

Management program for patients with clipper

Department of Therapeutic Radiology Korea Cancer Center Hospital

이효남 · 지광수 · 김재휴

Computer의 사용분야는 날로 변화하고 있으며 그 대표적인 것이 Hard Ware의 가격인하로 인해 많은 User들이 Computer와 가까워 질수 있는 계기를 많이 제공되고 있다. 하지만 SoftWare분야의 발전은 HardWare의 발전에 비해 전세계적으로 미약한 실정에 있다. 일반적으로 Computer언어의 있어서도 일반 User가 접근하기에는 많은 시간과 투자가 필요하게 된다. 하지만 Clipper는 적은 시간을 투자하여 친숙해 질 수 있는 Computer언어라 생각되며 이 Clipper는 Dbase계열의 매크로 기능과 유사성을 많이 가지고 있어서 각 병원 환경에 쉽게 CPP(Clipper Patient Program)를 변형하여 사용할 수 있게 만들어 보았다.

Management program for patients with clipper

Hyo Nam Lee , Kwang Soo Jee , Jae Hyu Kim

Department of Therapeutic Radiology
Korea Cancer Center Hospital

1. 환자관리의 효율성을 증대
2. 각 병원에서 쉽게 변형하여 사용할 수 있도록 제작함
3. 환자관리에 있어 필요한 항목을 나타내고자 함

Introduction

- **Computerization of patient care**
- **Standardization of computer input work for RT**
- **Simplification of transformation for patients' information**

클리퍼의 장점과 단점

장점

1. 모든 Dbase계열의 Soft Ware와 호환성이 좋다.
2. User가 접근하기 편리한 Computer언어이다.
3. 4세대 Computer언어와 유사성을 많이 가지고 있다.

단점 :

1. 시스템 제어에 문제점이 있고
2. 실행화일이 큰 관계로 인해 화면의 처리속도가 느리다.

Advantages of clipper as a tool

- **Good compatibility with all Dbase software**
- **Easy use by user**
- **Similarity to 4th generation language**

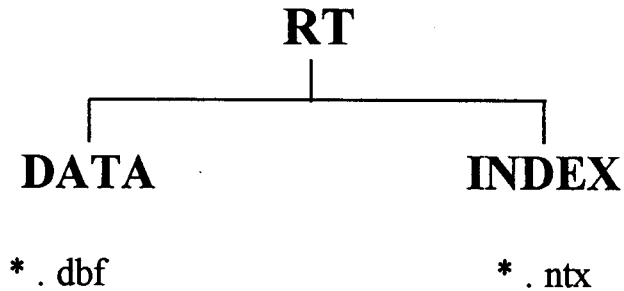
Project Performance in Environment

OS : DOS 6.0
System : 386 - DX2
Compiler : Clipper 5.01
Resolution : 640 x 480 256 Color
Graphic card : Pradise , ET4000
Memory : 4M
Editor : Q_edit 2.15

Main을 RT로 하고 두 개의 서브디렉토리로 되어 있다.

서브디렉토리 Data에는 각 각의 데이터를 저장하게 되어 있고 서브디렉토리 Index에는 데이터에 관한 Index를 보관하게 됩니다.

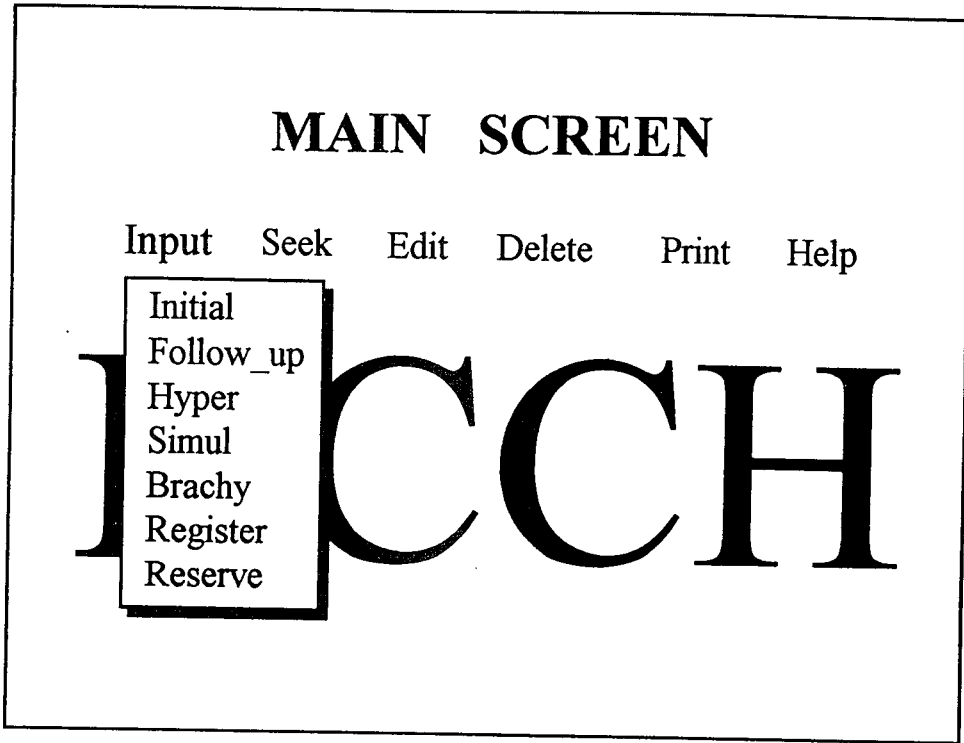
Directory Composition



RT . prg

```
#include "rt.ch"  
_DFSET ("yyyy . mm . dd", "yy . mm . dd")  
SET PATH TO C:\RT\DATA C:\RT\INDEX C:\RT  
win_color = " W+ / G+ , GR / R+ , , , W+ / G+ "  
USE PASSWORD  
IF ! FILE (" password . dbf ")  
    Creat_pass ()  
ENDIF  
SPLIT_SCR (), JOIN_SCR (), RT_INI ()
```

RT.prg는 메인 프로그램으로써 Pop-up menu를 구성하는 부분과 Include file, Path, 화면전체에 적용될 Color를 정의하며 환자관리에 대한 자료의 보안성을 유지하기 위해 Password를 체크하고 Password의 변환을 가능하게 하기 위해 Creat password부분으로 구성되어 있습니다. 또한 서브프로그램을 호출하는 RT-Inl() Function을 가지고 있음.



CPP는 입력부분으로 환자의 기본적인 자료를 입력받는 Initial data과 Follow-up data로써 구성되어 있고 기타 부분으로 온열치료의 자료를 입력받기 위해 Hyperthermia, Simulation의 자료를 저장하기 위해 Simulation, 근접치료의 자료를 입력받기 위해 Brachytherapy, 환자의 예약관계에 필요한 데이터의 보관을 위해 Register과 Reserve부분으로 서브 디렉토리가 구성되어 있다. 또한 각 각의 입력 항목에 대해 Seek, Edit, delete, Print등의 서브 루틴을 가지고 있음. 프로그램의 종료는 Esc Key를 동일하게 사용하고 있습니다.

Initial data에 필요한 항목은 37개로써 구성되어 있으며 환자의 기본적인 자료를 입력하는 부분과 임상(Pathologic)에 필요한 자료항목 그리고 방사선 치료(Radiation)에 관계되는 항목등으로 구성되어 있다.

환자들의 기본적인 자료 입력을 위한 Initial, Follow-up, 온열치료의 자료입력을 위해 Hyperthermia, 치료계획을 위한 simulation 강내조사에 관한 자료 입력을 위한 Brachytherapy, 환자의 예약을 위한 Register, Reserve로 입력자료 화면이 구성되어져 있다.

Initial Screen

Unit_no. Name. Jumin_no. Address. Comment . TNM
RT_start. RT_finish. Telpone. Diagnosis. Age. Sex
Dpartment. Machine_no. Referental_no. Operation
Ward. Base of follow up. Organ(specify). Organ(code)
Histology(code). Histology(specify). Primary Organ(code)
Primary Organ(spec). Pathologic grade. Tre_dose
Initial DX. Performance. Multi_prim Organ. Previous tx
D_Previous TX. Combination TX. AIm of RT. Tre_vol
Mod_of_RT. Res_to_RT. Compli TX

Organ Seek1

```
LOCAL ret := .T. , fine_dbf := "WHO.DBF"

IF !FILE(find_dbf)
    NON_DBF(11,19,13,53,find_dbf)
    DBCLOSEALL()
    RETURN(ret)
ELSE
    Iorgan_s = WHO -> CONO
    RETURN ( ret )
ELSE
    ERR_MESAG("F3
search")
ELSE
    ret = .F.
    Iorgan_s = SPACE(25)
    IF EMPTY(Iorgan_c)
        ERR_MESAG(" F3 Search")
        ENDIF
        ret = .F.
        ENDIF
        Iorgan_s = SPACE(25)
        ENDIF
    ELSE IF WHO -> (DBSEEK(Iorgan_c))
```

Organ Seek2

```
DO WHILE .T.
  WHO ->(DBGOTOP())
  DO WHILE WHO -> (!EOF())
    AADD (code_ary, WHO -> code + " " + WHO -> cono)
  ENDDO
  code_choice = ACHOICE (13,11,21,36,code_ary,.T., "USE_KEY", 1)
  IF LASTKEY () = ESC
    RETURN (.T.)
  ELSE
    Iorgan_c = SUBSTR (code_ary [code_choice], 1, 5)
    Iorgan_s = SUBSTR (code_ary [code_choice], 7, 25)
    KEYBOARD CHR(ENTER)
    RETURN (.T.)
  ENDDO
```

WHO 코드를 개별적으로 입력하는 것은 많은 시간을 소비하는 요인이 될 수 있으며 환자관리 효율성을 감소시키는 문제를 발생시킬 수 있습니다. 이 문제를 해결하기 위해 Seek1, Seek2를 만들어 제공합니다. Seek1은 환자의 질병코드를 알고 있을때 질병코드를 이용하여 진단명을 Search하는 방법과 질병코드와 진단명 각각을 알지 못할때 Pop-up Menu를 이용하여 검색할 수 있는 방법을 제공하고 있음. WHO코드는 Dbase(Software)를 이용하여 삽입과 삭제를 가능하도록 만들어 보았습니다.

Follow - up Screen

Unit_no . RT_no . Department . Name . Jumin_no
Ward . Base of follow - up . Machine . Address
Date, last follow - up . Period of follow - up
Start at follow - up . Date, Diag of Recurrence
Date, Distant Metastasis . Treatment After recurrence
Complication After TX . Complication After TX(spec)
Comment . next follow - up

Simulation Screen

Unit_no . RT_no . Department . Name . Jumin_no
Ward . Machine . Address . Organ (c) . Organ (s)
Diagnosis . TNM classification . Performance
Stage . initial TX . AIM of RT . Previous TX
TX site . Dose (cGy / f) . Energy . SSD
Total Dose (Gy / wk) . F _ size ()

Hyperthermia Screen

Unit_no . RT_no . Department . Name . Jumin_no
Ward . Machine . Address . Organ (c) . Organ (s)
Diagnosis . TNM classification . Performance
Stage . initial TX . AIM of RT . Previous TX
Power . Heating time (min) . Temperature
Microwave
1.External. G_ channel. Applicator
2. I_antenna . Numer . Channel
Ultrasound
1.External . Application . Freperature

Brachytherapy Screen

**Unit_no . RT_no . Department . Name . Jumin_no
Ward . Machine . Address . Organ (c) . Organ (s)
Diagnosis . TNM classification . Performance . Stage
Source . Applicator . Anesthesia . Date Applied
Time Applied . Date Removed . Time Removed
Total amount (mCi) . Applied time (hours)
Reference depth . total activity . Seed activity
Active length . Ribbon No . Total dose (point A)**

Register Screen

**Unit_no . RT_no . Department . Name . Jumin_no
Ward . Machine . Address . Organ (c) . Organ (s)
Diagnosis . TNM classification . Performance
Stage . initial TX . AIM of RT . Previous TX
RT _ time . RT _ start . MU**

Reserve Screen

Unit_no . RT_no . Department . Name
Jumin_no . Ward . Machine . Address
Organ (c) . Organ (s) . Diagnosis .
TNM . Performance . Stage
Date , Simulation . Time , Simulation . Energy

Utility . prg

Creat _ ntx () . Creat _ ini () . Creat _ hye ()
Creat _ dbf () . Creat _ din () . Creat _ dye ()
Index _ msg (filename , indexkey , block)
Warning _ msg (filename) . Pass _ change ()
PassLock (pass) . PassUnLook (pass)
Creat _ Pass ()

Utility-prg는 각각의 입력 데이터에 대한 데이터파일과 인덱스화일을 생성시키는 Function 부분과 Password를 데이터 파일의 생성 및 검색하는 function 및 Password의 변환에 필요한 Pass-change(), 그리고 인덱스화일에 대한 에러체크를 수행하는 프로시저 함수들로 이루어져 있다.

Function . prg

```
Ntx _ dir ( path , indexkey , block ) . Beep ( )  
Use _ key ( mode ) . Shadow ( t , l , b , r )  
Beel ( ) . Err _ mesag ( message ) . Mark _ chk ( )  
Non _ dbf ( t , l , b , r , file ) . Allselect ( )  
Yo _ il ( row , col , data ) . Yes _ No ( message )  
Window _ Ary ( t , l , b , r , array , key )  
Unselect ( ) . Choice _ Key ( mode , ele , rel )  
Sound1 ( ) . Percent ( iNg )
```

Function.prg는 데이터파일 및 인덱스 화일의 디렉토리를 검색하는 Ntx-dir(), 데이터파일 및 인덱스 화일의 검색 결과 실패하는 경우 경고음을 발생시키는 Beep(), Beel(), Sound1() 및 화면에 경고 메시지를 표시하는 Err-mesag(), 데이터화일이 존재하지 않는 경우의 에러체크 함수 Non-def(), 현재의 날짜를 체크하는 Yo-il(), 화면에 나타내는 Pop-up Screen에 Shadow를 표시하기 위한 Window-Ary(), 삭제 작업에 필요한 마크표시를 위해 Select(), 마크표시를 제거하기 위한 Unselect(), 삭제 작업의 진행정도를 화면에 보여주기 위한 Percent() 함수로 이루어져 있으며 프로그램의 저장이나 삭제 등의 결정에 필요한 Yes-No() 함수등이 있다.

Seek.prg는 두 개의 루틴을 수행하고 있으며 SEEK-POINT()는 환자의 Unit-no 대하여 개별적인 Search를 수행하도록 만들어졌으며 TB()는 한 화면에서 전체 환자의 Search가 가능하도록 만들어 놓았다. TB()는 데이터 베이스의 Browse 기능과 같은 역할을 하도록 제작하였다.

Seek . prg

```
DO WHILE !looping
  __ ATPROMPT ( 05 , 30 , Individual )
  __ ATPROMPT ( 06 , 30 , All Seach )
  MENU TO choice
  DO CASE
    CASE choice = 0 .OR. LASTKEY () == ESC
      DBCLOSEALL () // Exit
    CASE choice = 1
      SEEK _POINT () // Individual
    CASE choice = 2
      TB () // Browse
  ENDCASE
ENDDO
```

Edit . prg

```
DO WHILE !looping
  IF RT -> (DBSEEK ( Iunit_no )) // Index key
    Iunit_no := _FILE -> unit_no ; Iname := _FILE -> name
    @04 , 19 GET Iunit_no PICT "99999999" VALID S_C ()
    READ
    IF ( YES_NO ( " save " ))
      GO rec_point
      _FIELD -> unit_no := Iunit_no // Save
    ENDIF
  ENDIF
ENDDO
```

수정작업을 하기 위한 Edit.prg는 개별적인 수정작업을 수행한다. 환자의 Unit-no에 의하여 수정작업이 이루어지고 수정된 데이터는 Yes-No() 함수에 의해 다시 저장하는 루틴으로 되어 있다.

Delete . prg

```
DO WHILE !looping
  choice = ACHOICE(05,40,18,75, choice_ary, .T., "CHICE_KEY" ,nele , nrel )
  DO CASE
    CASE LASTKEY() = k_SPACE // Space key
      IF SUBSTR(choice_ary[choice], 2,1) == " "
        choice_ary[choice] := STUFF (choice_ary[choice], 2,1, MARK)
      ELSE
        choice_ary[choice] := STUFF (choice_ary[choice], 2,1, " ")
      ENDIF
    CASE LASTKEY() == F5 // F5 All select
      ALLSELECT ()
    CASE LASTKEY() == F6 // F6 Unselect
      UNSELECT ()
    CASE LASTKEY() == F8 // F8 Excute
      DELEETEEXE ()
      looping := .T.
  ENDCASE
ENDDO
```

Delete.prg는 개별적인 삭제작업과 전체를 동시에 삭제작업을 수행할 수 있는 루틴을 제공한다. Space bar는 Por-up Screen에서 삭제하고자 하는 데이터를 선택할 수 있고 F5Key는 전체 데이터에 대하여 Pop-up Screen에서 선택할 수 있게 만들어 보았다. 동시에 삭제 작업을 수행하는 경우에는 4050개 이하의 데이터를 동시에 삭제 작업을 할 수 있도록 만들어 놓았다. Unselect() 함수는 삭제 작업을 하기 위해 선택한 데이터를 해제시키기 위해 제작되었으며 실질적인 삭제작업은 Deleteexe() 함수에 의하여 이루어진다. 삭제작업의 편리성을 제공하기 위해 각 각에 대하여 Hot Key(F5, F6, F8)를 제공하고 있다.

삭제작업을 수행하기 위한 Deleteexe() 프로시저는 세개의 부분으로 구성되어 있습니다. 첫째로 삭제 표시가 있는가 없는가를 체크하고 둘째로는 삭제작업을 하기 전에 데이터의 백업을 수행하는 부분이 있다. 그 이유는 환자의 데이터는 현재 작업 영역에서는 삭제되어져도 환자의 자료 보관이 필요하게 됩니다. 또한 환자의 생존율이나 통계에 필요한 자료로 남아 있어야 합니다. 그래서 -DEPACK 함수를 호출하기전에 데이터의 백업이 수행되어 집니다.

백업작업이 수행된 이후에 - DEPACK 함수에 의해 삭제가 이루어 진다.

Deleteexe

```
FOR i := 1 TO LEN (choice_ ary)
  IF SUBSTR (choice_ ary [i], 2, 1) == MARK
    value := SUBSTR (choice_ ary [i], 5, 5)
    IF DBSEEK (value)
      RTBAK -> (DBAPPEND ())
      RTBAK -> unit_no := _FILED -> unit_no
      RTBAK -> name := _FILED -> name
      RT -> (DBDELETE ())
    ENDIF
  ENDIF
NEXT
RT -> (__DBPACK)
SOUND1 ()
```

Print . prg

- **All Printing**
 - Inicial_p (). Hyp_p (). Bra_p (). Sim_p ()
 - Follwp_p (). Regisrt_p (). Reserve_p ()
- **Part Printing**
 - Machine_p (). Check_p (). Telephon_p ()
 - Sim_r_p ()
- **Statistic**
 - Mon_m_p (). Year_m_p (). New_o_p ()
 - New_y_p (). Code_p (). Ward_p ()

Print.prg는 각 각의 입력에 대한 출력부분인 All Printing과 기계별, 환자의 예약관계, Simulation 예약관계, 치료기의 고장시 환자에게 연락이 필요한 경우를 위한 Telephon-p(), 환자의 질환과 관계한 출력 등등을 Part printing에서 출력하고 환자의 월별 또는 년도별 통계, 병실통계, 질환코드통계, 신환통계, 등을 Statistic에서 출력할 수 있게 되어 있다.