DOCUMENT RESUME

ED 297 782 IR 052 503

AUTHOR Koga, James S.
TITLE Break up a File with EDLIN.
PUB DATE [88]
NOTE 6p.
PUB TYPE Guides - General (050)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Software; *Data Processing; Filing; Floppy Disks; *Information Storage; Microcomputers; *Online Searching; *Word Processing

ABSTRACT

Online searchers who download large amounts of data into a single ASCII file often have difficulty fitting that large file into their MS-DOS word processors for post-processing. This paper gives step-by-step instructions for using EDLIN, a text editor for DOS files, to break up a large downloaded file into smaller files. (1 reference) (Author/EW)

BREAK UP A FILE WITH EDLIN

By

James S. Koga

EDLIN.DOC

This is the third and the most complicated method of breaking up a downloaded ASCII file. (The other two methods are: (1) have a word processor capable of importing large files and (2) using a utility program designed to break up large ASCII files.) I don't recommend it if any of the other methods are available. Still, there may be situations where it is the only method you have: all that's required is the EDLIN program, which comes as part of your DOS disk.

ABOUT EDLIN

EDLIN is a line-by-line text editor; its main function is to edit DOS files, as opposed to a word processor's function of producing printed documents. EDLIN is capable of handling large files, but I don't really know its limits. The largest file I have used this EDLIN method to break was only 96K. Wolverton's book states that EDLIN is capable of handling "very large files", but no specifics are given. Most other materials about EDLIN make no mention of file size limits. Since EDLIN works by swapping large chunks of data in and out of its workspace to and from the disk as needed, I suspect that there are no limits other than disk size.

EDLIN also has device specific limitations similar to that of file breaking utilities, only not as severe. EDLIN automatically makes a backup file the same size as the original file, thus consuming a lot of disk space. However, this method allows you to make daughterfiles one at a time, so you only need to have enough free space on the disk for a single daughterfile. You can copy the daughterfiles as they are made to another disk and then erase the daughterfile on the original disk to free up space for the next daughterfile. As a practical limit for 360K floppy users, EDLIN will allow you to break up a motherfile as large as 330K and still allow you to make 30K daughterfiles.

Text is edited in EDLIN on a line-by-line basis; each line is given a number that allows you to identify text to be acted upon. Using EDLIN is complicated by the fact that not all of a large file can fit into the workspace at one time, and that as data is moved in and out of the workspace, it is assigned new line numbers according to its place in the workspace. If you save to disk or delete lines from the workspace, the following lines move up to fill in the gap and take up the line numbers of the ones deleted. If you are used to a word processor, you will find EDLIN strange in that editing can be done to text that does not appear on the screen.

The following example assumes that you have a PC or clone with 2 floppy drives and that you have some familiarity with DOS. Some familiarity with EDLIN wouldn't hurt, either. You will need a blank formatted disk

and a copy of DOS. A calculator will make the arithmetic easier. Make a copy of your large downloaded file onto the blank disk. Keep the original of the large file as a backup. Place the DOS disk in drive A: and the disk with the file in B:.

CHOOSE THE NUMBER OF DAUGHTERFILES

You will first need to look at size of the file determine into how many equal pieces (N) you will need to make for the pieces to be a manageable size. Use a DOS DIR command to find the file size; divide it by your desired file size.

In our example we will be trying to divide a file called Subject.ref. Our DIRECTORY command shows the following:

```
SUBJECT   REF      96463    1-23-88   1:23p
```

I will attempt to get chunks of about 25K for a word processor with a capability of about 30K. I can see that it will take about four chunks (N=4) for me to get my 25K daughterfiles.

FIND THE NUMBER OF LINES IN YOUR WORKSPACE

Start up EDLIN by typing:

```
A:>edlin b:subject.ref
```

We will need to find the line number of the last line in the EDLIN workspace. This may vary with the amount of available RAM. You can find it by doing an EDLIN list command: enter a line number followed by an "l" <Enter>. EDLIN responds by showing a screenfull of lines starting at the line number you entered. When EDLIN responds with only the line number and no text, you have gone past the end of the workspace. For most 640K machines, the last line the workspace will be in the low 1000's. Try to find the end of the workspace by trial and error, trying to close in on the line number of the end of the file.

FIND THE NUMBER OF LINES IN THE MOTHERFILE

You can then move chunks of the file through the EDLIN workspace at a maximum rate to find the end of the file and the number of the last line in the file. If your downloaded file contains your ending logoff messages and such, it is slightly easier to recognize the end of your file. To move chunks of the file the size of your workspace though the workspace, you will use the APPEND (a) command and the WRITE (w) command. In our example I found the workspace to be 1030 lines long. I alternately wrote chunks 1030 lines long to the disk, freeing up the workspace, and appended the next chunks or 1030 lines until the end of file message appeared.

```
*1030w   (writes/saves first 1030 lines to disk)
*1030a   (brings the next 1030 lines into workspace)
*1030w   (writes/saves those 1030 lines to disk)
```

```
*1030a   (brings remainder of file into workspace)
 End of Input File
```

Keep track of the number of times you do each command. Again, use the list command, by trial and error, to find the end of the file. You can then calculate the total number of lines in the file by totaling up the number of lines you have written to disk, and the number of lines now in the workspace.

In our example, we found the end of the file at line 85. thus, our calculations for the total number of lines in the file are as follows:

```
        1030  (the first write command)
        1030  (the second write command)
          85  (the number of lines left in file)
        ----
Total:  2145  lines in the file
```

Exit EDLIN with the q command, which is invoked by typing the letter "q" and then enter. It allows you to exit back to DOS without creating a new file. You are given a choice for creating a file. Respond no. You will later use the "e" command, which will exit you back to DOS and save work you did with EDLIN.

FIND THE NUMBER OF LINES IN EACH DAUGHTERFILE

Divide the number of the last line in the file by the number of chunks you want to end up with. Round up to the neared whole number. This will be the desired number of lines (Y) in each daughterfile.

In our example:

2145 divided by 4 = 536.25

so our number of lines in each daughterfile (Y) = 537. This method works best if Y (the number of lines in each subfile) is less than the number of lines in the workspace. That is, if Y is larger than 1030, adjust N up to where Y is less than 1030.

MAKE THE FIRST DAUGHTERFILE

Start up EDLIN again.

We will first make the first subfile. To do this, we can write the first chunk to disk, effectively saving it. We then alternatively fill and delete the workspace in chunks of 537 lines until the end of input file message is given. We then delete the remainder.

```
*537w     (writes/saves first 537 lines to disk)
*537a     (fills workspace with next 537 lines)
*1,537d   (deletes top 537 lines)
*537a     (fills workspace with next 537 lines)
*1,537d   (deletes top 537 lines)
```

```
*537a     (fills workspace with remaining lines)
 end of input file  (that is all of the file)
*1,537d   (deletes remaining lines)
*e
```

Now at DOS, we can do a DIRECTORY command and check our work:

```
Subject  bak  96463  1-24-88   1:24p
Subject  ref  22711  1-24-88   1:39p
```

Our original large file is now called Suject.bak because EDLIN always saves a copy of the original file with the extension bak, and the first daughterfile we have created is now called subject.ref and should appear considerably smaller than subject.bak. Use the DOS RENAME command to RENAME the files; RENAME subject.ref to something like subject1.ref and subject.bak should then be restored to the name subject.ref.

```
A:>rename b:subject.ref subject1.ref
A:>rename b:subject.bak subject.ref
```

MAKE THE LAST DAUGHTERFILE

Start up EDLIN again, again using subject.ref as is file to work on:

```
A:>edlin b:subject.ref
```

We can now go to making the last daughterfile in our set. In this case, this is a lot simpler. All we need to do is delete and append chunks of lines until all that is left is the last chunk.

Here in our example, we have deleted 3 chunks of 537 lines, which would leave us with the forth chunk intact.

```
*1,537d  (deletes the first chunk)
*537a
*1,537d  (deletes the second chunk)
*537a
*1,537d  (deletes the third chunk)
*e
```

Again, at the DOS level, do a DIR command and view the results. RENAME as with above, only RENAME subject.ref to subject4.ref.

MAKE THE MIDDLE DAUGHTERFILES

To make the daughterfiles in the middle, a different method is called for. The basic principle is to delete up to the chunk that is desired and then, since the desired chunk is in the top of the workspace, delete and append all that is past it. To make the third chunk in our example, the following is done:

```
*1,537d   (deletes the first chunk)
*537a
```

```
*1,537d   (deletes the second chunk and moves third to
*537a         top of workspace)
*537w     (writes the desired third chunk to disk and
*537a         moves it out of the workspace)
 end of input file
*1,537d   (deletes the forth and remaining chunk)
*e
```

Do a DIR command again and perform the appropriate RENAMEs.

Making the second daughterfile is very similar except that we delete only one chunk of 537 lines to get the desired chunk from the workspace; we then write that chunk to disk and then append and delete to the end of the file.

You can see that making the middle chunks for a larger number of chunks can be done by extensions of this method.

FURTHER NOTES ON EDLIN

This method can be souped up with copying EDLIN.COM onto the floppy so you don't have to deal with A:'s and B:'s; batch files can speed up the RENAME process. I suppose the device size limitations can be overcome by well timed disk swapping as EDLIN writes to disk. A few times doing the EDLIN method will make you want to use a utility program; I don't think you should spend much time trying to improve it.

REFERENCE

Wolverton, Van. *Running MS-DOS*. Redmond, WA.: Microsoft Press, 1985. Chapters 11 and 12 make a good tutorial for EDLIN, but most of it deals with changing text rather than dealing with deleting parts of large files.

James S. Koga is the Coordinator of Online Reference Service at the University Library at the California State Polytechnic University in Pomona. Communications to the author can be sent to: University Library, California State Polytechnic University, Pomona, P.O. Box 3088, Pomona, CA 91768. 714/ 869-3098 Dialmail: 14415.