PTR \_Register\_DL

MOV BP,\_Register\_BP  
MOV DI,\_Register\_DI  
MOV SI,\_Register\_SI

POP AX  
POP DS

To\_DOS IRET  
ENDP

TEXT ENDS

\_DATA SEGMENT word public 'DATA'

EXTRN \_Timer:WORD  
EXTRN \_Lock:WORD

EXTRN \_Register\_Flag:WORD  
EXTRN \_Register\_CS:WORD  
EXTRN \_Register\_IP:WORD

```

EXTRN  _Register_AL:BYTE
EXTRN  _Register_AH:BYTE
EXTRN  _Register_BL:BYTE
EXTRN  _Register_BH:BYTE
EXTRN  _Register_CL:BYTE
EXTRN  _Register_CH:BYTE
EXTRN  _Register_DL:BYTE
EXTRN  _Register_DH:BYTE

```

```

EXTRN  _Register_BP:WORD

```

```

EXTRN  _Register_DI:WORD
EXTRN  _Register_SI:WORD

```

```

EXTRN  _Register_DS:WORD
EXTRN  _Register_ES:WORD

```

```

_DATA

```

```

ENDS
END

```

```

*** Block Structure ***

```

```

;
;
;
;
;
;
;
;
;
;

```

- |   |                 |           |
|---|-----------------|-----------|
| 1 | Forward Pointer | 2 bytes   |
| 2 | Track Number    | 2 bytes   |
| 3 | Sector Number   | 1 byte    |
| 4 | Head Number     | 1 byte    |
| 5 | Drive Number    | 1 byte    |
| 6 | Status Word     | 1 byte    |
| 7 | Buffer Area     | 512 bytes |

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย



\_TEXT

SEGMENT byte public 'CODE'

ASSUME CS:\_TEXT,DS:\_DATA

102

PUBLIC \_Time\_Routine

Time Routine

```
_Time_Routine PROC FAR
PUSH AX
PUSH AX
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH DS
PUSH BP

MOV AX, _DATA
MOV DS, AX

MOV AX, _Lock
AND AX, AX
JZ _Label001
JMP _Label010
_Label001: MOV AX, _Timer
AND AX, AX
JZ _Label002
JMP _Label009
```

Free Buffer Block Before Do Other Function

```
_Label002: MOV DX, 0h
MOV BX, _Head
_Label003: CMP BX, 0h
JZ _Label010
MOV AL, [BX+6]
AND AL, 080h
JZ _Label004
MOV DX, BX
MOV BX, [BX]
JMP _Label003
_Label004: MOV CX, [BX]
CMP DX, 0h
JZ _Label005
XCHG BX, DX
MOV [BX], CX
XCHG BX, DX
JMP _Label006
_Label005: MOV _Head, CX
_Label006: MOV CX, _Not_Used
```

```

MOV [BX],CX
MOV _Not_Used,BX

CMP DX,0h
JZ _Label007
MOV BX,DX
MOV BX,[BX]
JMP _Label008
_Label007: MOV BX,_Head
_Label008: JMP _Label003
;-----;
; Return ;
;-----;
_Label009: SUB AX,1
MOV _Timer,AX
_Label010: MOV BP,SP
MOV AX,_Old_Timer + 2
MOV [BP+14],AX
MOV AX,_Old_Timer
MOV [BP+12],AX
POP BP
POP DS
POP DX
POP CX
POP BX
POP AX
RET
_Time_Routine ENDP
_TEXT ENDS
_DATA SEGMENT word public 'DATA'
EXTRN _Head:WORD
EXTRN _Not_Used:WORD

EXTRN _Lock:WORD
EXTRN _Timer:WORD
EXTRN _Old_Timer:word
_DATA ENDS
END
;
; *** Block Structure ***
;
; 1 Forward Pointer 2 bytes
; 2 Track Number 2 bytes
; 3 Sector Number 1 byte
; 4 Head Number 1 byte
; 5 Drive Number 1 byte
; 6 Status Word 1 byte
; 7 Buffer Area 512 bytes

```



## ประวัติผู้เขียน

นายชวลิต จามิตรกุล เกิดวันที่ 2 สิงหาคม พ.ศ. 2503 ที่กรุงเทพฯ สำเร็จการศึกษาปริญญา วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า จากภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2525 เข้าทำงานที่กองติดตั้งเครื่องอุปกรณ์โทรศัพท์ทางไกล องค์การโทรศัพท์แห่งประเทศไทย ตั้งแต่วันที่ 1 มิถุนายน พ.ศ. 2526 วันที่ 4 มิถุนายน ถึงวันที่ 24 สิงหาคม พ.ศ. 2527 รับการฝึกอบรมด้าน Digital Microwave Radio และ Optical Fiber Communication ที่ NEC Tamagawa Taining Center และ NEC Yokohama Taining Center เมืองโยโกฮาม่า ประเทศญี่ปุ่น วันที่ 4 พฤศจิกายน ถึงวันที่ 18 ธันวาคม พ.ศ. 2534 เข้ารับการฝึกอบรมด้าน Exchange Planing ที่ Siemens Taining Center of Communication Networks เมืองมิวนิค ประเทศเยอรมันนี เข้าศึกษาปริญญา วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า(ระบบเชิงเลข) ตั้งแต่ภาคการศึกษาที่ 1 ปีการศึกษา 2530 ปัจจุบันทำงานอยู่ที่ส่วนดำเนินการโครงการ องค์การโทรศัพท์แห่งประเทศไทย

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย