Memory Sizes in Various Implementations of TEX

Barbara Beeton
American Mathematical Society

Probably the most discouraging error message to appear in the middle of a TEX job is "`TeX capacity exceeded, sorry.`" No one is sorrier, of course, than the victim. However, knowing one's requirements and TEX's capacity makes it possible to plan ahead, or at least to be prepared for the eventuality.

The tables on the following pages give the memory capacity compiled into most of the implementations of TEX now in distribution. The values listed are those assigned in §§11–12 of `TeX.WEB`. TEX memory capacity is discussed in some detail on page 300 of *The TEXbook*, and the names given there for segments of memory that can be overloaded (and thus can occur in error messages) are also shown in the tables.

A historical digression is in order. In the original implementation of TEX82, the main dynamic memory was split into two segments, "box memory" (boxes, glue, and paragraph breakpoints) and "macro memory" (token lists and characters). (A diagram of TEX82's memory structure for tables and work areas, but not including the main dynamic memory, appeared in *TUGBOAT* 3, no. 2: 13.) This division proved too limiting — jobs which required a relatively large amount of box memory (such as the output device charts published regularly in *TUGBOAT*) would overload that segment while there was still plenty of untouched macro memory, and vice versa. A major redesign, implemented in version 1.3, combined these two segments; now, capacity is not exceeded until all available "main memory" is exhausted. If you are plagued by `capacity exceeded`, and the message specifies `box memory` or `macro memory`, you are still using a version of TEX older than 1.3, and should (in addition to taking whatever immediate remedies are necessary to complete your current job) do everything you can to get a more up-to-date version installed.

To determine your actual "overhead" requirements, follow the suggestions on page 300 of *The TEXbook*: set `\tracingstats` to a positive value, input the header files you will be using, and say `\bye`; TEX will report the memory required in each category. At some sites, this feature has been turned off to maximize processing speed; if you find this to be true, consult your local wizard.

Examination of the tables will show a wide variation in the memory allotments. The values assigned in the "generic" implementation, `TeX.WEB`, are suitable for general use with `PLAIN.TeX` and can be accommodated by nearly all computer architectures and suitable Pascal compilers. There are many reasons for enlarging the memory allocations; following are some of them.

Large macro packages (e.g. LATEX or AMS-TEX) require significant amounts of memory just to be loaded in, before any user data is input. This has been cited by Kellerman & Smith as the reason for their large string and string pool sizes, and at the AMS is the reason for the pool size increase.

Nearly every implementor seems to agree that increasing the main memory is a good idea. However, some compilers are unable to pack a value greater than 32K into a half-word; TEX permits the main memory starting address to be negative, and the VAX UNIX and DG MV implementations take this option.

At the AMS, some TEX input is program-generated, and input lines can be much longer than those typical of user-keyboarded input.

Bilingual TEX accommodates hyphenation patterns for both English and French, and thus requires additional pattern memory (but not much!). Alternate fonts may also be desirable in a multilingual environment (see the article by Jacques Désarménien, "How to run TEX in a French environment," *TUGBOAT* 5, no. 2: 91–102).

Figure 1.

Memory Sizes.

[1] Values in brackets are what is used at the AMS, not in the distributed version.

[2] Value in brackets = actual used.

[3] Values in brackets are for IniTEX.

[4] Adjusted at run time for available memory; average sizes in brackets.

[5] Dynamically changed depending on system size and user demands; values are typical maximum values using DOS 2.1 on a 640K XT.

	TeX.WEB §11–12	DEC 20 (Stanford, AMS)	VAX UNIX (U of Wash)	VAX/VMS (Stanford)
Version Distributed	2.0	2.0 [1.5][1]	2.0	1.5
Form of Distribution		web, exec	web, exec	web, exec
Compile-Time Parameters				
mem_max main memory	30000	58000	30000	58000
mem_min	0	0	-30000	0
buf_size buffer	500	500 [1500][1]	500	500
error_line	72	79	79	79
half_error_line	42	50	50	50
max_print_line	79	79	79	79
stack_size stack	200	200	200	200
max_in_open text input levels	6	6	15	6
font_max	75	100	100	100
font_mem_size font memory	20000	25000	25000	25000
param_size parameter stack	60	60	60	60
nest_size semantic nest	40	40	40	40
max_strings number of strings	3000	4400	4400	4400
string_vacancies	8000	15000	15000	15000
pool_size pool size	32000	40000 [50000][1]	45000	40000
save_size save size	600	600	600	600
trie_size pattern memory	8000	8000	8000	8000
dvi_buf_size	800	800	800	1024
file_name_size	40	69	1024	69
Parameters Requiring Rerun of `INITEX`				
mem_bot	0	0	-30000	0
mem_top main memory	30000	58000	30000	58000
font_base	0	0	0	0
hash_size hash size	2100	2500	3000	2500
hash_prime	1777	2113	2551	2113
hyph_size exception dictionary	307	307	307	307

VAX/VMS (K&S)	VAX/VMS TeX (INRS)	CDC Cyber (U of Wash)	IBM VM/CMS	Prime	Sperry 1100	Apollo	Data General MV	HP 3000	IBM PC (MicroTeX)	IBM PC (PC TeX)
1.5	1.1	1.5	1.1	1.3	1.3	1.5	1.5	1.5	1.5	1.5
web, exec	web, exec	web	web, exec	web, exec	web, exec	web, exec	web, compiled	exec only	exec only	exec only
65000	65000	32000	60000	58000	58000	60000	30000	65534[4] [58000]	64000[5]	65534[4] [58000]
0	0	0		0	0	0	–30000	0	0	0
500	500	500	1024	500	500	500	500	500	1000	500
79	79	72	79	72	72	79	79	72	79	72
50	50	42	50	42	42	50	51	42	50	42
79	79	79	79	79	79	79	79	79	79	79
200	200	50	200	200	200	200	200	200	200	200
6	6	6	6	6	6	6	6	6	16	6
100	200	75	100	100	75	100	99	75	100	75
25000	50000	18000	25000	25000	20000	25000	25000	30000[4] [26000]	29000[5]	30000[4] [26000]
60	60	60	60	60	60	60	60	60	120	60
40	40	40	40	40	40	40	40	40	40	40
10000	4400	3000	5000	4400	3000	5000	4400	4000	6000[5]	4000
20000	20000	8000	13000	15000	8000[3] [4000]	20000	16000	13000		13000
65000	60000	32000	40000	40000	32000[3] [28000]	50000	40000	40000	60000[5]	40000
600	600	600	600	600	600	600	600	600	600	600
8000	16000[2] [7316]	7000	8000	8000	8000	8000	8000	8000	6000	8000
800	1024	80	2048	800	800	1024	1024	512	1024	512
69	76	26	40	40	88	256	256	40		40
0	n/a	500		0	0	0	–30000	0	0	0
65000	n/a	25000		58000	58000	60000	30000	65534	64000[5]	65534
0	0	0	0	0	0	0	0	0	0	0
5000	2500	2100	2100	2500	2100	3000	2500	2500	2500	2500
4253	2129	1777	1777	2113	1777	2551	2113	2129	2113	2129
307	307	307	307	307	307	307	307	307		307

Sources of Information

`TeX.WEB` Stanford distribution

DEC 20 Stanford distribution

VAX UNIX Pierre MacKay,
University of Washington

VAX/VMS David Kellerman,
Kellerman & Smith

VAX/VMS Stanford distribution

VAX/VMS TeX Michael Ferguson, INRS

CDC Cyber Jim Fox,
University of Washington

IBM VM/CMS Alan Spragens,
Stanford Linear Accelerator Center

Prime John Crawford,
Ohio State University

Sperry 1100 Bill Kelly,
University of Wisconsin, Madison

Apollo Bill Gropp, Yale University

DG MV Bart Childs,
Texas A&M University

HP 3000 and IBM PC (PC TeX) Lance Carnes,
Personal TeX

IBM PC (MicroTeX) David Fuchs,
Stanford University