bcc	title	prefix/class-number.revision
	SPECIFICATION OF PROGRAM IMAGE FILES	PIF/S-21

checked	authors	approval date	revision date
Carl Freeman	Larry L Barnes Larry Barnes L. Peter Deutsch L. Peter Deutsch	10/21/69	
checked Butler W Lampson		classification Specification	
approved Mel		distribution Company Private	pages 7

ABSTRACT and CONTENTS

This document specifies the format of program images (files containing complete programs) in the Model 1 software system. Such files may be run as subprocesses, loaded into SPL, or written on tape for use in system startup. The ATTACH UCALL, for making a subprocess out of a program image, is described in general terms.

A program image consists of a collection of data pages which form a program, a map describing how to arrange them in the appropriate ring of the M1 address space, and enough additional information to:

(1) turn them into a subprocess;

(2) write them out as part of a system initialization tape;

(3) load them into SPL, including the source program they were compiled from.

The information required for (1) and (2) is actually part of the data; for (3), SPL appends information of its own beyond the actual data pages. The overall layout of program images is shown in figure 1; the three uses of such files is now discussed in detail.

The first forty-six words (see Figure 2) of a program image contain the information necessary to set up a complete sub-process entry. The format code is included to allow us to revise the header while maintaining the use of old files. If CUNF is non-zero, the MIB owner number will be copied to PAK1 of the sub-process. EP and EG are copied to the SPT entry. EP should point to the sub-process entry point array descriptor. The Utility Name contains either the Basic File System Name for the utility under which this file runs or else the main:name should be blank to mean use the current utility.

The map has 64 12-bit bytes. Each byte is interpreted as follows:

Code (CD)	Action
0	Put Ø in this map byte.
1	Put the file page (specified by the index) into PMT read-only.
2	Create a PMT page.
3	Copy the file page to a new PMT page.

In cases 1 to 3 the PMT byte number is placed in the corresponding map byte.

Specification for ATTACH

Declaration:

ATTACH (SPTX, ARRAY FN), FRETURN;

Success Return:

RETURN SPTX;

Failure Returns:

If the file cannot be opened, or if the sub-process cannot be created, or if PMT fills up.

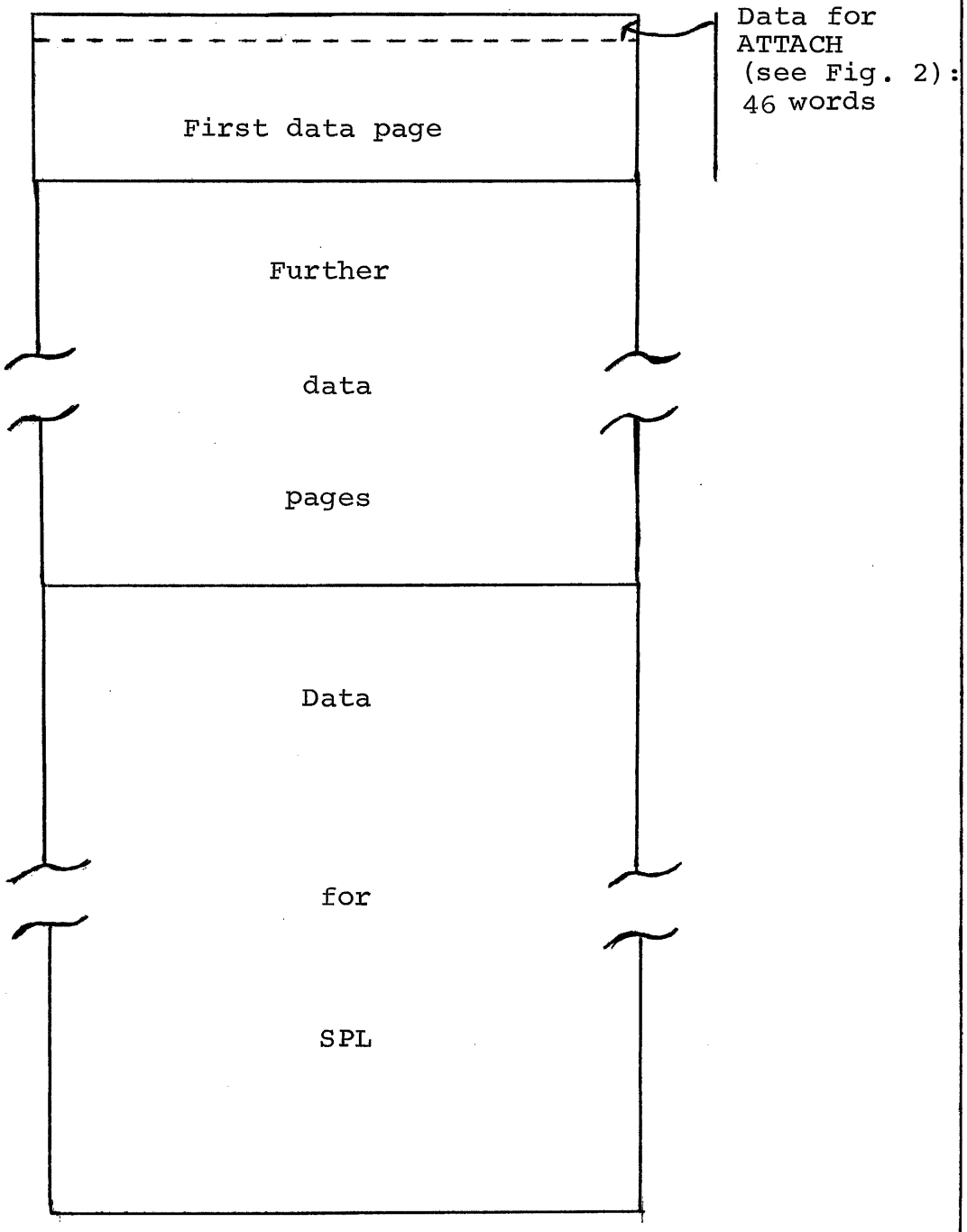
Action:

The file FN is opened and a sub-process is created using SPTX as a parameter. Page Ø of the file is examined and the SPT entry is set according to the header. Then using the file map, the pages of the file are placed in PMT or copied to a new page and the new sub-process' map is constructed. Finally the file is closed. If an error arises during the execution of ATTACH, the sub-process is destroyed, all memory is released, and the file closed. All PMT bytes created by ATTACH will be controlled by the new sub-process. Initially ATTACH will be part of the standard utility.

Real Name Table

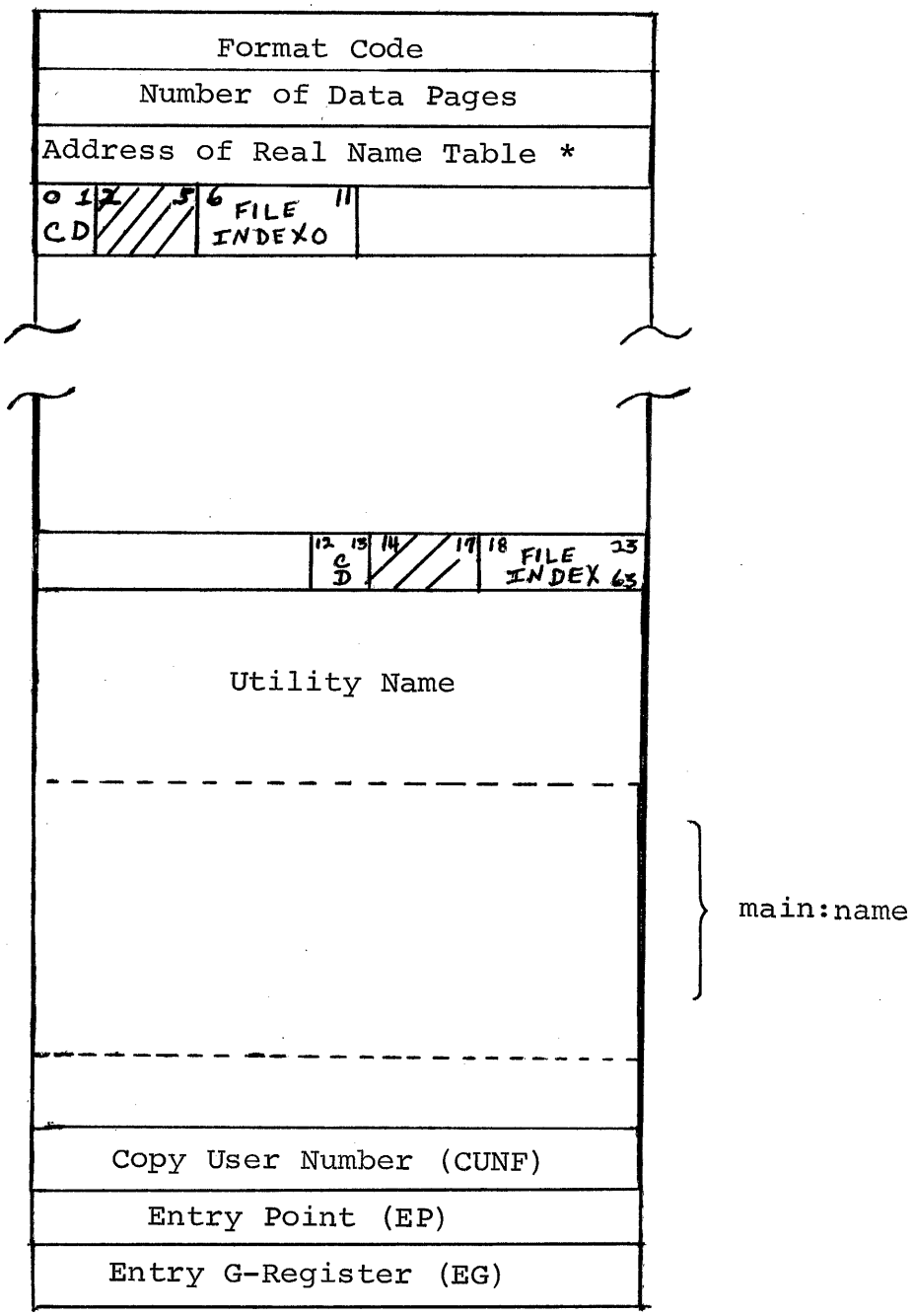
The purpose of the real name table is for "bare machine" simulation. Thus it only concerns operating system programmers. (Details forthcoming in the SPL debugging manual.)

The table pointed to by the second word of the file is shown in Figure 3. The address given in the header is the virtual address of the table and thus must be mapped. This table contains the real name of each page, and for 'bootstrap' programs will also contain its initial core address. A program will be written to generate a system image tape from these files. This table will also be used to simulate the "bare machine".



Outline of Program Image

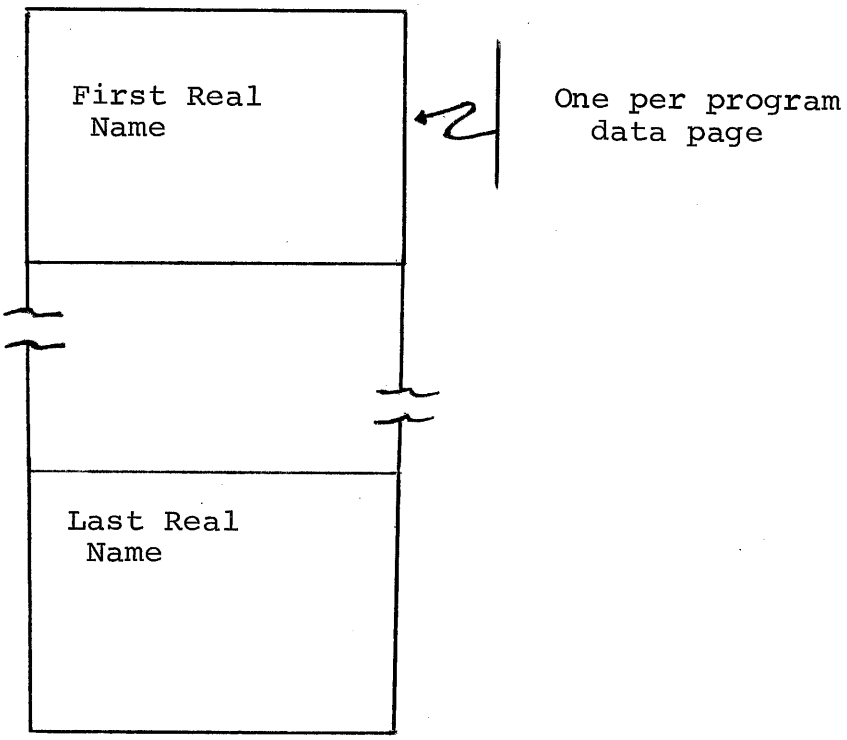
Figure 1



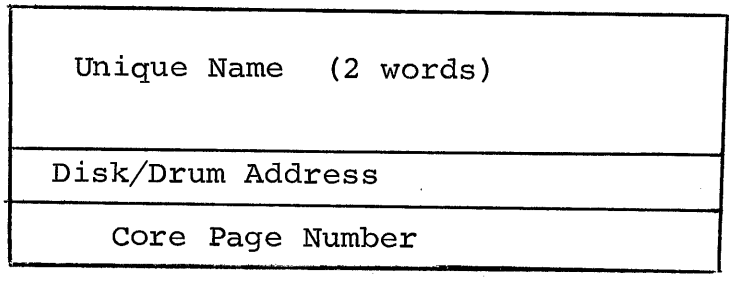
* See Figure 3.

Data Used by ATTACH and Tape Writer

Figure 2



a) Real Name Table



b) Data in arbitrary table entry

Format of Real Name Table

Figure 3