Cahiers GUTenberg

TEX INTEGRATED SHELL FOR IBM PC

Basil Malyshev, Alexander Samarin

Cahiers GUTenberg, nº 10-11 (1991), p. 45-55.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1991___10-11_45_0>

© Association GUTenberg, 1991, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*
(http://cahiers.gutenberg.eu.org/),
implique l'accord avec les conditions générales
d'utilisation (http://cahiers.gutenberg.eu.org/legal.html).
Toute utilisation commerciale ou impression systématique
est constitutive d'une infraction pénale. Toute copie ou impression
de ce fichier doit contenir la présente mention de copyright.

TeX Integrated Shell for IBM PC

Basil MALYSHEV and Alexander SAMARIN

Institute for High Energy Physics, 142284, Protvino, USSR
malyshev@m9.ihep.su, samarin@m9.ihep.su

Abstract. This article presents the TeX Integrated Shell (TIS) - special environment for TeX at IBM PC to conceals some problems from an ordinary user. TIS contains the screen interface for different actions during TeXing. It can be configurated to satisfy user's requirements and hardware & software conditions. It downloads only the files to be used, in particular, pixel font files which are required for a given .dvi file.

Résumé. *Nous présentons* TIS (TeX Integrated Shell), *un environnement TeX pour IBM-PC.* TIS *permet de masquer certains aspects du DOS à l'utilisateur.* TIS *intègre une interface écran qui gère les relations TeX-utilisateur. Cet environnement est paramétrable et s'adapte à la configuration matériel/logiciel utilisée.* TIS *n'autorise que le chargement des fichiers utilisés, par exemple les fontes nécessaires à l'impression d'un fichier* .dvi.

Key words: Special environment, integrated shell, TeX, emTeX, IBM PC, MS-DOS.

Introduction

The software for TeX at IBM PC compatible computers is permanently growing. New drivers, utilities and etc. are appeared in public domain archives. The fine collection of TeX at IBM PC by Eberhard Mattes contains a lot of programs, drivers, utilities with many parameters and capabilities.

Such abundance of programs, fonts and features makes difficult the usage of TeX. The special environment for TeX might conceal some problems from an ordinary user. We have developed TeX Integrated Shell (TIS) which offers the following:

- A screen interface for different functions during TeXing. Depending on extension of a file name and current configuration some functions are available. For extension .tex one can run TeX or LaTeX, the text editor or spell checker. For extension .dvi one can preview a file or print it on a specified device.

- TIS can be configurated for the hardware & software conditions and user's requirements. One can change a printer(s), select TEX formats, specify how a printer is connected to computer, what fonts will be used for printing, etc.
- Downloading from different resources (distribution diskettes, file server or a FTP site) files to be used, in particular, pixel font files which are required for printing a given `.dvi` file.

The standard command interpreters are not powerful enough to provide the concerted working of screen interface, TEX collection and operating system An interpreter RUN with LISP-like language has been developed. Most of system-organizing procedures and services are realized via RUN. It handles with files and environment variables, runs programs, organizes the simple windows and menus, supports mouse and contains many useful arithmetic, logic and string operations. Full list of its commands is presented in Appendix A.

In the text below the references on the keys for IBM PC keyboard are given in the following notation: `F1` means the pressed key `F1`, `Alt`+`F1` means the pressed keys `Alt` and `F1`.

1. The screen interface

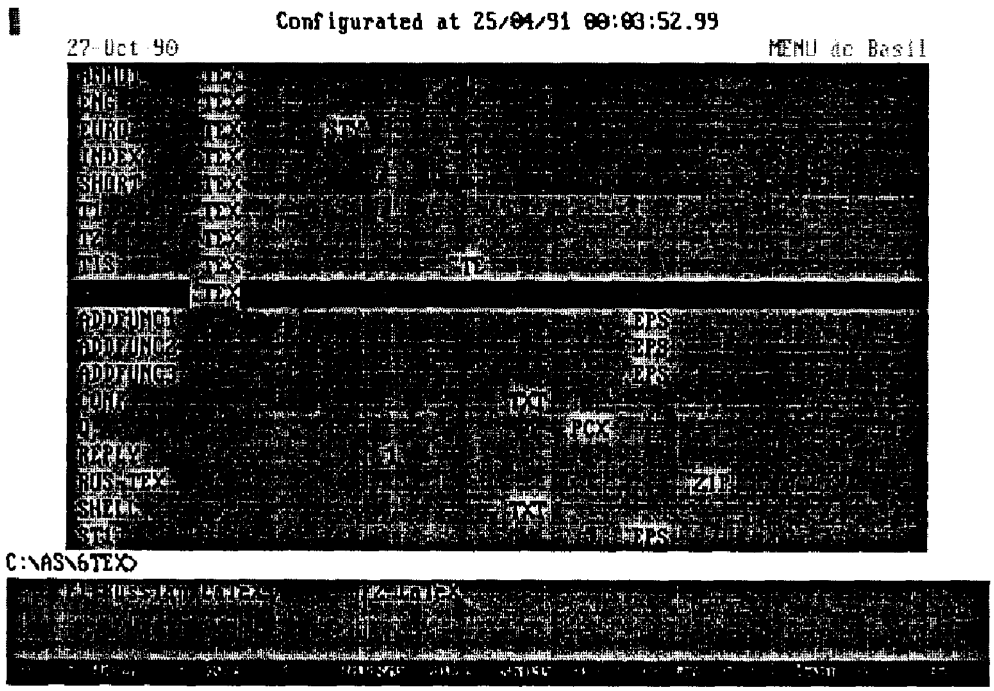
The screen interface displays a picture (see fig. 1), which contains the four field. The top field shows the status information, e.g. the date of the last reconfiguration[1].

The next field is the *file window* for current and working directories[2]. The files are arranged in the following way: files with the same name are in one line, the name is in the left most column and further the extensions are. If all files with the same extension don't fit to the field, then a line will extent to left or/and right and coloured markers will show "overflows".

The current file is pointed by strip with other colour (covered the whole line) and by rectangular with other colour (covered the extension). On the fig. 1 the current file is `x.tex`. If a file is in the current directory then extension followed by character ".", if a file is in the working directory then – ":", if a file is in the both directories then – "*".

[1] The configuration of TIS is described in section 4.

[2] The directories used in TIS are described in section 3.



```
Configurated at 25/04/91 00:03:52.99
27 Oct 90                                          MENU de Basil
C:\AS\6TEX>
```

Figure 1: An example of the screen interface.

The next field consists from one line and starts from MS-DOS prompt. This is the *command line*, which accumulates the ordinary characters typed on the keyboard. They are considered as program name or command for MS-DOS.

The next field is three-lines *functional window* which contains names of functions executed by keys `F1` – `F9`. The contents of this field depends on the extension of the current file and the configuration. There are about twenty predefined functions to running TEX with different formats, viewing `.dvi` files, printing `.dvi` files on many printers, calling utilities and etc. In addition a user can supply new functions by special procedure (see section 2).

The bottom line contains some useful keys.

1.1. Example of working with screen interface

Let us see how to do some popular actions like editing a file, running LATEX, viewing and printing the result.

Inside screen interface one should select (make it current) a file being handling. Note, that all files with extension `.tex` will be at the beginning of the file window. To edit the current file one should press $\boxed{\texttt{Alt}}$+$\boxed{\texttt{E}}$. The editor which is defined in configuration will started.

On the fig. 1 for a file with extension `.tex` two functions are available (they are presented in functional window): by pressing key $\boxed{\texttt{F1}}$ the file will be processed by Russian LaTeX, $\boxed{\texttt{F2}}$ – by LaTeX.

Considered that key $\boxed{\texttt{F2}}$ has been pressed. The screen interface will disappear and one will see *user's screen* which is hidden by screen interface. This user's screen contains all messages generated by programs running under screen interface, e.g. TeX. After LaTeXing the screen interface will appear again and it will display the new contents of the current and working directories. The current file will be `x.dvi` from the working directory.

For example, two functions are available for `.dvi` files: "Previewing" by pressing $\boxed{\texttt{F1}}$ and "Printing LaserJet_LJ2" by pressing $\boxed{\texttt{F2}}$. One can call these functions very simply.

1.2. Usefull keys

There are some useful keys:

$\boxed{\texttt{Alt}}$+$\boxed{\texttt{E}}$ edit the current file (if an editor is specified in configuration);

$\boxed{\texttt{Alt}}$+$\boxed{\texttt{V}}$ list the current file (if a proper program is specified in configuration);

$\boxed{\texttt{Ins}}$ switch between screen interface and user's screen;

$\boxed{\texttt{Alt}}$+$\boxed{\texttt{C}}$ configurate TIS (see section 4).

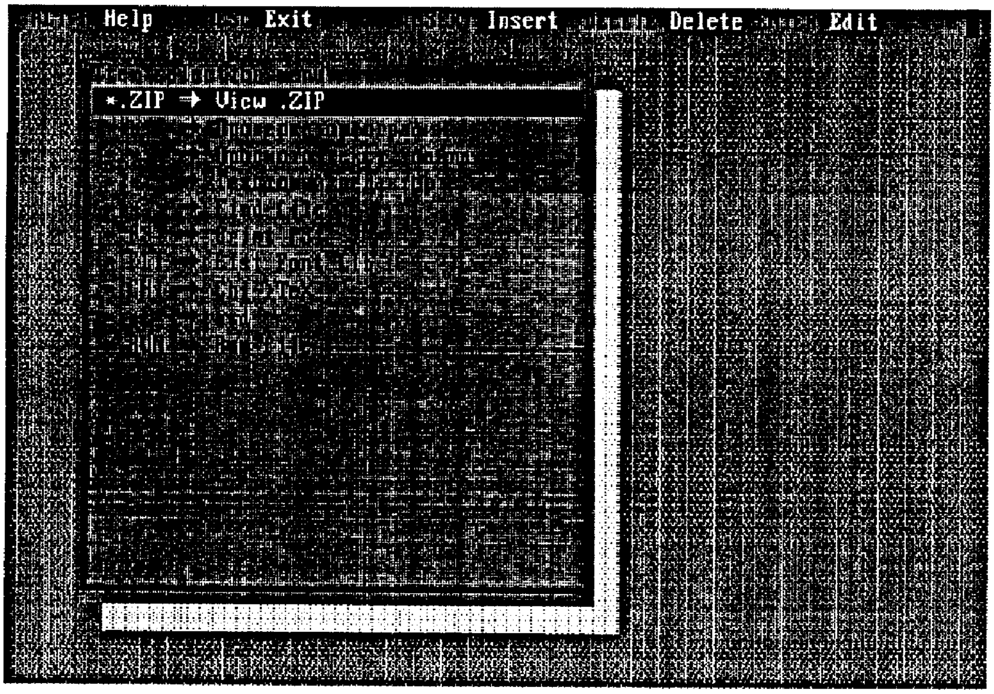
$\boxed{\texttt{Alt}}$+$\boxed{\texttt{A}}$ define the user-supplied functions;

$\boxed{\texttt{Alt}}$+$\boxed{\texttt{U}}$ execute some user-supplied functions.

The "mouse" is available in file and functional windows. Via mouse, one can select the current file and a function to be executed.

2. Adding new functions by user

New functions for screen interface can be simple added by user via service procedure written by RUN. We considered that in general these functions



*.ZIP → View .ZIP

Figure 2: Procedure for adding new functions

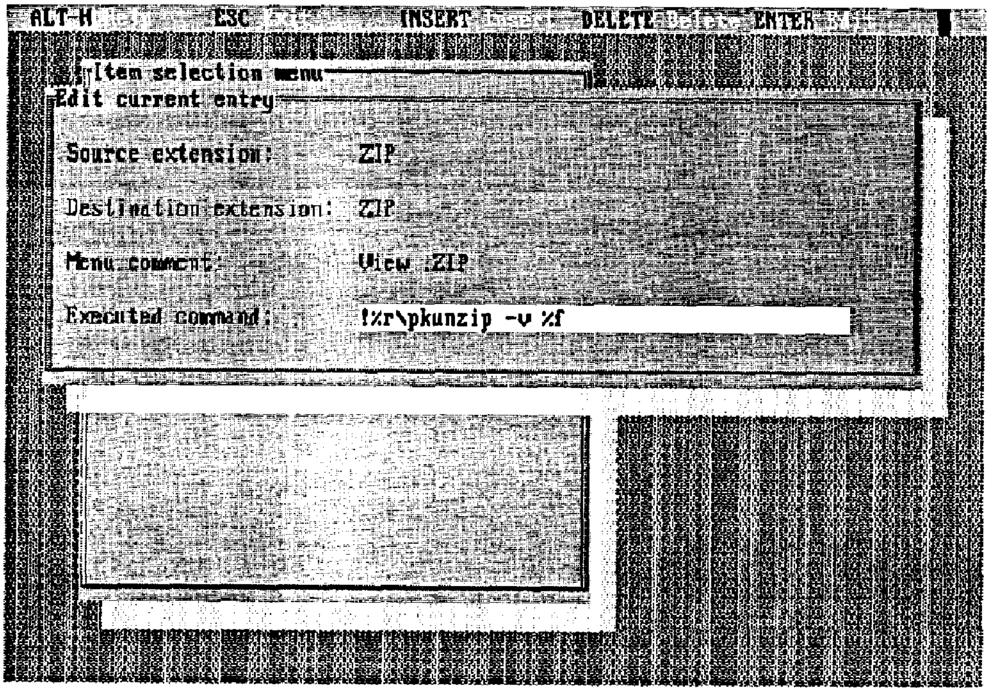
convert a file of one type into a file of another type. So the description of a function consists of four items:

- an extension of the source file;
- an extension of the destination file;
- a short comment (10-15 characters) for functional window;
- a command to be executed.

Being started the procedure displays (see fig. 2) the list already defined functions – the source extension and the comment.

One can add new function, edit or delete one of them. The editing and adding of a function is performing by filling the "blank" which consists of four fields. For example the definition of the popular archive program is presented on fig. 3.

In a command one can use the following two letters string to substitute proper information:



Source extension: ZIP

Destination extension: ZIP

Menu comment: View ZIP

Executed command: !%r\pkunzip -v %f

Figure 3: Definition of the function for archive program.

`%n` the name of the current file;

`%e` the extension of the current file;

`%f` full name of the current file;

`%r` TIS directory (e.g. `C:\TEX\TIS`);

`%w` name of the working directory.

The following characters at the first position of a command have the special meaning:

`@` program for RUN will be executed;

`!` ordinary program will be executed and MS-DOS command interpreter is not needed;

`$` the working directory is used instead of current one.

In the list there are some functions which are available for all files (extension of the source file is "*"). These permanent functions do not appear into functional window – they are activated by keys $\boxed{\texttt{Alt}}+\boxed{\texttt{F1}}$ – $\boxed{\texttt{Alt}}+\boxed{\texttt{F9}}$. Spell checker is an example of permanent function – one would check any text file. The list of permanent function is displayed by $\boxed{\texttt{Alt}}+\boxed{\texttt{U}}$.

3. File structure considerations

TIS works with following directories:

current is a directory where an user is working;

local is a root directory for all system (specified during installation);

working is a directory for temporal files (it defined in configuration);

network is a public directory at remote disk (if any) with the structure as a local directory (e.g. network directory at Novell file server);

fast is a directory at RAM drive (if any).

The working, local, network and fast directories allow to rearrangement files to speed up the working or reducing the usage of hard disk if the different medias are available at given computer. Having large RAM drive one can copy into working directory format files, pixel font files and etc. If a network directory is available then only a small part of system might be at a hard disk. Fast directory is useful if a RAM drive is small and can't hold all working information.

The structure of the local and network directories is not fixed. One can change the disposal of the executable, `.tfm` and `.fmt` files, TeX and BibTeX input files and the pixel font files. At present the emTeX collection with another structure of directories is used. For unusual arrangement of the files the small corrections might be required.

4. Configuration

One can configurate TIS by following parameters:

1. *language* – at present Russian and English languages are supported in TIS. Including another languages is taken into account.
2. *directories* – they are presented in section 3.

3. *screen interface customization* – one can setup colours of the different fields, specifies the programs for editing and listing files and changing the current directory and etc.
4. *TEX's formats* – eight formats are supplied, but any `.fmt` file in proper directory is considered as format for TEX and such format would be available at screen interface.
5. *graphical card* – sometimes previewing program `DVISCR` from emTEX requires the implicit setting of video adapter.
6. *printers* – all drivers from emTEX are available. For each driver one must specify:
   - resolution of fonts to be used;
   - magnification of the printing;
   - printer's port;
   - how to print: directly, through file or does not send a file to device;
   - additional options.
7. *source for downloading* – TIS can copy required files during working from following sources:
   - distribution diskettes;
   - network directory (if a network is slow);
   - host accessed via TCP/IP.
8. `utilities` – BibTEX, TEXCAD, Makeindx, different convertor and etc. might be turned on and proper function will appear in the functional window.

In addition one can change the style of the screen interface – it may be reduced up to two lines or realized as a small menu.

5. Downloading of files

To download files from distribution diskettes, the lasts are accompanied with maps. A map contains the contents of the `.zip` files stored at the set of diskettes. At beginning of installation the maps for software and pixel files are copied at hard disk. Then any file is retrieved by straightaway procedure. One can created its own maps and use this facility.

In the case of downloading from network directory or via FTP we consider than the source directory have the same file structure as the local directory.

A. List of RUN's commands

• *General functions*

```
(local ...)                (block ...)                (index...)
(continue level)           (void ...)                 (do run-file)
(input run-file)           (endinput)                 (# N)
(not cond)                 (and ...)                  (or ...)
(case ...                  (while cond body)          (break level)
(letqq function-name body)
(if cond on-true [on-false])
(function name minarg [maxarg])
```

• *String manipulation functions*

```
(cat string>...)           (len string)               (cut string Char)
(nes string1 string2)      (ges string1 string2)      (gts string1 string2)
(les string1 string2)      (lts string1 string2)      (eqs string1 string2 ...)
(find string tag)          (upper string)             (lower string)
(align_left string Width)
(align_center string Width)
(align_right string Width)
(extr string offset length)
(subst string tag subst)
(token string delimiter N)
(elist string delimiter modificator list)
```

• *Numberical functions*

```
(ne exp1 exp2)             (lt exp1 exp2)             (gt exp1 exp2)
(le exp1 exp2)             (ge exp1 exp2)             (eq exp1 exp2)
(sum exp1...)              (div exp1 exp2)            (sub exp1 exp2)
(mul exp1...)              (neg exp1)                 (inc var)
(dec var)                  (min ...)                  (max ...)
```

• *File manipulations*

(fsize file)	(cd [directory])	(search path file)
(del file ...)	(fdate file)	(ftime file)
(fparse format file)	(ren old_name new_name)	(exist file pattern list)
(for pattern function)	(fordir pattern function)	(fmode file [attributes])
(copy dst_file src_file ...)		

• *Local variables*

(@ name)	(@@ name)	(quote name)
(let name [expansion])	(letq name [expansion])	

• *Environment*

(reset_env)	(% name)	(%% name)
(set name expansion)	(setq name expansion)	(gset name [expansion])
(gsetq name [expansion])		

• *Input/Output functions*

(binary ...)	(openread file)	(openwrite file)
(read fd [mode])	(eof fd)	(close fd)
(tell fd)	(seek fd ptr)	(openappend file)
(openinput file root)	(write fd str [mode])	

• *Display functions*

(type ...)	(screen ...)	(clearscreen [pattern])
(screenew ...)	(colors [colors])	(clearwindow [pattern])
(getkey)	(lastkey)	(cursor [cursor_position]
(keycolors [colors])	(clearline [pattern])	(menucolors [colors])
(framecolors [colors])	(link [link point])	(textcolors [colors])
(window [window_coord])		
(setkey Key_Name Comment)		
(edline window help start_line)		
(vcontrol window help (item ...) ...)		
(vicontrol window help list Auxiluary)		

• *Miscelaneous*

(arg num|id)	(dos ...)	(err)
(return string)	(date)	(time)
(exit [err-code])	(service [new-service])	(run program arg-list)
(exec program arg-list)	(future MS-DOS command)	(call run-file arg-list)

• *Hacks*

```
(int intno ...)           (mem address ...)          (port address ...)
```

An example – "Selection of one item from the list". At first the function `obj` is defined. It has two parameters: item identifier and name of the item. Really it calls command `(item )` to output the text into menu and to execute the command if an item is selected.

```
(letqq (function obj 3)
   (item
      ;      Output the text  -- name followed by "*"
      ;      if this item is selected.
      (cat (if (eqs curr_obj (# 1)) " * " "   ") (# 2) )
      ;      Execute the command
      (if (eqs curr_obj (# 1))       ; Is this selected item?
         (break)                     ; Yes. Second selection
                                     ; terminate the loop.
         (void                       ; No.
            (letq curr_obj (# 1))    ; Make item ''selected''
            (continue)               ; and continue the loop.
         )
      )
   )
)
(letq curr_obj "ID?")               ; Initial setting

(vcontrol (window ,,50,,SUU) "code" ; Making menu function
   (obj "ID1" "Name_1")             ; The first item
   (obj "ID2" "Name_2")
   ..................
   (obj "IDN" "Name_N")
)
(type "Selected item " curr_obj)    ; Type the result
```