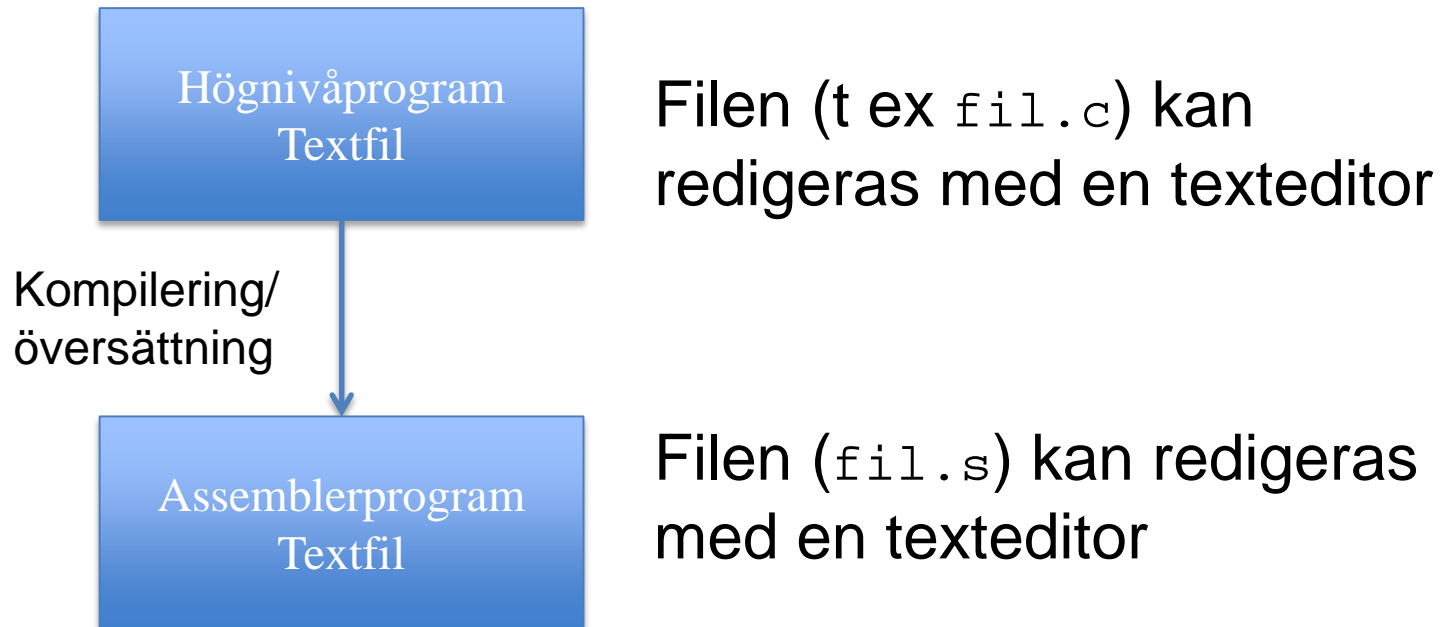


Assembler & Länkning

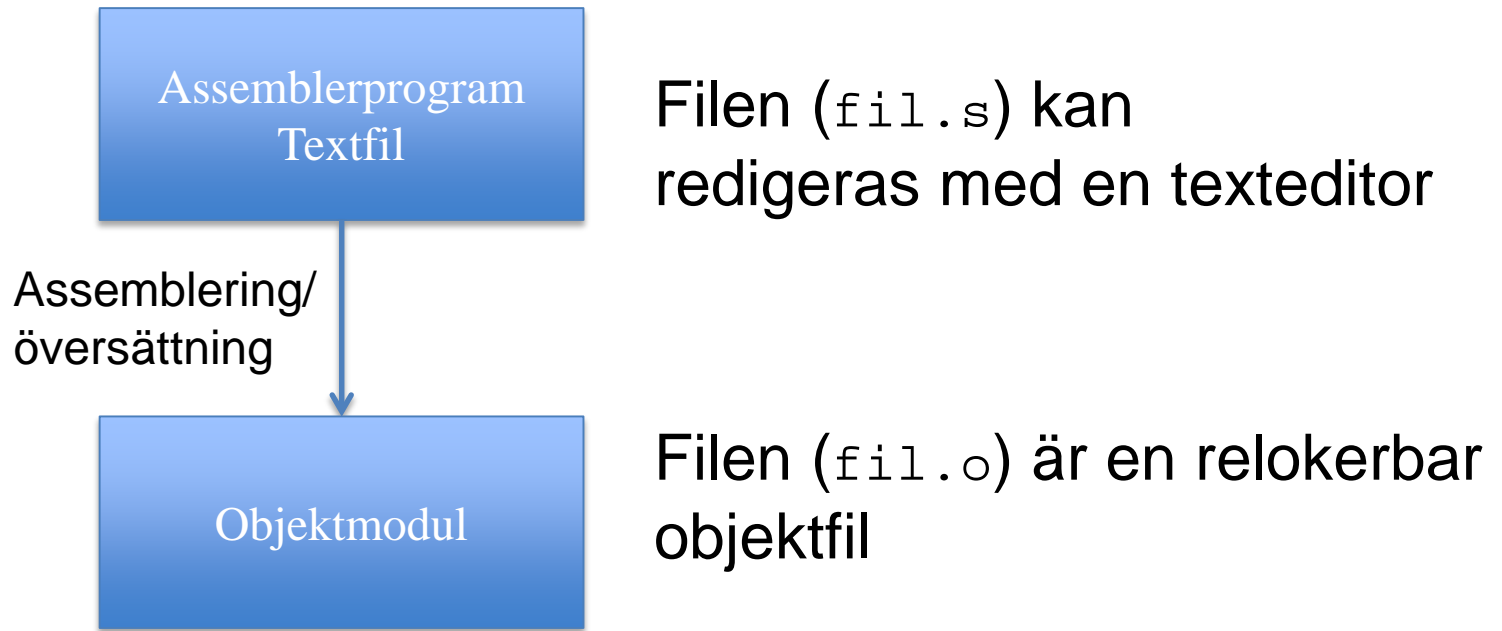
Steg i programutveckling

- Editera program i högnivåspråk
- Kompilera högnivåspråk till assemblerkod
- Editera programkod i assembler
- Assemblera assemblerkod till objektkodsmodul
- Länkning: Objektmodul(er) → Laddningsbar modul
- Laddning av laddningsbar modul till minne
- Körning/exekvering av program

Kompilering: Översättning av högnivåkod till assemblerkod



Assemblering: Assemblerkod → objektkod



Relokerbar = Ännu inte bunden till fasta minnesadresser (ej länkad)

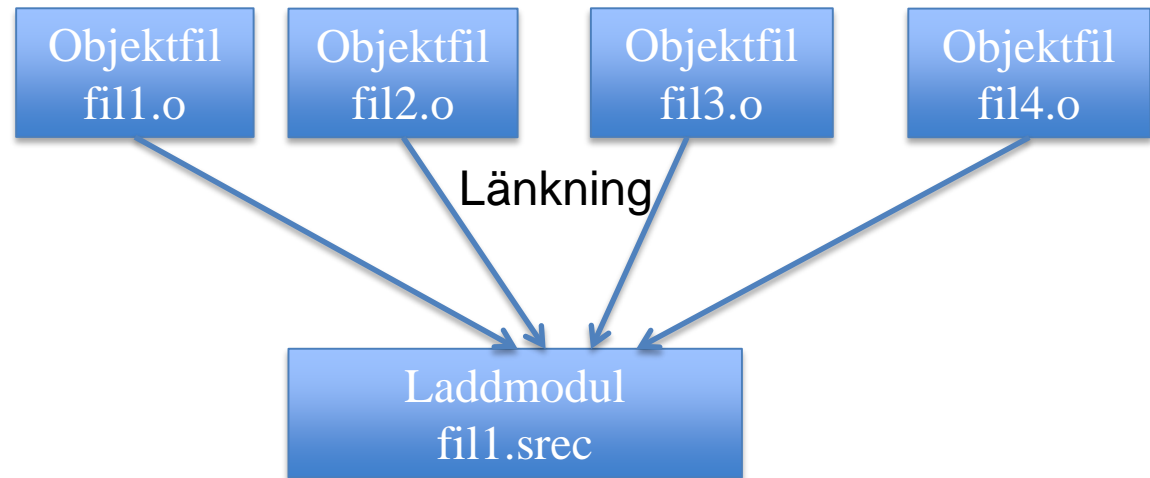
Länkning:

"Syr ihop" objekt modul(er) minnesmässigt till en enhet

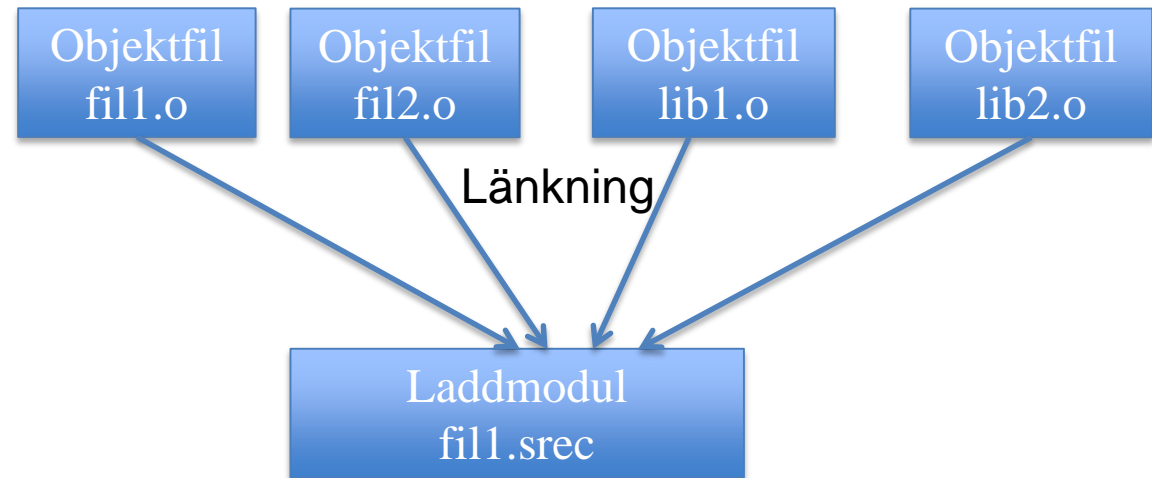


Det är vid länkningen som programdelarnas plats i minnet bestäms och referenser mellan olika programdelar löses.

Länkaren kan länka flera objektmoduler till en laddmodul



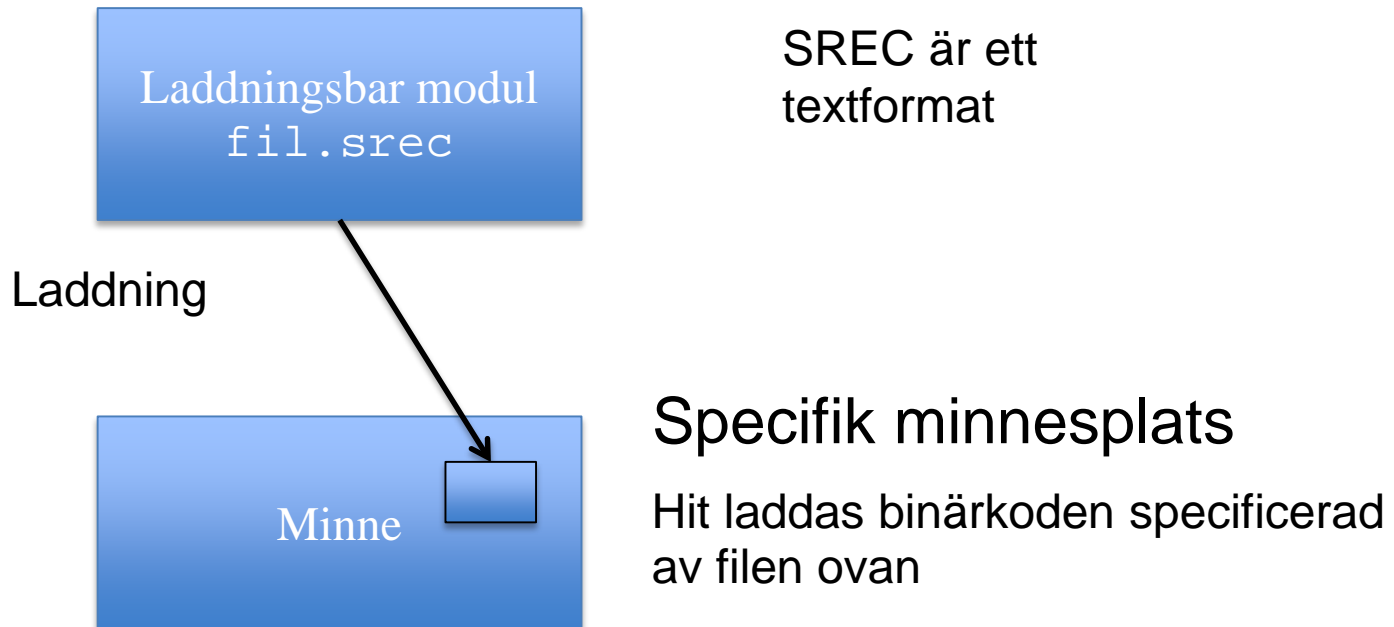
Inlänkning av biblioteksfiler



Länkning, forts

- Länkaren kan foga ihop godtyckligt många objektmoduler
- Länkaren fogar även in befintliga, redan kompillerade, biblioteksmoduler

Laddning: Laddningsbar modul laddas till minnet

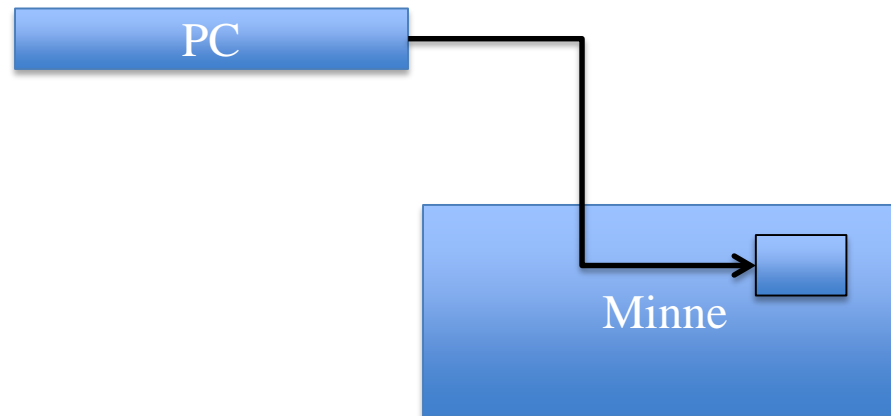


Laddning forts.

Laddningen görs i följande steg:

1. Huvudet i laddfilen läses för att avgöra storleken på programmet
2. Ett adressutrymme stort nog allokeras i minnet.
3. Instruktionerna kopieras in från filen till minnet.
4. Om programmet har parametrar kopieras de in.
5. Register och stackpekare initieras.
6. En "start up"-rutin körs som kallar på programmets "main"-funktion.

Körning/exekvering av program: Programmet körs av CPU:n



Programräknarens värde sätts så att
instruktionerna kan hämtas från minnet

Vad består en objektfil av?

Vanligtvis finns sex olika delar i en objektfil:

1. **Huvudet:** Anger storlek och position för de övriga delarna av objektfilen
2. **Text-segment:** Själva maskinkoden
3. **Statiskt data-segment:** Data-area allokerad för programmet
4. **Relokeringsinformation:** Identifierar instruktioner och data som beror av absoluta adresser
5. **Symboltabell:** Etiketter som inte är definierade i modulen t ex externa referenser
6. **Debugginginformation:** Information som förmår debuggern att associera maskinkoden till textinformation i källfilerna så att datastrukturer blir läsliga för OSS

Exempel på objektkodsformat är .ELF (Executable and Linkable Format)

Länkningen (statiskt)

Länkningen sker i flera steg:

1. Placera kodmoduler och datamoduler symboliskt i minnet
2. Bestämma adresser för data och etiketter (labels) för instruktioner
3. Lappa ihop interna och externa referenser

Dynamisk länkning (.dll)

- Länkar och laddar en modul bara först när programmet ska köras och länkar då bara de moduler som ska användas (vid anrop)
 - Rutinens kod måste vara relokerbar
 - Undviker att koden i minnet blir för stor på grund av länkning av hela bibliotek
 - Plockar in uppdaterade bibliotek automatiskt

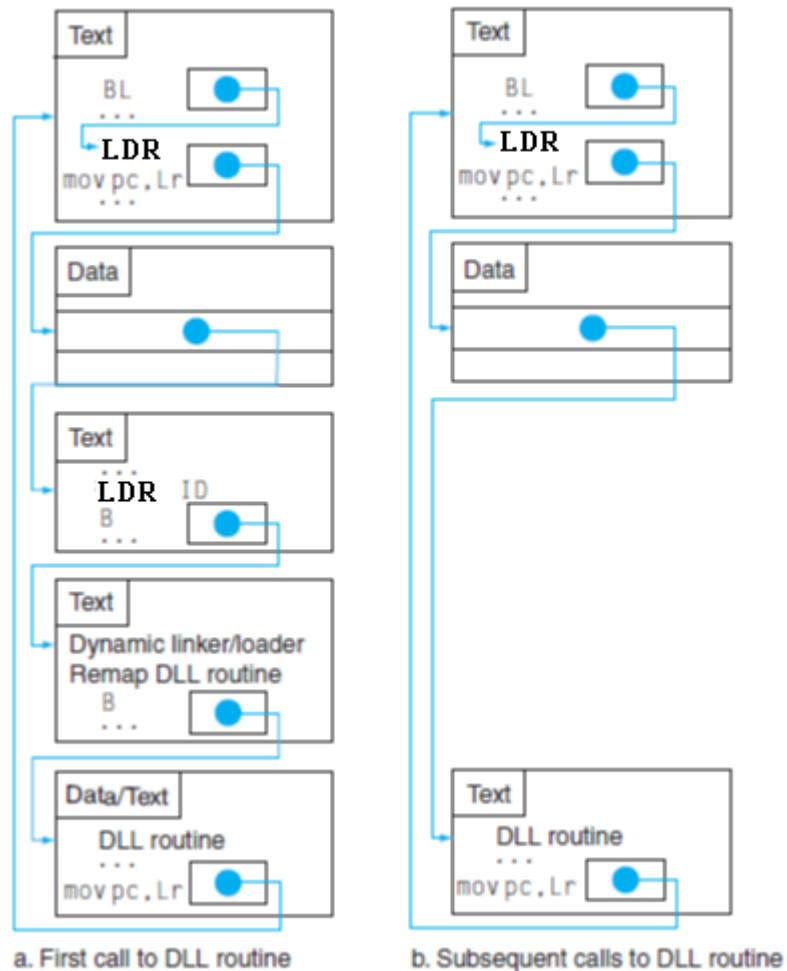
"Lazy linkage"

Omdirigerings
tabell

Identifierar rutinen ID,
Hopp till länkare/laddare

Kod för länkare/laddare

Dynamiskt mappad kod



Statisk vs dynamisk länkning

Statisk länkning

- + Statisk länkning ger snabbare exekvering
- Biblioteksrutiner blir en del av exe-filen, vilket gör att uppdateringar i bibliotek inte kan användas om programmet inte kompileras om
- Laddar alla rutiner i biblioteket vare sig de används eller inte => exe-filen blir stor

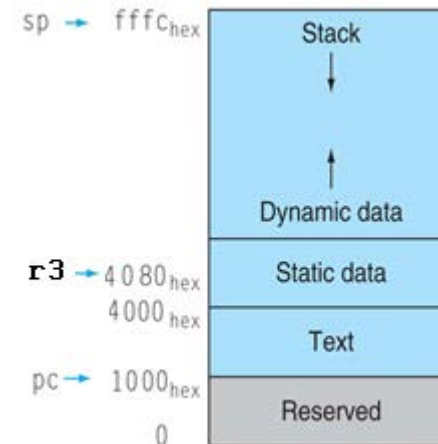
Dynamisk länkning

- + Ingen onödig kod laddas in i minnet
- + Flera processer kan dela på samma kod
- + Uppdaterade kodmoduler kan användas utan att kompilera om hela programmet
- Det tar tid att ladda in koden första gången

Länkningsexempel *(statiskt, boken s133)*

Länka de två objektfilerna. Adressreferenser till funktioner A, B och dataord X, Y måste lösas.

Objektfilshuvud			
	Namn	Funktion A	
	Textstorlek	0x100	
	Datatorlek	0x20	
Textsegment	Adress	Instruktion	
	0	LDR r0,[r3, #0]	
	4	BL 0	
	
Datasegment	0	(X)	
	
Relokeringsinfo	Adress	Instruktionstyp	Beroende
	0	LDR	X
	4	BL	B
Symboltabell	Etikett (label)	Adress	
	X	-	
	B	-	
Objektfilshuvud			
	Namn	Funktion B	
	Textstorlek	0x200	
	Datatorlek	0x30	
Textsegment	Adress	Instruktion	
	0	STR r0,[r3, #0]	
	4	BL 0	
	
Datasegment	0	(Y)	
	
Relokeringsinfo	Adress	Instruktionstyp	Beroende
	0	STR	Y
	4	BL	A
Symboltabell	Etikett (label)	Adress	
	Y	-	
	A	-	



Huvud för laddfil		
	Textstorlek	0x300
	Datatorlek	0x50
Textsegment	Adress	Instruktion
	0x1000	LDR r0,[r3,#-0x80]
	0x1004	BL 0x00FC

	0x1100	STR r0,[r3,#-0x60]
	0x1104	BL 0xFEFC

Datasegment	Adress	
	0x4000	(X)

	0x4020	(Y)

Egentligen gör ARM PC-relativa hopp vilket ger offset 0x00FC (0x1100 – 0x1004) respektive 0xFEFC (0x1000 – 0x1104)

SREC-formatet

- Format för utvecklingsverktyg för programmering av inbyggda processorer
- Hexadecimal text som läses och laddas binärt till målmaskinen.
- Varje del (record) följs av en checksumma, för att kontrollera att överföringen till målmaskinen gått bra.

Korskompilering

- Om man kompilerar kod för ett målsystem på en annan dator (utvecklingssystem) kallas förfarandet *korskompilering*
- Korskompilering är vanligt vid utveckling av inbyggda system (*embedded*)
- Vanligt format för att överföra objektfiler *.srec*
- Vi tittar på det formatet eftersom de är "*human readable*"

Struktur hos SREC-formatet

1. Startkod, ett tecken, ett S
2. Blocktyp, en siffra 0-9, definierar typ av datablock
3. Antal byte, två siffror, antal byte (adress, data och checksumma)
4. Adress, fyra, sex eller åtta siffror preciserat av blocktypen.
5. Data, 2 siffror per byte
6. Checksumma, två siffror, summera tvåsiffrors hextal från punkt 3-5. Ta de två minst signifikanta siffrorna och invertera alla bitar.

Olika blocktyper

Det finns 8 olika blocktyper

Blocktyp	Beskrivning	Adressbyte	Datasekvens
S0	Blockhuvud	2	Ja
S1	Datasekvens	2	Ja
S2	Datasekvens	3	Ja
S3	Datasekvens	4	Ja
S5	Antal block	2	Nej
S7	Slut på block	4	Nej
S8	Slut på block	3	Nej
S9	Slut på block	2	Nej

Innehåll i blocken

S0 - Filnamn

S1, S2, S3 – Datasekvens beroende av
adresseringsrymdens storlek 16-bitsadresser =>
S1

S5 – Antal block som tidigare överförts

S7, S8, S9 – Adressfältet här anger startadressen
för programmet



Exempel (lab11.s)

```
.data
numbers:
    .word  2, 3, 8, 3, 9, 12, 0
sum:
    .word  0

.text
    .globl main
main:
    LDR    r1, =numbers
    MOV    r0, #0
again:
    LDR    r2, [r1]
    CMP    r2, #0
    BEQ    finish
    ADD    r0, r0, r2
    ADD    r1, r1, #4
    BAL    again
finish:
    LDR    r1, =sum
    STR    r0, [r1]
halt:
    BAL    halt
.end
```

Vår fil översatt till .srec

(ej länkad)

```
S00D00006C616231312E7372656386
S113000024109FE50000A0E3002091E5000052E3E6
S11300100020000A020080E0041081E2F9FFFFEA16
S113002008109FE5000081E5FEFFFFEA00000000E4
S10700301C000000AC
S113000002000000030000000800000003000000DC
S1130010090000000C00000000000000000000000C7
S9030000FC
```


Länkad fil

S00C0000546573742E7372656378
S2140101342F6C69622F6C642D6C696E75782D6172F3
S20D0101446D68662E736F2E330000
S214010150040000001000000001000000474E55009A
S2140101600000000002000000060000002000000061
S214010170040000001400000003000000474E550074
S21401018013EBBB33CFFC1AC23AB62DA87B57138BA1
S2080101903B684896E4
S214010194020000000200000001000000050000004B
S2140101A40048022002000000030000002F4E3DF626
S2080101B47DED110FB7
S2140101B8000000000000000000000000000000031
S2140101C823000000000000000000000020000000DE
S2140101D8110000000000000000000000012000000EE
S2140101E80B0000000000000000000000012000000E4
S2140101F8006C6962632E736F2E360061626F7274CB
S214010208005F5F6C6962635F73746172745F6D61CE
S214010218696E005F5F676D6F6E5F73746172745F9E
S2100102285F00474C4942435F322E340011
S20C0102340000000002000200B8

....

S21401023C0100010001000000100000000000000099
S21401024C1469690D00000200320000000000000075
S20C01025CA805020015010000CF
S2140102649C05020016020000A0050200160100000B
S20C010274A405020016030000B8
S21001027C08402DE91D0000EB0880BDE8DD
S21401028804E02DE504E09FE50EE08FE008F0BEE50A
S214010298F802010000C68FE210CA8CE2F8F2BCE54B
S2140102A800C68FE210CA8CE2F0F2BCE500C68FE207
S20C0102B810CA8CE2E8F2BCE575
S2140102C000B0A0E300E0A0E304109DE40D20A0E14F
S2140102D004202DE504002DE510C09FE504C02DE5A2
S2140102E00C009FE50C309FE5EBFFFFEBF0FFFFEB0B
S2140102F080040100E80301001C04010014309FE59E
S21401030014209FE503308FE0022093E7000052E3BC
S2140103101EFF2F01E3FFFFEA840201001800000020
S2140103201C309FE51C009FE5033060E0060053E3A8
S2140103301EFF2F9110309FE5000053E31EFF2F0193
S21401034013FF2FE1D7050200D405020000000000CC
S21401035024109FE524009FE5011060E04111A0E113
S214010360A11F81E0C110B0E11EFF2F0110309FE5F3
S214010370000053E31EFF2F0113FF2FE1D4050200F7
S214010380D40502000000000010402DE918409FE54A
S2140103900030D4E5000053E31080BD18DFFFFFFEB0B
S2140103A00130A0E30030C4E51080BDE8D4050200AA
S2140103B008402DE924009FE5003090E5000053E356
S2140103C00100001A0840BDE8E0FFFFEA10309FE593
S2140103D0000053E3FAFFFF0A33FF2FE1F8FFFFEABD
S2140103E0A40402000000000024109FE50000A0E322
S2140103F0002091E5000052E30200000A020080E0BE

....



....
S214010400041081E2F9FFFFEA08109FE5000081E58C
S214010410FEFFFFEAB4050200D0050200F8432DE90D
S2140104200070A0E14C609FE54C509FE506608FE0B0
S21401043005508FE0066065E00180A0E10290A0E132
S21401044408DFFFFE0B4661B0E1F883BD08045045E23D
S2140104500040A0E3014084E20430B5E50700A0E1D6
S2140104600810A0E10920A0E133FF2FE1060054E1C6
S214010470F7FFFF1AF883BDE86C0001006400010075
S2080104801EFF2FE145
S20C01048408402DE90880BDE8DF
S20801048C0100020063
S20C01049030FEFF7F01000000B1
S2080104980000000005A
S20802049CB0030100A1
S2080204A088030100C5
S2080204A40000000004D
S2140204A801000000010000000C0000007C020100B0
S2140204B80D00000084040100190000009C040200DC
S2140204C81B000000040000001A000000A00402003E
S2140204D81C00000004000000F5FEFF6F94010100F6
S2140204E805000000F801010006000000B80101003E
S2140204F80A0000003C0000000B000000100000008C
S21402050815000000000000003000000900502002D
S214020518020000001800000014000000110000008D
S2140205281700000064020100110000005C020100CE
S2140205381200000008000000130000000800000077
S214020548FEFFFF6F3C020100FFFFF6F0100000085
S214020558F0FFFF6F340201000000000000000000F8
S214020568000000000000000000000000000000007C
S214020578000000000000000000000000000000006C
S20C02058800000000000000000064
S214020590A80402000000000000000000880201001B
S2100205A088020100880201000000000032
S2140205AC000000000000000020000000300000033
S2140205BC0800000003000000090000000C0000008
S20C0205CC00000000000000000020
S8040102C038

Vad döljer sig i första blocket?

S00D00006C616231312E7372656386

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]	}	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL