

FAST EDITING AND PROGRAM DEVELOPMENT

USING A FULL SCREEN EDITOR

J. VAN DAMME

J. VAN DAMME
SYDES
A. GOSSETLAAN 30A
B 1720 GROOT-BIJGAARDEN
BELGIUM

FAST EDITING
AND
PROGRAM DEVELOPMENT
USING A
FULL SCREEN EDITOR

1. INTRODUCTION.

In most computer systems, text handling is one of the main tasks. Program development, which is a form of text handling, can take a considerable amount of CPU resources. The choice of a good text editor can result in a large increase of productivity.

We will now look at the advantages and disadvantages of several kinds of text editors.

2. THE SEQUENTIAL EDITOR.

The first text editors considered three sequential files. The first file contains the old text to be modified; a second file contains commands to allow the deletion, replacement and addition of full lines. The result is a new sequential file, containing the corrected text.

An example of this editor type is QUERY's 'ALTER' command to modify a procedure.

Typical to these editors is that the line numbers, given in the commands, must be strictly ascending.

Only few of the sequential editors have string handling capabilities.

3. THE INTERACTIF EDITOR.

A major enhancement to the sequential editor was the introduction of interactif capabilities. In general these include

- the possibility to jump back and forth in the text file.
- string handling capabilities (e.g. FIND "string")

Interactif editors act on one single line at a time. So, if the user wants to see the context of the changes he is introducing, parts of the text have to be listed after entering some edit commands.

Most interactif editors work on a copy of the original text (a work file). The operation of copying the original text into the work file and vice versa, takes a considerable amount of time and system resources. On the other hand, working directly on the original text, is an hazardous task to undertake since a mere typing error could destroy large parts of the text.

Regularly taking a copy of the text solves the problem. An alternate solution is to copy the file, at the start of each edit run, and save the work file as the new text at the end. This saves the time of a 'KEEP' operation.

4. THE FULL PAGE EDITOR.

At the time, when more intelligent terminals became available a large part of the editing task could be performed by the terminal instead of the computer.

The editor sends a full page of text to the the screen. This text is edited using the edit features of the terminal (e.g. 'INSERT LINE', 'DELETE CHAR').

After editing of the page is finished, the terminal sends the full page back to the computer and this one sends a new page to the terminal.

The advantages of this method are obviously:

- a good oversight of the context of the changes performed.
- a very natural way of editing (it looks like editing on a sheet of paper).
- very powerfull if the terminal used has good editing capabilities.

The disadvantages are:

- Sending a page back and forth takes a considerable amount of time if the terminal speed is low.
e.g. eight seconds to send a 24 lines by 80 characters page at 2400 baud.
This transmission is performed twice, even if only a single character is to be modified.
- A sophisticated and hence expensive terminal is needed.
- The amount of text that can be added to a page is dependant on the amount of local store in the terminal.

5. THE FULL SCREEN EDITOR.

To overcome the disadvantages of the full page editor, the editor itself can emulate the sophisticated edit features of an expensive terminal. The user then edits the whole file at a time instead of only one page. The editor accepts one edit command (this can be a single key stroke) and performs the edit on the text file and on the screen.

A typical example of this editor type is the editor available on the HP 300 system.

The advantages of such a system are:

- all the advantages of a full page editor.
- considerable savings of time.
- The only special features needed on the terminal are
 - . cursor positioning
 - . character and line insertion and deletion

All other features can be emulated by the editor.

This allows to implement a full screen editor on a low cost terminal.

6. ADDITIONAL FEATURES.

Many additional features are added to text editors. Some of them are:

- The capability of compiling directly from the work file.
- Disc space saving by record compression.
- Handling of MACRO's.
- Automatic generation of program code.
- Word processing capabilities.
- Loop constructs.
- Conditional editing.

FULL SCREEN EDITING

7. FSEDIT.

FSEDIT is a full screen editor for the HP3000 computer systems.

It will run on any terminal from the HP264x and HP262x families, including the low cost HP2621.

Its main features are:

- full screen editing as described above.
- single key-stroke commands.
- direct compile, prepare and run from the work file.
- word processing capabilities.
- COBOL source generation.
- macro handling.
- powerfull command set.

These and other features make FSEDIT an easy to use tool, that allows a high increase in productivity of programmers and other users.

FSEDIT is developed by

SYDES N.V.

TEL 02/4662813

A. GOSSETLAAN 30A

TELEX 63435

B 1720 GROOT BIJGAARDEN

BELGIUM.

Demonstrations of FSEDIT are given at the SYDES N.V. booth.