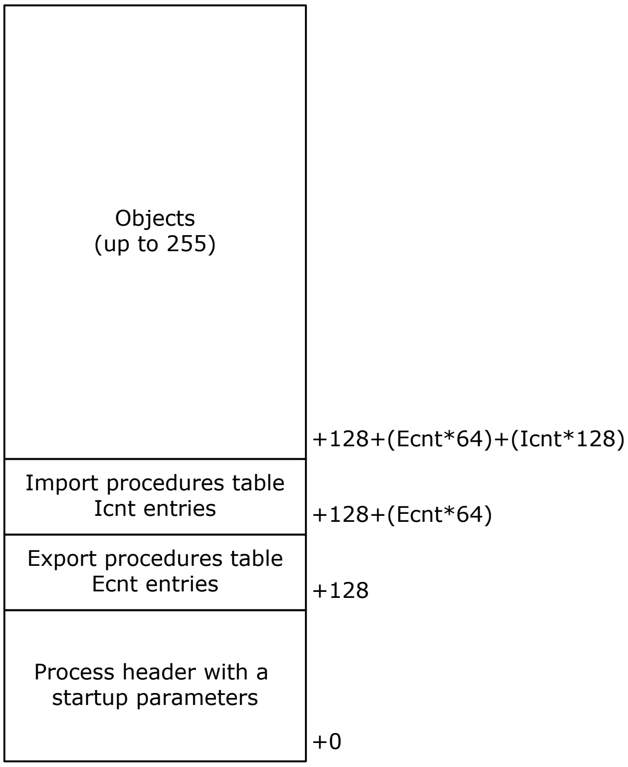
Contents

*.process files format 1
Process header. 2
Object format description. 2

*.process files format

This file is used by Kernel for to create process files and run the process. *.process the file contains a process header with start parameters, a table of exported procedures, a table of imported procedures, and a set of objects.



Objects can be from 1 to 255. Objects are located one after another. Their length is always a multiple of the 32-byte paragraph. Therefore, when the pointer reaches the length of the object containing the *.process file when expanding the process in memory, the Kernel fixes the completion of the process creation.

After creating a process and running it for execution, the object containing the PROCESS file must still remain in RAM, since many of its data are used to dynamically link processes by messages, to evaluate the status of the process, and to find all objects belonging to the process.

Process header.

Offset 48h in the process file always contains four null bytes. It is filled in by the Kernel when the process is created in memory. The Kernel sets the PSO selector of the created process to this position and resets it to 0 when the process is deleted using the KP /Kill Process/ command, while the object with the PROCES file remains in memory and will make it possible to re-create the process.

Offset	Length	Description
0/00h	8	Header 'PROCESS',00h.
8/08h	64	Name of process. Zero-terminated string.
72/48h	4	PSO selector. Contains zero dword in PROCESS file. This position used by Kernel to store a PSO selector.
76/4Ch	4	Initial object index, initial CSR value. CPL, flags and TASK ID.
80/50h	4	Code offset entry point.
84/54h	4	Maximum allowable memory. In 32-byte paragraphs
88/58h	4	Maximum count of objects, what process can create.
92/5Ch	4	System stack length in bytes.
96/60h	4	Process timer /in 200ns ticks/. For example, value 4000h gives 4000h*0.2=3276.8 μs activation period.
100/64h	2	System messages queue length /messages/.
102/66h	2	Regular messages queue length /messages/.
104/68h	2	Stack size for CPL=0, in 32-byte paragraphs.
106/6Ah	2	Stack size for CPL=1, in 32-byte paragraphs.
108/6Ch	2	Stack size for CPL=2, in 32-byte paragraphs.
110/6Eh	2	Stack size for CPL=3, in 32-byte paragraphs.
112/70h	2	Export table count. /ECnt/
114/72h	2	Import table count. /ICnt/
116/74h	12	Reserved
128/80h	ECnt*64	Export procedures table. 58 bytes procedure name, 1 byte of index of object, 1 byte procedure type and 4 byte procedure offset.
128+ECnt*64	ICnt*128	Import procedures table. 64 bytes process name and 64 bytes procedure name.
128+ ECnt*64+ ICnt*128		First object of process.
128+ ECnt*64+ ICnt*128+Object length		Next object of process.

Object format description.

First object begins from 128+ECnt*64+ICnt*128 displacement in the process file.

Offset	Length	Description
0/00h	60	Object name, zero-terminated string.
60/3Ch	4	Object selector (used at runtime) Kernel use this position

		at a runtime.
64/40h	4	Length, in 32-byte paragraphs. Including this 72-byte header
68/44h	4	Code configuration dword. Bytes 0-2 will in the future contain a config. object selector required to program a custom extension of the base instruction set. Byte 2 encode a base instruction set. 0 – X16, 1 – X32, etc.
72/48h	~~~	Object data.