



Ideas and Techniques for smaller Executables



Marcus Winter aka muhmac/freestyle
<http://20to4.net/>



Compression Basics

The 20to4 Executable Compressor

The SuShI Introsystem



■ Entropy

- Describes disorder in a file
- Calculated based on probability of symbols
- Smaller entropy means less symbols
- Compression increases entropy

■ See the following pages

- <http://datacompression.info/>
- <http://www.maximumcompression.com/>



■ Simple Example

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Input	D	R	D	O	B	B	S	_	D	R	D	O	B	B	S	_	D	R	_	D	O
Literal	D	R	[]	O	B	[]	S	_	[-	-	-	-	-	-	-]	[]	[]	
Match			-2			-1			-8										-3	-9	
			1			1			10										1	2	

Encoding: Prefixes

0 – literal

10 – 1 byte match, 4 bit for offset

11 – longer match, 4 bit offset, 4 bit length

Original: $21 * 8 = 168$ bits

Encoded: $6 * 9 + 3 * 6 + 2 * 10 = 92$ bits

- **Low memory requirements**
- **Decompressor can be kept small**
- **Normally, compression is slower than decompression**

Compression Algorithms

Huffman Encoding

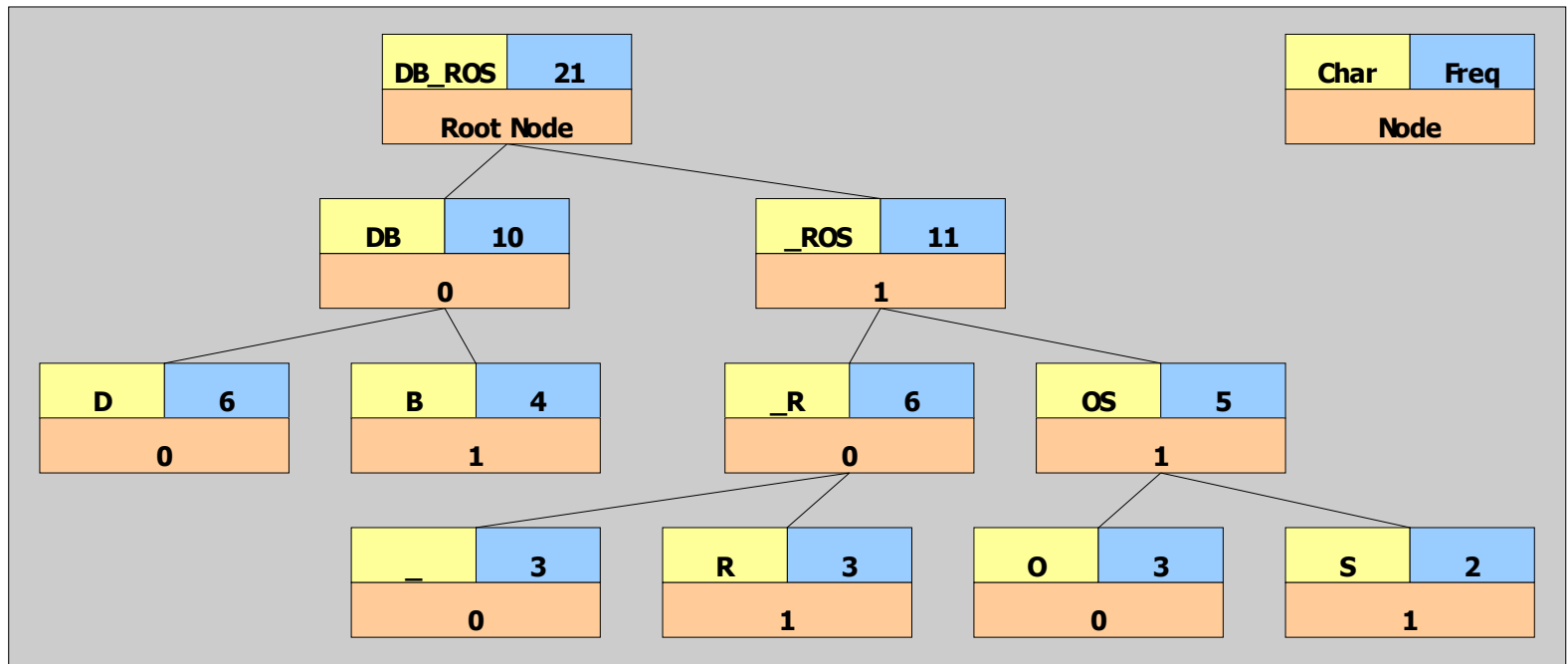


Example

Input
Code

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
D	R	D	O	B	B	S	_	D	R	D	O	B	B	S	_	D	R	_	D	O
0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	1	0	1	1	0	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1
	1		0			1	0		1		0			1	0		1	0		0

Tree



Original: $21 \times 8 = 168$ bits

Encoded: $10 \times 2 + 11 \times 3 = 53$ bits

Compression Algorithms

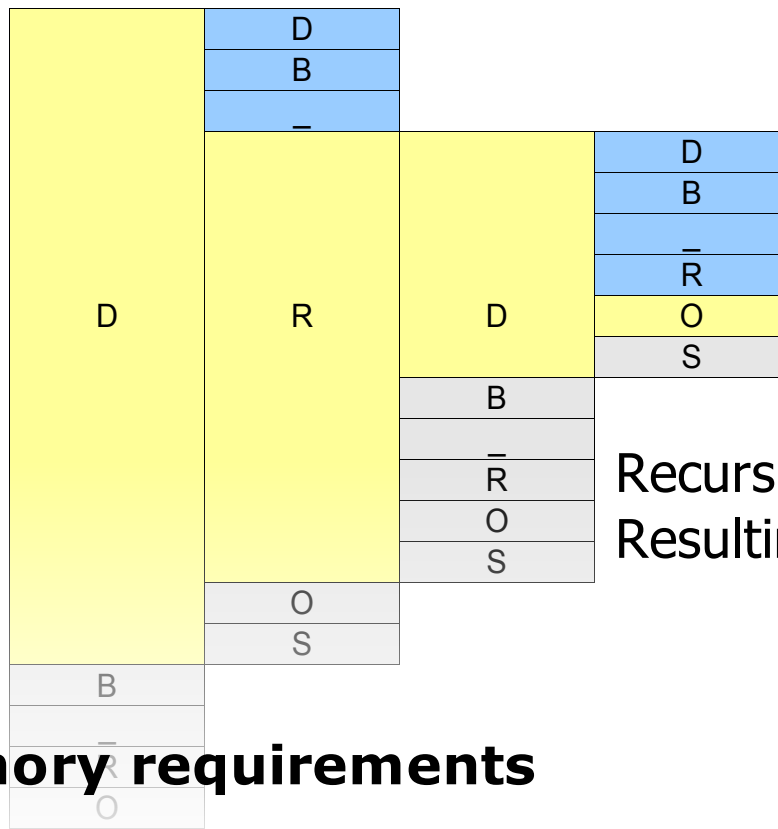
Range Coding



Example

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Input	D	R	D	O	B	B	S	_	D	R	D	O	B	B	S	_	D	R	_	D	O	
Code	0	0,18	0,18	0,19																		

Symbol	Probability	Cumulative
D	0,29	0
B	0,19	0,29
_	0,14	0,48
R	0,14	0,62
O	0,14	0,76
S	0,1	0,9



Recursive embedding of probability intervals
Resulting value identifies input stream

- Low memory requirements
- Fast decompression
- Limited accuracy is a problem



■ Simple code

- Code compresses only 2:1 (usually even less)
- Data can be layed out for optimal compression performance
- Data driven architectures usually perform better
- Virtual machine is a good choice

■ Reuse of code is critical

- Less code in the executable
- Good side effect: less errors
- Lots of the subroutines for sound and graphics are quite similar
- Example: Interpolators can be shared easily



- **Improvement of compression of 4k intros**
 - Decompressor size is critical
 - Target OS is Windows
- **Optimisation of PE File structure**
 - Headers
 - Sections
 - Imports
- **Microsoft CAB compression**



■ Concept

- Image of memory block
- Unused parts are left out

■ Directory

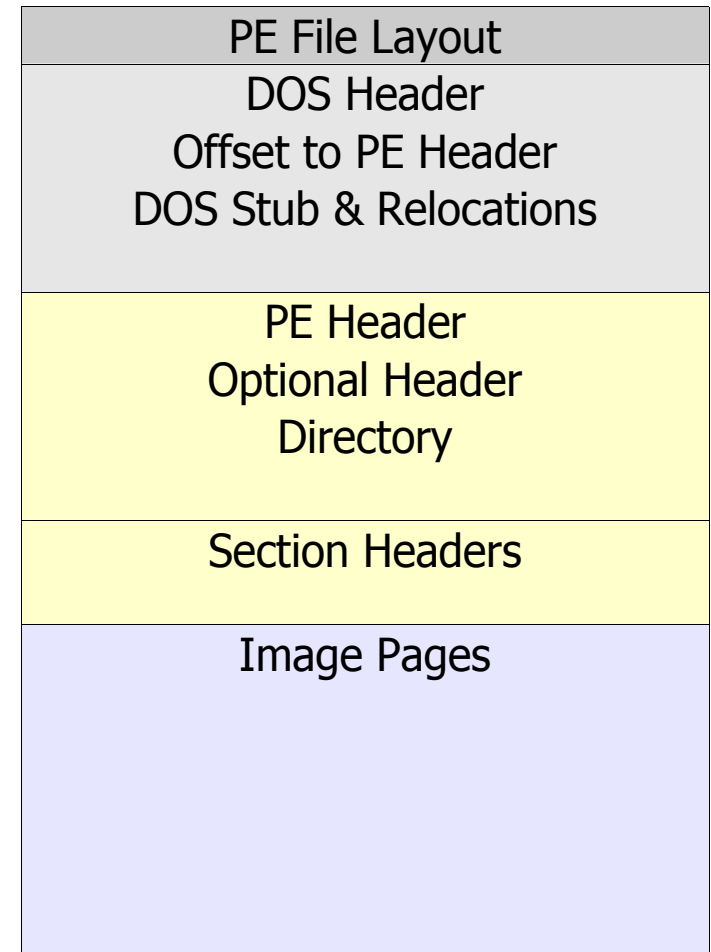
- Offsets and sizes of imports and exports

■ Section headers

- Relative location and size in memory
- Location and size in file
- Protection information

■ Image pages

- Contain actual data
- Must be aligned





■ Header cleanup

- Most of the values in the headers are not checked
- Headers can be interleaved

■ Section realignment

- Removal of trailing zeroes
- Sections are initialised to zero up to their virtual size

■ Merging sections

- Sections can be combined into a single section
- Protection suffers, but file usually compresses better

■ Reorganisation of imports

- Import by name
- Import by number
- Hashed import

■ **Compression**

- MSZIP – bad performance on small files
- LZX – better than apack or upx
 - Pattern Matching, Lazy Matching
 - Matches are encoded using Huffman Encoding

■ **Decompression**

- Batch File
 - Output file will be a batch
 - Decompression code resides in the file name field
- Exe file
 - Functions from cabinet.dll
 - Interleaved headers
 - Decompression code resides between headers and first section



■ Batch file

```
set t=%temp%\x.exe
del %t%
extrac32 %0 %t%
%t%
del %t%
exit
```

■ PE exe decompressor

```
fdintCOPY_FILE
    mov eax,dword [ra(_destfile)]
openfile
    push 384
    push 33057
    push eax
    call [ra(_open)]
    add esp,12
    ret

cabinet_dll
    db 'cabinet.dll',0
crtddll_dll
    db 'crtddll.dll',0

    align 512

entrypoint
;   jmp endofheaders-4096+512

;       align 4

cabinet_thunks
FDICreate
    dd 80000000h|20
FDICopy
    dd 80000000h|22
FDIDestroy
    dd 80000000h|23
    dd 0

crtddll_thunks
_malloc
    dd 80000000h|427 ;va(crtddll_malloc)-2
_free
    dd 80000000h|378 ;va(crtddll_free)-2
```



- **Virtual machine**
- **Supershape object generator**
- **OpenGL texture generator**
- **Modular softsynth**
- **Scripting engine**
- **Optimising script compiler**
- **Tools for ease of scripting**



■ Organisation

- 4096 float/int registers
- Separate code and data streams
- Instructions work on register ranges, only base register is specified
- Some instructions require buffers (eg. reverb effect)
- Instruction selected from code stream
- Instruction reads from the data stream and advances the data pointer

■ Instructions

- Specification of geometry, materials, lighting and camera setup
- Generating of geometry and textures
- Interpolators, Oscillators
- Data movement



■ XML format

- Easy and reliable parsing
- Flexible and easily extendable
- Versioning is no problem

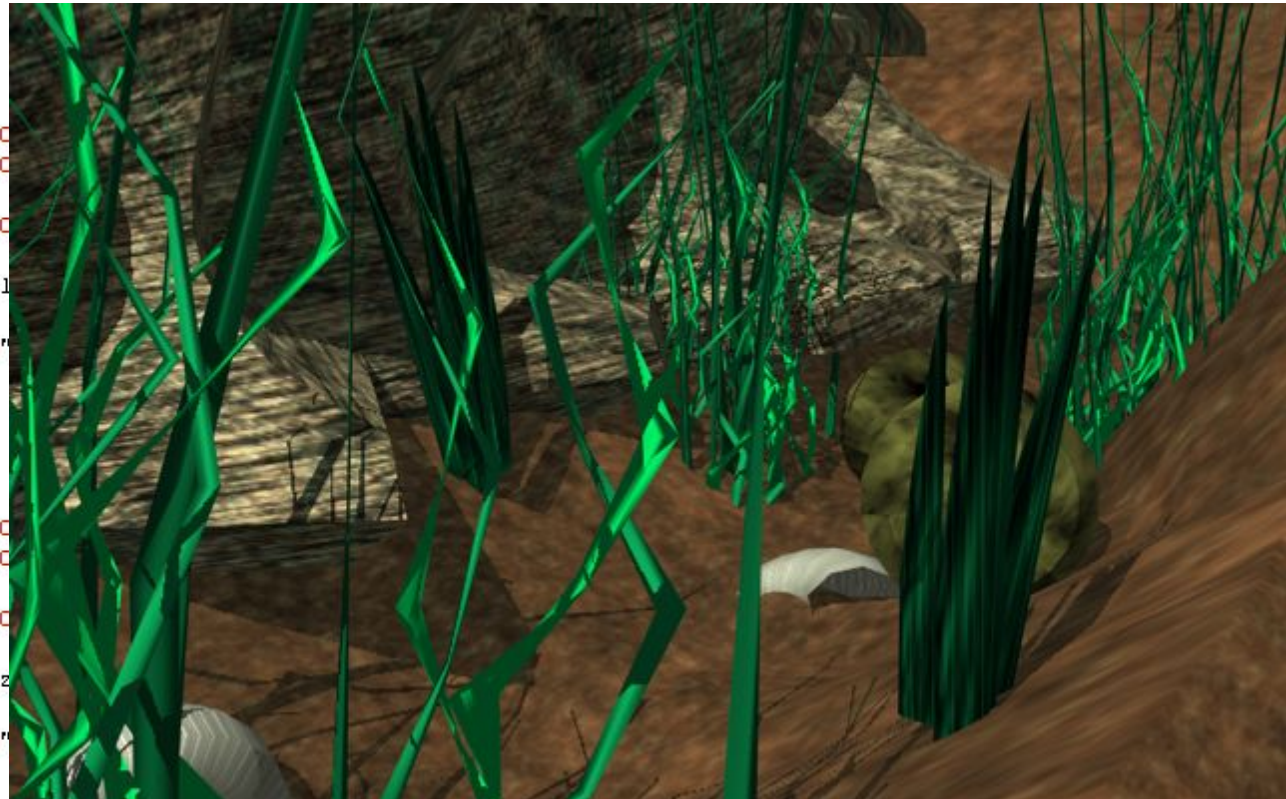
■ Scripting

- Scripts are generated by tools
- Scripting by hand should be possible (easy reordering and optimising)
- Interpolation instructions are main means of scripting
- Oscillators from soundcode can be used too

SuShI Introsystem Example Script



```
<!-- s_blatt_2 -->
<supershape name="s_blatt_2"
  baseshape="1"
  r0="1.000000" rphi="-1.000000"
  s1_m="4.000000" s1_a="1.000000" s1_b="1.000000"
  rfunc="0" rfunca="1.000000" rfuncb="1.000000"
  vfunc="1" vfunca="40.000000"
  s2_m="20.000000" s2_a="1.000000" s2_b="1.000000"
  phires="7" thetares="3"
  scalex="-20.000000" scaley="20.000000" scalez="1.000000"
  phimin="-180.005310" phimax="700.020691"
  thetamin="-270.007965" thetamax="90.002655" />
<!-- end s_blatt_2 -->
<!-- s_troeten_blatt -->
<supershape name="s_troeten_blatt"
  baseshape="1"
  r0="1.000000" rphi="0.000000"
  s1_m="4.000000" s1_a="1.000000" s1_b="1.000000"
  rfunc="2" rfunca="1.000000" rfuncb="7.000000"
  vfunc="1" vfunca="40.000000"
  s2_m="20.000000" s2_a="1.000000" s2_b="1.000000"
  phires="1" thetares="11"
  scalex="-6.000000" scaley="1.000000" scalez="1.000000"
  phimin="120.003540" phimax="220.006485"
  thetamin="-90.002655" thetamax="270.007965" />
<!-- end s_troeten_blatt -->
<!-- s_schnecke -->
<supershape name="s_schnecke"
  baseshape="1"
  r0="-3.000000" rphi="0.000000"
  s1_m="20.000000" s1_a="1.000000" s1_b="1.000000" s1_n1="100.000000" s1_n2="100.000000" s1_n3="100.000000"
  rfunc="1" rfunca="4.000000" rfuncb="1.000000"
  vfunc="1" vfunca="3.000000"
  s2_m="30.000000" s2_a="1.000000" s2_b="1.000000" s2_n1="100.000000" s2_n2="30.000000" s2_n3="100.000000"
  phires="11" thetares="9"
  scalex="-100.000000" scaley="100.000000" scalez="100.000000"
  phimin="-0.000012" phimax="220.006485"
  thetamin="-90.002655" thetamax="270.007965" />
<!-- end s_schnecke -->
<!-- s_saeule -->
<supershape name="s_saeule"
  baseshape="1"
  r0="1.000000" rphi="0.000000"
  s1_m="4.000000" s1_a="1.000000" s1_b="1.000000"
  rfunc="0" rfunca="1.000000" rfuncb="1.000000"
  vfunc="1" vfunca="40.000000"
  s2_m="20.000000" s2_a="1.000000" s2_b="1.000000"
  phires="7" thetares="3"
  scalex="-20.000000" scaley="20.000000" scalez="1.000000"
  phimin="-180.005310" phimax="700.020691"
  thetamin="-270.007965" thetamax="90.002655" />
<!-- end s_saeule -->
```





■ Features

- Pattern analysis
- Instruction independence check
- Data/State flow analysis
- Scriptcode reordering
- Removal of unnecessary statements

■ Facts

- 15000 lines of code
- Optimisations lead to compression ratios up to 10:1
- Without them compression ratio was about 4:1



- **Tools as plugins for standard software**
 - Artists do not need to learn a new interface
 - Additional functionality can be embedded in a „subinterface“
- **Modelling**
 - Cinema 4D object plugins
 - Standard materials
- **Texturing**
 - Custom texture generator
 - Embedded in Cinema 4D as a shader
- **Sound**
 - Custom software synth
 - Any MIDI sequencer can be used
 - Sound can be „programmed“ directly in the synth

SuShI Introsystem Object Generator



CINEMA 4D - [Ohne Titel 1 *]

Datei Bearbeiten Objekte Werkzeuge Selektion Struktur Funktionen Plug-ins Rendern Animation Fenster Hilfe

Ansicht
Bearbeiten Kameras Darstellung Ansicht

Objekte | Browser
Datei Bearbeiten Objekte Tags Textur
SuperShape

Attribute | Struktur | Werkzeug | Snap
Modus Bearbeiten Benutzerdaten
SuperShape Object [SuperShape]

Objekt-Eigenschaften
Basis Koord. Objekt

Base shape **Sphere**
r0 1 r(phi) 0

Supershape 1
m 11 a 1 b 1
n1 100 n2 43 n3 156
r* 1 a 1 b 1
y += 0 a 1

Supershape 2
m 4 a 1 b 1
n1 100 n2 39 n3 31

phi min -180° phi max 180°
theta min -90° theta max 90°
phi subdiv 120 theta subdiv 60
scale 100 100 100
preset specified

Koordinaten
Position Größe Winkel
X 0 m X 253.592 m H 0°
Y 0 m Y 253.592 m P 0°
Z 0 m Z 253.592 m B 0°
Objekt Abmessung Anwenden

SuShI Introsystem Texture Generator



The screenshot shows the Cinema 4D interface with a 3D pumpkin model. The 'glTexgen' window is open, displaying a list of texture layers (Layer 0-15) and a list of texture generators. The 'Material [Mat]' window is also open, showing color and texture settings.

glTexgen

Layer	Generator
Layer 0	genDots
Layer 1	fltBright
Layer 2	mixFlex
Layer 3	fltRotate
Layer 4	mixFlex
Layer 5	fltRotate
Layer 6	mixFlex
Layer 7	fltRotate
Layer 8	mixFlex
Layer 9	fltRotate
Layer 10	mixAdd
Layer 11	mixAdd
Layer 12	mixAdd
Layer 13	fltBlur
Layer 14	fltBright
Layer 15	fltScale

Material [Mat]

Farbe: R 255, G 255, B 255
Helligkeit: 80 %
Textur: SuperShader
Interpolation: MIP
Offset: 0 %
Skalieren: 0 %
Misch-Modus: Normal
Misch-Stärke: 100 %

Materialien

Key	Value
type	mixFlex
name	mixFlex
dst_layer	2
src_layer	1
mix_func	add
dst_factor	zero
src_factor	one

C4D: layer 7 scriptsize: 128 bytes

SuShI Introsystem Software Synth



SimpleSynth - Y:\tools\SimpleSynth\data\examples\muh_07.sss

File Edit

structure

connector	value
out	0.0
loop	0
mode	point
trigger	0.00
points	9
p0t	0.000
p0v	0.000
p1t	32.000
p1v	1.000
p2t	48
p2v	0.000
p3t	80
p3v	1
p4t	112
p4v	0.000
p5t	148
p5v	1
p6t	152
p6v	0.0
p7t	156
p7v	1
p8t	160
p8v	0.0

properties

settings
BPM 140 LPB 4 Snap

envelopes

done.

vol - 0.00:0.00

End



- **Have a nice day!**

