Institutionen för Data- och Systemvetenskap

STOCKHOLMS UNIVERSITET

KUNGLIGA TEKNISKA HÖGSKOLAN

***:96 (SU) and 2I1263 (KTH) Internet Application Protocols and Standards**

Exam 1999-09-18

The following documents are allowed during the exam:

a) Documents in Compendium 1, printed on coloured paper.

b) Documents in Compendium 2, printed on coloured paper.

c) Documents in Compendium 3, printed on coloured paper.

d) Ordinary language dictionaries between English and Swedish.

Note 1: Compendium 4, 5 and 6 are not allowed during the exam.

Note 2: Some students may have the compendiums from the previous time this course was given. Some of these compendiums have yellow paper only on the front page of the allowed documents, and there was a separate document Appendix A: ASN.1 syntax (basic items) which is allowed during the exam.

Note 3: Compendium 4 was wrongly printed on yellow paper in August 1998, but is not allowed during the exam.

Note 4: A few copies of these compendiums (part 1-3) will be available for loan during the exam for students who have not bought the compediums.

Important warning

It is not acceptable to answer an exam question by just a verbatim quote from the allowed documents above. You must show that you understand the question and your answer by using your own words.

Questions during the exam

Jacob Palme can be reached by phone 08-664.77 48 during the exam between 9:30-11:00.

Notification of result by e-mail

If you write your e-mail address on the front cover page of the exam, then you will be notified by e-mail if you did not pass the exam.

No.	Question in English	Question in Swedish	Max points
1	Make a comparison between ASN.1/BEF, ABNF and DTD/XML as languages for specificaitons of protocol data. What are the main differences, advantages and restrictions of these methods?	Jämför ASN.1/BER, ABNF och DTD/XML som språk för specificering av protokoll-data. Vilka är det viktigaste skillnaderna, fördelarna och begränsningarna med de olika metoderna?	6

Solution:

Specification lanugage: ASN.1 and DTD allow the spefication of complex data structures and data types. ASN.1 is more strongly typed than DTD. ABNF is a more low level language for specifying more arbitrary text strings.

The human readability of the specification is a matter of opinion. I personally think that ASN.1 is easy to read and DTD is difficult to read, but other people have other opinions.

Two problems with ABNF is (a) that there are so many different variants of ABNF used in different standards, and (b) that many actual usages of ABNF does not fully specified where, and what kind of "white space" is allowed.

Encoding format: BER allows binary data, while ABNF and XML encodes text strings, binary data can only be included by conversion to text format (base64) or by indirect reference to binary data transported outside the protocol.

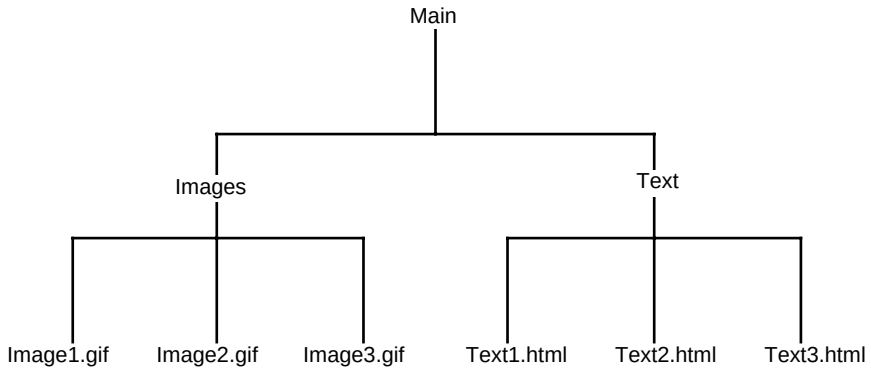
Both ABNF and XML have problems with including delimiter characters in the data transported, various kinds of "quoting" is necessary. This is no problem with BER since it uses tag-length-value to encode data sent.

XML data can, combined with style sheets, be displayed by some web browsers. XML can thus be used as a replacement for HTML, while at the same time sending data encoded in an application-specific way.

The human readability of the encoded data is good with XML even without style sheets. With ABNF, it is possible to specify data in more or less human-readable format. Some e-mail header fields, specified with ABNF, like "From:", "Date:" and "Subject:", are quite readable.

Neither BER nor XML does specify one single unique encoding for data, the same data can be encoded in more than one way, which is a disadvantage in some respects. With ABNF, since it is so low-level, data may be uniquely or non-uniquely encoded depending on how it is specified.

Encoded XML data tends to require more octets than BER and ABNF, but on the other hand, XML data is will suited for automatic compression.

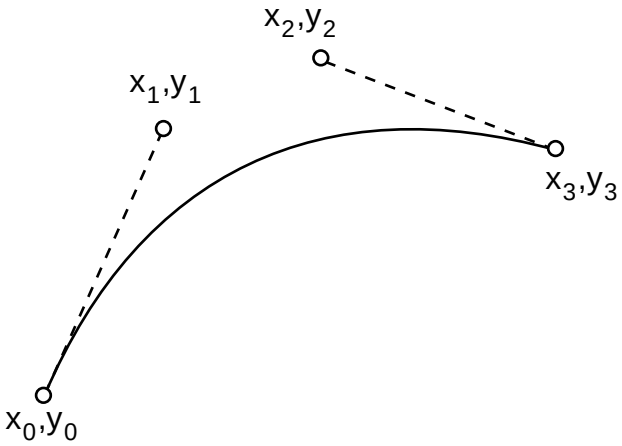
<table>
<tr><th>No.</th><th>Question in English</th><th>Question in Swedish</th><th>Max points</th></tr>
<tr><td>2</td><td>A Beziér curve segment is specified by a start point (x_0,y_0), an end point (x_3,y_3) and two control points (handles) (x_1,y_1) and (x_2,y_2).
Several curve segments can be combined in a line, where the end point of each segment is the start point of the next segment.
Write an ASN.1 specification to transfer such a line of Beziér curve segments.</td><td>Ett Beziér-kurv-segment specificeras av en start-punkt (x_0,y_0), en slut-punt (x_3,y_3) och två kontrollpunkter (x_1,y_1) och (x_2,y_2). Flera segment kan kombineras i en linje, där slutpunkten på varje segment är startpunkt på nästa segment.
Skriv en ASN.1-specification för att överföra en sådan serie av Beziér-kurvsegment.</td><td>6</td></tr>
<tr><td></td><td colspan="2">Solution:
Line ::= SEQUENCE OF Segment
Segment ::= SEQUENCE {
start Point,
handle1 Point,
handle2 Point,
end Point OPTIONAL }
Point ::= SEQUENCE {
x REAL,
y REAL }</td><td></td></tr>
<tr><td>3</td><td colspan="2"> 
</td><td>6</td></tr>
<tr><td></td><td>The directory Main has two sub-directories Images and Text. The Images directory contains gif files, the Text directory contains html documents using these gif files. How can such an html document refer to one of the images?
Your solution should work independently of the position of the directory Main, and not require rewriting of the html code if Main with its subdirectories is moved.</td><td>Katalogen Main har två under-kataloger Images och Text. Katalogen Images innehåller gif-filer, katalogen Text innehåller html-dokument. Hur kan ett sådant HTML-dokument referera till en av bilderna?
Lösningen måste fungera oberoende av var katalogen Main befinner sig, och inte kräva att html-koden skrivs om om Main och dess underkataloger flyttas.</td><td></td></tr>
</table>

No.	Question in English	Question in Swedish	Max points
	Solution: <IMG src="../Images/Image1.gif"> Note: Many students provided various solutions which contained explicit absolute URLs in the BASE or IMG statement. But the question requred a solution where the directory Main can be moved without changing the code, and then no solution with absolute URLs in the source will do what is required.		
4	Threads are sequences of e-mail messages which directly or indirectly refer to each other. How is this information specified when the messages are transported.	Trådar är sekvenser av e-post-meddelanden, som direkt eller indirekt hänvisar till varandra. Hur specificeras denna information när meddelandena överförs.	6
	Solution: Each message has a "Message-ID:" header. Replies have an "In-Reply-To:" header with the Message-ID of the previous message. Messages in a thread also often hav a "References:" header with a list of the Message-ID for the whole chain from the message to the first message in the thread. Keeping the "Subject:" intact, except for additions of "Re: ", is also sometimes used to indicate the thread.		

The Math Behind the Bézier Curve

(This information is not necessary in order to answer the exam question, but is included in case you are interested.)

(Based on a web page by Darell Plant, at URL: http://moshplant.com/direct-or/bezier/math.html)



(This description is a little incomplete, additional constraints are sometimes applied.)

A cubic Bezier curve is defined by four points. Two are *endpoints*. (x_0,y_0) is the *origin* endpoint. (x_3,y_3) is the *destination* endpoint. The points (x_1,y_1) and (x_2,y_2) are *control points*.

The tangent of the Beziér curve in the point (x_0,y_0) should be the vector of the line (x_0,y_0) to (x_1,y_1) and the tangent in the point (x_3,y_3) should be the vector of the line (x_2,y_2) to (x_3,y_3).

Two equations define the points on the curve. Both are evaluated for an arbitrary number of values of t between 0 and 1. One equation yields values for x, the other yields values for y. As increasing values for t are supplied to the equations, the point defined by $x(t)$,$y(t)$ moves from the origin to the destination. This is how the equations are defined in Adobe's PostScript references.

$$x(t) = a_x t^3 + b_x t^2 + c_x t + x_0$$

$$x_1 = x_0 + c_x / 3$$
$$x_2 = x_1 + (c_x + b_x) / 3$$
$$x_3 = x_0 + c_x + b_x + a_x$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + y_0$$

$$y_1 = y_0 + c_y / 3$$
$$y_2 = y_1 + (c_y + b_y) / 3$$
$$y_3 = y_0 + c_y + b_y + a_y$$

This method of definition can be reverse-engineered so that it'll give up the coefficient values based on the points described above:

$$c_x = 3 (x_1 - x_0)$$
$$b_x = 3 (x_2 - x_1) - c_x$$
$$a_x = x_3 - x_0 - c_x - b_x$$

$$c_y = 3 (y_1 - y_0)$$
$$b_y = 3 (y_2 - y_1) - c_y$$
$$a_y = y_3 - y_0 - c_y - b_y$$

Now, simply by knowing coördinates for any four points, you can create the equations for a simple Bézier curve