

Basic Encoding Roles and ASN.1

ASN.1 je univerzální jazyk pro specifikaci datových typů. Dovoluje definovat nejen typ dat, ale i jejich velikost (rozsah hodnot) a význam. BER (Basic Encoding Roles) je jednou z možností jak data specifikovaná ASN.1 zakódovat. Kromě BER existuje DER, PER, ...

Jednoduché typy

INTEGER	hodnoty jsou kardinální čísla
OCTET STRING	Prvky v rozsahu 0 .. 255
OBJECT IDENTIFIER	odkazuje na označení objektu
NULL	rezervuje místo

Konstruované typy

SEQUENCE	Uspořádaný seznam nula nebo více elementů, každý je typem ASN.1 (heterogenní seznam)
SEQUENCE OF	uspořádaný seznam stejných elementů

„Tagged“ typy

Jsou odvozeny od předtím definovaných typů, nové a staré typy se odlišují značením (tagy – značkami)

- univerzální (universal) – definovány pro libovolné použití
- aplikační (application) – vztahují se k dané aplikaci [APPLICATION 1]
- závislé na kontextu – mají význam pouze v místě definice (řídící proměnná ve struktuře)
- privátní (private) – definované „soukromě“ [PRIVATE 1]

Kódování dat pomocí BER

Ke kódování se používá způsob v literatuře označovaný jako TLV (type, length, value – typ, délka, hodnota). Kódování tedy obsahuje explicitně vyjádřený typ hodnoty a její délku. Typ (tag), délka ani hodnota nemají omezenou délku.

Kódování čísel – dvojkový doplněk

Tag – typ, přívěšek, doplňková informace

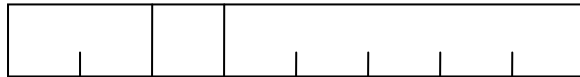
- universal – všeobecně zavedené datové typy
- application – jednoznačně určuje typ v modulu (definován v aplikaci)
- závislý na kontextu – uvádí se v SEQUENCE, CHOICE
- private – zaveden privátně

K jednoduchým „základním“ typům patří BOOLEAN, INTEGER, REAL, BIT STRING, OCTET STRING, NULL, OBJECT IDENTIFIER.

Ke strukturovaným „základním typům“ patří SEQUENCE, SEQUENCE OF, SET, SET OF.

Všechny ostatní tagy jsou z nich odvozené.

Tag



číslo tagu (typ)
 f-bit 0 – jednoduchý typ
 1 – konstruovaný typ

třída 00 – univerzální
 01 – aplikační
 10 – závislá na kontextu
 11 – privátní

číslo tagu - <31 ... definice typu
 =31 ... číslo je uloženo v následujících slabikách

Univerzální typy

BOOLEAN	1
INTEGER	2
BIT STRING	3
OCTET STRING	4
NULL	5
OBJECT IDENTIFIER	6
OBJECT DESCRIPTOR	7
EXTERNAL	8
REAL	9
ENUMERATED	10
SEQUENCE, SEQUENCE OF	16
SET, SET OF	17
Numeric String	18
Printable String	19
Teletex String	20
Videotex String	21
IA5 String	22
UTCTime	23
Generalized Type	24
Graphics String	25
Visible String	26
General String	27
Character String	28

Aplikační typy (SMI, MIB)

IpAddress ::= [Application 0] IMPLICIT OCTER STRING (SIZE (4))
--

Counter ::= [Application 1] IMPLICIT INTEGER (0 .. 4294967295)
Gauge ::= [Application 2] IMPLICIT INTEGER (0 .. 4294967295)
TimeTicks ::= [Application 3] IMPLICIT INTEGER (0 .. 4294967295)
Opaque ::= [Application 4] IMPLICIT OCTET STRING

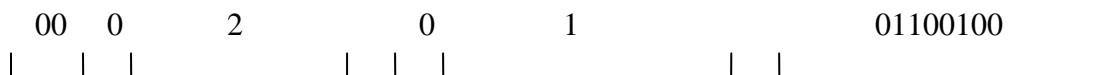
Závislý na kontextu

CHOICE

get-request	0
get-next-request	1
get-response	2
set-response	3
trap	4

Příklady kódování

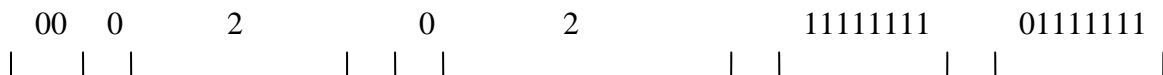
Jednoduchý typ – INTEGER



UNIVERSAL INTEGER

délka

100

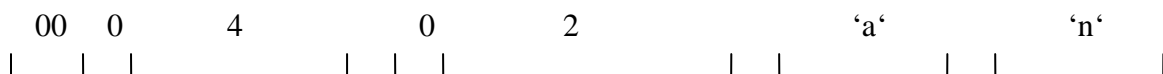


UNIVERSAL INTEGER

délka

-129

Jednoduchý typ – OCTET STRING



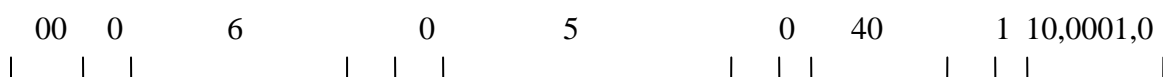
Jednoduchý typ – NULL

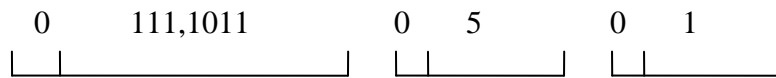


Jednoduchý typ – OBJECT IDENTIFIER

první dva elementy se kódují $X*40+Y$

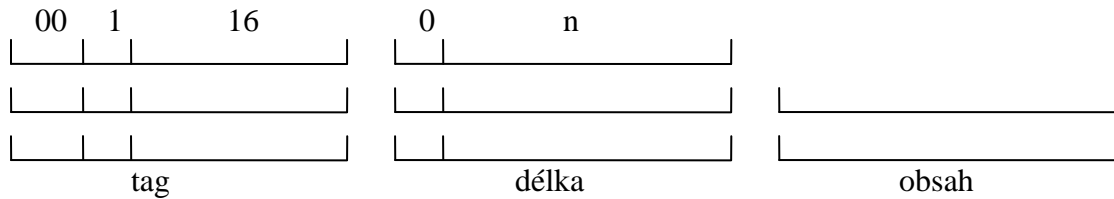
Příklad: 1.0.8571.5.1



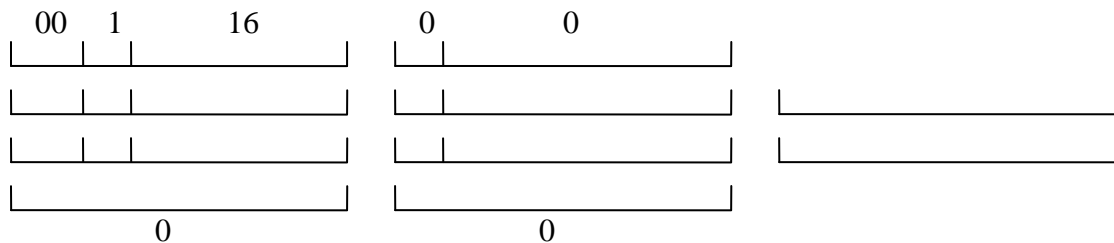


Konstruovaný typ – SEQUENCE

```
Př: Proměnná ::= SEQUENCE {
    name  ObjectName;
    value ObjectSyntax
}
```



Nebo:



Konstruovaný typ – SEQUENCE OF

- používá stejný způsob kódování, tag je redundantní

Odvozené typy

- aplikační, privátní, závislé na kontextu

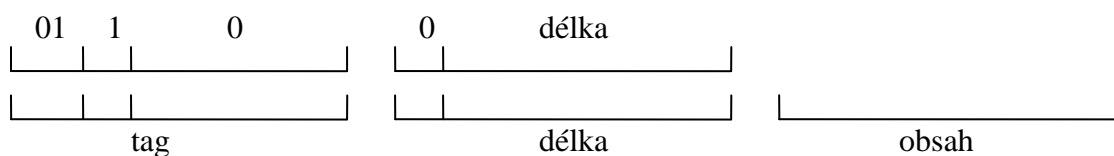
SomeType ::= [tag] IMPLICIT OtherType

- kódováno podle pravidel pro OtherType, ale jiný obsah pole tag.

Definice typu: SomeType ::= [tag] OtherType

- je chápána jako SomeType ::= [tag] IMPLICIT SEQUENCE {OtherType}

Př: SomeType ::= [APPLICATION 0] OtherType



Př: SNMP rámeček typu get-response

```
message ::= {
    version      version-1
    community    "public"
    data {
        get-response {
            request-id 17
            error-status noError
            error-index 0
            variable-bindings {
                {
                    {
                        name 1.3.6.1.2.1.1.1.1.0
                        value "UNIX"
                    }
                }
            }
        }
    }
}
```

SEQUENCE
INTEGER
OCTET STRING
CHOICE, SEQUENCE
INTEGER
INTEGER
INTEGER
SEQUENCE
SEQUENCE
OBJECT IDENTIFIER
OCTET STRING