Research Note

Seat-Number Scanning Program "do" in the Computer Rooms

Mikio Kanokogi

(Received on May 10, 2001)

I The Environment and Facilities in HSU

As is well known the computer languages "sed" (stream editor) and "awk" collaborated by Brian V. Aho, Peter J. Weinberger and Brian W. Kernighan by 1985 belong to rather older ones preceding to "C" language that has become for itself already nowadays classic in the "IT Age". Nevertheless, especially the languages "awk"[1] can still deal with the UNIX system in our university and effective for solving some necessary jobs to the teaching staff. The following report might be one of the poor recommendations for the colleagues from an amateur person, whose original research field is the Mediävistik of Germany.

The UNIX system in the Computer Center of HSU is not always familiar either with teaching staff or with students, mainly because of its traditional engineer-oriented architecture, but partially because of its bilingual or mixed expresseions in the display of both Japanese and English in the Computer Center HSU in its responses for the commands "date", "last", "man", etc. So, only few persons in the laboratories select the "Tera Term" icon in order to use the various functions of the UNIX system. As one of the teaching staff of the computer classes, the author has made effort often to convey some hints to the students to utilize some fascinating functions, but almost in vain. Comparing with some "soft" wares just like as "Al Mail" or "Eudora", its e-mail sending or receiving way with ".mailrc" file is one of the most complicating one, especially because some skill to use the already too classical "vi"(sual) editor is indispensable for the purpose.

Nevertheless, only when the user accesses the UNIX system by selecting "Tera Term" icon, there can be restored the user's account, the intermediate machine, the laboratory and seat-number (TCP/IP), log-in and -out time or "still logged in" status, etc. in the "last" file in the Center. As far as the hearings or interviews of the author at the Computer Center

1) Ver. XCU5.0 bound to the Compaq Tru64 UNIX V5.0A, Rev. 1094 of HSU.

concerned, the file "last" is the only log-book opened for every interesting user for the purpose. Thus, the author made an "awk" program for an automatic roll book checking in the last schoolyear 2000, about which he contributed a research note to this "the Report of the Economic Sciences", vol. 4, no. 2.

After contriving an automatic roll book program with shift-back funtion named "table" in the last schoolyear, the author has improved it for all the teaching staff in the computer rooms as the name of "do", by enlarging its adaptability to every laboratory, by supplementing a "late commer" symbol for the student who comes later than the informed/assigned time and by enabling a new display according to the seat-number order. In any way, the students are not urged to sit at some fixed seat, if not previously assigned, but they can choose or change their seat freely by the teaching-staff's order or by their own will. In other words, this "free" organized, reorganized or unorganized class making idea is reflected the philosophy of practical education of the author for variously skilled students in the "IT" age.

II Two Features of the Language "awk"

As far as the studies and practices of the author concerned, the language "awk" shows the following two elementary features. The author would like to explain them not as a system, but rather to report how he has made use of them in his programming.

a) The Specific Notion of the "Pattern"

Succeeding to the preceding editor language "sed", the "awk" seeks or chase some definite patterns in the log-in file provoked by the command "last", on the one hand. This access-record in HSU has such bilingual form as folows:

```
kanokogi  pts/2   rm549              木  9月 4日  16:30   still  logged in
okamura   pts/6   pc_18.bl.shudo-u  木  8月31日  15:05 - 16:03   (00:57)
```

The meanings of the items are clear: the log-in name, the intermediate machine, log-in point, the day of the week and the month and the day (in Japanese) and the time record or situation. For the "awk" eye nevertheless, these two lines seems quite different. While the first line has ten "fields", the second one has only nine "fields", because there is no space-"separater" between the month and the day. If somebody can 'teach' the "awk" the new separater notion of "月日" (month and day) during the "last" command, the difference of the

number of the "fields" can be ignored. But because of the doubled usage of "日" and "月" in the day-of-the-week item, the author could not find the adequate remedy to do it. Perhaps, in this 'confession of trouble', you can find the notion of the "pattern" of the "awk", that is consist of both the "fields" and the "separaters".

b) The Function of the "Associational Array"

When the author reports about the second feature of the "awk" on the other hand, he has to single out a kind of mistranslation in the Japanese "awk" manuals[2]. For the word "association" or "associational" they are used to adapt the psychological term for it. This means a kind of some trouble or embarrassment for us. By the way, the array function is very usefull. As parameters or value, we can use both numerical values and strings freely. The "awk" arrays can "associate" or bind two words or notions easily.

The labo-code and time factors are the key words for scanning work in the accessed records called by the command "last". After finding the adequate labo-code at the top of the third column ($3) at the right time of the day, the "awk" makes an associational array "seat", whose parameter is the log-in name ($1), and sets the succeeding two numerics (the seat-number from 01 to 99) or other code (e.g. from a01 to f06) as its value: Thus each scanning record as the array "seat[log-in name] = seat-number" is gained.

At the next stage, the "awk" compares these associational arrays "seat" with the class list of the user. For the purpose, the "awk" users in HSU must be confronted with difficulties regretfully. Though the conditional expression "if (parameter in array)" is widely recommended, we cannot use the conditional as "if (log-in name in seat)" in the UNIX. So the author had to seek to find another expression, that could be inferred by analogy with the "C" language grammer. The conditional "if (seat[log-in name] != "") was a solution, whose meaning is "if seat[log-in name] is not nullstring". The auther can be a little delightful for this finding or study.

III The Pattern of the Register Files

The author proposes a design for the roll book, but he knows well that his proposal is one

2) A. V. Aho, B. W. Kernighan and P. J. Weinberger, *The AWK Programming Language*, 1988, AT&T Bell Laboratories: transl. into Japanese by Takanori Adachi, published 1989, Toppan. Shinsuke Sasaki, *Nihongo AWK Puroguramingu Tekunikku,* Mainichi Komyunikeeshonzu, 1991

of the possible display designs for the users, that is attached at the end of the program. He can accept any criticism or proposals from the users willingly.

a) The Possible Ways to Get the Program with the File Pattern

As the program exists in the root directory of the author in the readable mode, anybody can copy it easily, by using the direct command "cp" (copy) in UNIX or through some software like "FTP". Or the author can send it, by the e-mail, as the "awk" text is described in pure "text" style. Or he can also give it to the user in a floppy disk, that can be easily edit for the purpose, before loading it up in the users directory. But in every case, the user should give it the "execute" mode before use, e.g. by "chmod" (change mode) command in UNIX or somehow else.

b) How to Make the Register File of the User's Class (es)

In any way, as the specially patterned register and the log-in names of the students cannot be afforded from the Academic Section (Kyomu-ka) of HSU, the user should make trouble for him/herself.

For the log-in names, the user of this program can ask the class students what are their log-in names. But another way is to use the "finger" command in UNIX. If the user types "finger student-code", there can be called the registered log-in name of the student.

For the register, the user should be urged to be acquainted with the notions "fields" and "separaters" of the "awk" language a little. The proposed design of the roll book is the tabulator separated type. And the 1st headline(s) should not have any "separater" (tabulator). The 2nd column line has a long blank on the left, by 5 "separaters" (tabulators), and the 6th "field" ($6) for counting of the attendances and the class dates. After the 3rd line, there should come the list of the class students: the 1st "field" ($1) for the student-codes, the 2nd "field" ($2) for the student names, the 3rd "field" ($3) for the log-in names of the students and the 4th "field" ($4) for the number of attendances.

To obtain some satisfactory result after the 3rd line, the user should be aware of the meaning of "separaters" (tabulators) between the 1st "filed" ($1) for the student-code and the next "field" (#2), "separater" point between the 2nd "field" ($2) for the student name and the 3rd "field" ($3) for the log-in names and the length of the 3rd "field". If the 2nd "field" ($2) is shorter to override the tabulator point in between, the user should append space(s) enough to override it. And if the length of the length of the 3rd "field" ($3) is shorter than 8, the user

should append space(s) at the end. Then, at the top of the 4th "field" ($4), there should be written 2 characters as " 0" for the total of attendances.

```
Headline:   Class Name, Class Day-of-the-Week, Class Room, etc.
|       |       |       |       |       Σ
yyffnnn Student-Name    Log-Name        0
```

IV The Program File "do"

Compared with the former version[3], the new one can be rather enriched with more useful functions as following, that has been envisaged from the kind advices of colleagues or from the practices of the author in the classes[4].

a. Generalization Formerly, the program was designed for the 9 class rooms in the 6th building, where the DOS/V CPU computors are equipped. But HSU has other facilities even in the 6th building itself, for example, Macintosh machine or workstation rooms and moreover multi-media rooms in the 1st and 2nd buildings. Thus the lengths of room-codes vary between 3 (e.g. pc1 or mac) to 8 (kyosyoku) and at the same time seat-numbers are not limited from 01 to 99. In the 1st multi-media room 2303 the seat-number from a01 to f06 are used.

b. Two Way Display The author doesn't choose the fixed seat system in the classes, where the students take their seats freely. In such cases, the first display of the presenting situation might be better in sorted order according to the seat order. But for the registration to the original register book order might be more convinient. So the author prepares two scenes to display the results: one in the sorted order and another in the original register order, switching by the same key Backspace.

c. Later Commers Check In the practical teaching, the later commers are often confusing factor for teaching, such students should be given perhaps some penalty. For such possible educational consideration, the author gives delay symbol "–" next to the seat-number of each student. The symbol "–" can have psychological warning effect for the later commers. Though 15 minutes delay is given as default value of "dd" in the program, it can be altered freely according to the teaching principle of each user.

---

3) Mikio Kanokogi, Programming for an Automated Roll Book at the Computor Classrooms – Using a Pattern Scanning and Processing Language 'awk' –, *Journal of Economic Sciences*, vol. 4, no. 2 (Feb. 2001) pp. 169–183.

4) In the program the Japanese symbol "¥" is used for "\" in the universal character set.

```
awk '# original program file: "table" 2000.09.19～10.08
     # 1st rev. program file: "do"    2001.04.08～05.07
     # 2nd rev. program file: "do"    2001.05.29～06.17 (c) kanokogi
     #   The register pattern by the author is attached at the end.

#1-1 parameters by the author 年月日・曜日・時刻データ
BEGIN { OFS = "¥t"
        "date +¥"%c %a¥"" | getline     # read date, week and time
        split($1,date,"[年月日]")       # kanji as separaters
        split($2,time,"[時分秒]")       # kanji as separaters
        week = $3                       # kanji weekday
        item = date[2] date[3]          # date item for subheading
        date[2] += 0; date[3] += 0      # numeralizing
        month = date[2] "月"            # kanji month
        day = date[3] "日"              # kanji day
        now = time[1]*60 + time[2]      # present time
        x = 1                           # for sorting cancell

#1-2 prepared parameters for users 使用者のための定数
        ab = "欠"                       # kanji absent symbol
        ab = "=="                       # for "欠" mask this line
        sum = "計"                      # kanji total symbol
        sum = "Σ"                       # for "計" mask this line
        w1 = "月"; w2 = "火"; w3 = "水" # Mon, Tue, Wed
        w4 = "木"; w5 = "金"; w5 = "土" # Thu, Fri, Sat
        p1 = 545; p2 = 645; p3 = 785    # p1～7 are periods, where
        p4 = 885; p5 = 980; p6 = 1075   # -5 min. latidude are
        p7 = 1165                       # given for early commers

#1-3 codes of class rooms assigned by the center 教室のコード
# room          code            notice
#-------------------------------------------------
# 1101          kyosyoku
```

```
# 1102          psyx
# 1201          kpc             seat number: to 51
# 1203          kpc             seat number: from 52
# 2103          rm2103
# 2303          mul3            1st multi-media room
# 2304          mul4            language laboratory
# 2309          mul9            2nd multi-media room
# 6302          pc1
# 6303          pc2
# 6304          pc9             former 6301 room
# 6305          pc10            former 6301 room
# 6401          pc3
# 6402          pc4
# 6403          ws              workstation room
# 6404          mac             Macintosh room
# 6405          pc5
# 6406          pc6
# 6407          pc7
# 6408          pc8             former material room
#   seat number of teacher: 99 (except for 6408 room)

#1-4 free parameters for user: example 使用者の指定変数 [一例]
# (user has to edit here for the purpose 使用者の記入部分１)
        dd = 15                         # allowed min. for late commer
                                        # accordint to the principle!
        d1 = w1                         # class day of the week
        t1 = p4                         # class period
        r1 = "pc4"                      # class room code  cf. above
        f1 = "sogo"                     # file name
# for 2 classes or more of user: example 使用者の指定変数 [追加の例]
# (user has to add parameter(s) if necessary 使用者の記入部分２)
        d2 = w3; d3 = w4                # additional class day(s)
        t2 = p4; t3 = p2                # additional class period(s)
```

```
        r2 = "pc6"; r3 = "mul3"          # additional class room(s)
        f2 = "hist"; f3 = "goho3"        # additional file name(s)

#2-1 curriculum of the user 使用者のコンピュータ授業
#2-1-1 for class 1  第1の授業 [運用例]
    if ( week == d1 && t1 <= now && now < t1 + 95 )
        { labo = r1; c = length(r1); file = f1; subf = f1 ".sub"
        monf = f1 ".mon"; bt = t1; dt = t1 + 5 + dd }
# [Mon, 4th period, rm 6402, "joho"  月 4限 6402教室 情報化社会と人間]
# ------------------------------------------------------------------
#2-1-2 for class 2  第2の授業 [運用例]
    else if ( week == d2 && t2 <= now && now < t2 + 95 )
        { labo = r2; c = length(r2); file = f2; subf = f2 ".sub"
        monf = f2 ".mon"; bt = t2; dt = t2 + 5 + dd }
# [Wed, 4th period, rm 6406, "hist"  水 4限 6406教室 歴史ゼミナール]
# ------------------------------------------------------------------
#2-1-3 for class 3  第3の授業 [運用例]
    else if ( week == d3 && t3 <= now && now < t3 + 95 )
        { labo = r3; c = length(r3); file = f3; subf = f3 ".sub"
        monf = f3 ".mon"; bt = t3; dt = t3 + 5 + dd }
# [Thu, 2nd period, rm 2303, "goho3"  木 2限 2303教室 語法研究III]
# ------------------------------------------------------------------
# sections from #1-2 to #2-1-3 should be enlarged, deleted or masked by
# user, expert personnel or the author for practical use in the classes
# #1-2～#2-1-3 部分の拡張・削除・マスクには "vi","mule"などUNIX上で使え
# るエディタによりますが 詳細は著作者またはセンター職員に尋ねてください
# ------------------------------------------------------------------
#2-2 for test use of this program  指定授業以外の日時の使用
    else
        { labo = "pc1"; c = 3; file = "list"; subf = file ".sub"
        monf = file ".mon"; bt = p3; dt = p4 + 15
        month = "9月"; date[3] = 4; day = "4日" }
# test site works except for the day(s) and period(s) of the user above
```

```
# 指定授業以外の日時にプログラムを運用したときに動作するルーティン
# ------------------------------------------------------------------

#2-3 program delay message プログラム作動の遅れ表示
        printf("  At the first check, wait for max. 15 sec. ......")
#       printf("  使用の始めに....15秒ほど..待つ場合があります....")

#3-1 read "last" data before 10th of month (NF=10) "last"検索(日付1桁)
  if ( date[3] < 10 ) {
    while ("last" | getline)
    if ( substr($3,1,c) == labo && $5 == month && $6 == day)
        { s = substr($3,c+1)                              # scan seat
        ss = split(s,sss,"."); s = sss[1]                 # net number!
        w = length(s)                                     # seat code
        split($7,l,":"); lt = l[1]*60 + l[2]              # log-in time
      if (bt < lt && lt <= dt) { seat[$1] = s " " }       # right commer
      else if (dt < lt) { seat[$1] = s "-" } }            # later commer
        }

#3-2 read "last" data after 10th of month (NF=9) "last"検索(日付2桁)
  else {
    while ("last" | getline)
    if ( substr($3,1,c) == labo && $5 == month day )
        { s = substr($3,c+1)                              # scan seat
        ss = split(s,sss,"."); s = sss[1]                 # net number!
        w = length(s)                                     # seat code
        split($6,l,":"); lt = l[1]*60 + l[2]              # log-in time
      if (bt < lt && lt <= dt) { seat[$1] = s " " }       # right commer
      else if (dt < lt) { seat[$1] = s "-" } }            # later commer
        }

#3-3 check the bottom of the "last" file ファイル"last"末の点検
        system ("clear")
```

```
        print "  ¥"last¥" bottom data → " $0          # wtmp of "last"
        if ( $4 != "8月13日" && subf == "list.sub" )
        { printf("  ¥"last¥" file has been renewed. The test result")
        print " can not been displayed normally."
#       { printf("  ¥"last¥"ファイルが更新されたので, ¥"list¥"の試験")
#       print "結果が正しく表示されません。"
        exit }

#4 copy file to subfile if item differs 別日付のとき保存ファイルを更新
        { FS = "¥t" }                               # renewed FS
  while ( (getline < file) > 0 )
    if (NF == 6)                                    # read subheading
        { z = split($6,it," ")
        if ( it[z] != item ) {                      # if latest item differs
        system( "cp " file " " subf )               # copy file to subfile
        close(subf) }
        }

#5 (re)write sum and seat-number/delay/absent data into file データ記入
  while ( (getline < subf) > 0 && $0 != "" )        # avoid last line
    if (NF == 1)                                    # write heading
        { print $0 > file }
    else if (NF == 6)                               # add new item
        { print $0 " " item >> file }
    else if (NF == 4) {                             # add new data
        split($3,n," "); name = n[1]                # strip login name
      if (seat[name] != "") {                       # seat array value
          p = substr($4,1,2); ++p                   # sum increment
          q = substr($4,3)                          # former queue
          s = seat[name]                            # seat data
        if ( w == 2 ) {
          printf("%s¥t%s¥t%s¥t%2d%s%2s%-3s¥n",$1,$2,$3,p,q," ",s) >> file }
                                                    # (re)write seat data
```

```
        else if ( w > 2 ) {
          printf("%s¥t%s¥t%s¥t%2d%s%s%-4s¥n",$1,$2,$3,p,q," ",s) >> file
        } }
      else print $0 "  " ab " " >> file          # (re)write absent symbol
        }
        close(file)

#6 sort the latest seat/late/absent data 当日データ対象のソート
   while ( getline < file )
        if ( NF == 4 )                          # data lines
        { z = split($4,k," ") }                 # number of data
        close(file)
        system ("sort -n +" z+2 " -" z+3 " " file " > " monf)
        close(monf)

#7-1 disply common heading ソート表示：見出し
        while ( (getline < monf) > 0 && $0 != "" )
    if ( NF == 1 )                              # print heading
        { print }

#7-2 display sorted register within 7 weeks  ソート表示(7週以内)
    else if ( NF == 6 && split($6,j," ") < 9 )  # count fields
        { print }
    else if ( NF == 4 && split($4,k," ") < 9 )  # seat/late/absent data
        { print }

#7-3 display sorted register more than 7 weeks  ソート表示(7週以上)
    else if ( NF == 6 ) {
        printf ( "¥t¥t¥t¥t¥t" )                 # print 5 tabs
        z = split($6,j," ")
        printf sum                              # print sum symbol
        for ( i = z-6; i <= z-1; ++i )          # limit date item range
                printf " " j[i]
```

```
        print " " j[z] }
    else if ( NF == 4 ) {
        printf ("%s¥t%s¥t%s¥t",$1,$2,$3)        # print fixed data
        z = split($4,k," ")
        printf ("%2d",k[1])                     # print sum
          if ( w == 2 ) {                       # seat/late/absent data
            for ( i = z-6; i <= z-1; ++i )
                printf("%2s%-3s"," ",k[i])
            printf("%2s%-3s¥n"," ",k[z]) }
          else if ( w > 2 ) {
            for ( i = z-6; i <= z; ++i )
                if ( k[i] == ab ) { k[i] = " " k[i] }
            for ( i = z-6; i <= z-1; ++i )
                printf("%s%-4s"," ",k[i])
            printf("%s%-4s¥n"," ",k[z]) } }
        close(monf)

#7-4 prompt for another display and message 表示変更のプロンプト
    if ( z > 8 ) {
        print "  From: " j[z-6] ", To: " item }
#       print "  表示始め = " j[z-6] ", 表示終り = " item
    if ( x == 1 )
        { printf("  Sorting can be cancelled by Backspace or Ctrl+H ")
        printf("key (q to quit). ") }
#       printf("  Backspace(またはCtrl+H)キーでソーティングを解除でき")
#       printf("ます(q to quit)。") }
    else
        { printf("  Viewrange can be changed by Backspace or Ctrl+H ")
        printf("key (q to quit). ") } }
#       printf("  Backspace(またはCtrl+H)キーで表示の範囲を変えられ")
#       printf("ます(q to quit)。") } }

#8-1 scan keyboard input プロンプトへの入力キー
```

```
    $0 ~ /^([qQ]|quit)$/ { exit }
    $0 != "" {
        if ( $0 == "¥b" )                          # scan BS key
                { --x
                system ("clear")

#8-2 display standard register within 7 weeks 出席簿の表示(7週以内)
    close(file)
  while ( (getline < file) > 0 && $0 != "" )
  if ( z < 9 ) {
    if ( NF == 1 ) { print }                       # print heading
    else if ( NF == 6 ) { print }
    else if ( NF == 4 ) { print }
    }
  if ( z < 9 ) { exit }

#8-3 display partial register with scrolling 出席簿の部分表示(8週以上)
    close(file)
  while ( (getline < file) > 0 && $0 != "" )
    if ( NF == 1 )                                 # print heading
        { print }
    else if ( NF == 6 ) {
        printf ("¥t¥t¥t¥t¥t")                      # print 5 tabs
        z = split($6,j," ")
        printf sum                                 # print sum symbol
        for ( i = z-6+x; i <= z-1+x; ++i )         # shift date item
                printf " " j[i]
        print " " j[z+x] }
    else if ( NF == 4 ) {
        printf ("%s¥t%s¥t%s¥t",$1,$2,$3)           # print fixed data
        z = split($4,k," ")
        printf ("%2d",k[1])                        # print sum
          if ( w == 2 ) {                          # seat/late/absent data
```

```
            for ( i = z-6+x; i <= z-1+x; ++i )
                printf("%2s%-3s"," ",k[i])
            printf("%2s%-3s¥n"," ",k[z+x]) }
          else if ( w > 2 ) {
            for ( i = z-6+x; i <= z+x; ++i )
                if ( k[i] == ab ) { k[i] = " " k[i] }
            for ( i = z-6+x; i <= z-1+x; ++i )
                printf("%s%-4s"," ",k[i])
            printf("%s%-4s¥n"," ",k[z+x]) } }
        close(file)
    print "  From: " j[z-6+x] ", To: " j[z+x]
#   print "  表示始め = " j[z-6+x] ", 表示終り = " j[z+x]
        } }

#8-4 prompt for viewrange scrolling スクロール続行のプロンプト表示
    { if ( z-6+x > 2 )
        { printf("  Viewrange can be changed by Backspace or Ctrl+H ")
        printf("key (q to quit). ") }
#       { printf("  Backspace(またはCtrl+H)キーで表示の範囲を変えられ")
#       printf("ます(q to quit)。") }

#8-5 ending message of viewrange scrolling スクロール終了の表示
    else { print "  Viewrange changing is over."
#   else { print "  表示範囲の変更を終了しました。"
        exit }
        }

 ' _
#-----------------------------------------------------------------
#
#                    Pattern of Registers
#
# Headline(s): Classname, day of the week, period, room number, etc.
```

```
#   (No Tab-key should be used in the line(s)!  Unless there can be
#   written whatever you want, even in plural lines.)
#
# Column Line: 5 Tab-keys and free 2 letters or 1 kanji
#   (For the purpose, spaces cannot be used. These dummies are not
#   displayed.)
# |       |       |       |       tt   : tab scale in image
# |       |       |       |       SS   : e. g.
#                                 Σ    : in reality by kanji
#
# Data Lines: student-code, name, login-name, _0
#   (Each item should be separated only by Tab-keys and the shorter
#   item tail should be filled with space(s) that is(are) written
#   here by underbar '_' image.)
# |       |       |       |       |    : tab scale in image
# 9792110 Park Jong-nam   jpark7__       _0   : in concept
# 9841074 Kim Sun-gin     skim8___       _0   : in concept
# 9912215 Yoshio Hasegawa yhaseg9_       _0   : in concept
# 0031126 Taro Yamada     hyamad01        0   : in reality
# 0121219 Rie Doi         rdoi11          0   : in reality
#   (Login-names can be called by "finger student-code" command in
#   the UNIX sytem through TeraTerm icon.)
# ------------------------------------------------------------------
```

V How to Use "do" and the Results

a. Simple Use of the Program The user gives "do" command to the prompt of the UNIX system, after when all the students logged in once to the UNIX system, by double clicking the "Tera Term" icon on the screen.

b. The 1st Phase in the Sorted Order (Also displayed in the monitors or on the screen to the students in the class room)

```
■ T e s t  F i l e        Rm 6302        on  Sept 4, 2000
                                    Σ  0425 0503 0620 0622 0623 0625 0714
staff   Okamura Yasushi okamura     33  01   01   01   01   01   01   01
9751231 Uemoto Youko    yuemot71    33  17   17-  17-  17-  17-  17-  17-
9741382 Takatani Atuko  atakat7     33  20   20   20   20   20   20   20
  From: 0425, To: 0714
  Sorting can be cancelled by Backspace or Ctrl+H key (q to quit).
```

c. The 2nd Phase in the Register Order (Also displayed in the monitors or on the screen to the students in the class room)

```
■ T e s t  F i l e        Rm 6302        on  Sept 4, 2000
                                    Σ  0425 0503 0620 0622 0623 0625 0714
9751231 Uemoto Youko    yuemot71    33  17   17-  17-  17-  17-  17-  17-
9741382 Takatani Atuko  atakat7     33  20   20   20   20   20   20   20
staff   Okamura Yasushi okamura     33  01   01   01   01   01   01   01
  From: 0425, To: 0714
  Viewrange can be changed by Backspace or Ctrl+H key (q to quit).
```

If the user needs the "hard copy" for the educational purpose, it can be also available here throuth e.g. "秀丸" editor icon, by using copy-and-paste way and printing command. In this case, the font style "MS 明朝" or "MS ゴシック" is advisable, if not, the column lines cannot be printed not in order.

Normally, two results till for 30 students can be printed in a page by using 12 points font, according to the experiences.

d. The Former Records in Horizontal Scrolling Mode (Also displayed to the students)

```
■ T e s t   F i l e          Rm 6302          on  Sept 4, 2000
                                        Σ  0417  0425  0503  0620  0622  0623  0625
9751231 Uemoto Youko     yuemot71       33  17    17    17-   17-   17-   17-   17-
9741382 Takatani Atuko   atakat7        33  20    20    20    20    20    20    20
staff   Okamura Yasushi  okamura        33  01    01    01    01    01    01    01
  From: 0417, To: 0625
  Viewrange can be changed by Backspace or Ctrl+H key (q to quit).
```