

NOM, Prénom

Groupe

- ▷ 45 minutes – Documents non autorisés
- ▷ 1 point est réservé à la *propreté* de la copie
- ▷ Après avoir spécifié vos nom et prénom, répondez aux questions dans les espaces correspondants.

## 1 Exercices

### Exercice 1.1 : Questions de cours

(5 points)

**Q 1.** Quelle commande UNIX permet de décrire les interfaces réseaux d'un poste?

- dig
- ip
- ipconfig
- ping

**Q 2.** Le protocole SNMP distingue plusieurs types de fonctionnalités. Lesquels?

- Les alertes
- La gestion des utilisateurs distants
- L'interrogation de la MIB
- L'arrêt et la remise en route des matériels

**Q 3.** Quelle information un moniteur de supervision doit posséder pour s'authentifier auprès d'un agent SNMPv1?

- Une clé de chiffrement
- Une communauté
- La liste des droits d'accès

**Q 4.** Dans le protocole SNMP, les *traps* permettent :

- d'interroger la MIB d'un équipement distant
- d'authentifier les utilisateurs d'un agent
- de signaler à la station de supervision des événements particuliers
- de suffixer les balles

**Q 5.** Comment appelle-t-on le langage de description des bases de données des objets SNMP?

- un agent
- un moniteur
- ASN.1
- SMI
- MIB

**Q 1.** A partir de l'extrait de MIB donné en annexe 2, page 3 et de la liste des interfaces suivantes, indiquez sous forme **symbolique et numérique** l'OID qui doit être passé en paramètre d'une opération GET pour obtenir le nombre d'interfaces réseaux présentes?

```
eth0      Link encap:Ethernet  HWaddr 00:24:e8:c2:e9:61
          inet adr:192.168.0.3  Bcast:192.168.0.255  Masque:255.255.255.0
          adr inet6: fe80::224:e8ff:fec2:e961/64  Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:17656 erreurs:0 :0 overruns:0 frame:0
          TX packets:20728 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:11093411 (11.0 MB) Octets transmis:3254165 (3.2 MB)
          Interruption:22 Mémoire:f6fe0000-f7000000

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128  Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:3820 erreurs:0 :0 overruns:0 frame:0
          TX packets:3820 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:415007 (415.0 KB) Octets transmis:415007 (415.0 KB)
```

- Symbolique

- Numérique

**Q 2.** En fonction de l'extrait de MIB fournie en annexe 3, page 4, et du tableau ci-dessous, indiquez l'**objet recherché** et la **valeur trouvée** par la commande :

```
snmpget -v 2c -c public localhost 1.3.6.1.2.1.4.22.1.2.2.192.168.0.254
```

ipNetToMediaIfIndex	ipNetToMediaPhysAddress	ipNetToMediaNetAddress	ipNetToMediaType
1	0:24:d4:ca:56:01	172.16.0.1	dynamic
2	0:24:d4:b3:39:f0	192.168.0.254	dynamic
3	0:24:d4:a1:12:23	195.3.8.2	static

- Objet

- Valeur

## 2 Extrait du sous-arbre interfaces de la MIB-2

Pour mémoire .iso.org.dod.internet.mgmt.mib-2.interfaces = .1.3.6.1.2.1.2.

```
ifNumber OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The number of network interfaces"
    ::= { interfaces 1 }

ifTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF IfEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "A list of interface entries. The number of entries is [...] ifNumber."
    ::= { interfaces 2 }

ifEntry OBJECT-TYPE
    SYNTAX      IfEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "An interface entry containing objects [...] for a particular interface."
    INDEX      { ifIndex }
    ::= { ifTable 1 }

IfEntry ::=
    SEQUENCE {
        ifIndex INTEGER,
        ifDescr DisplayString,
        ifType INTEGER,
        ifMtu INTEGER,
        ifSpeed Gauge,
        ifPhysAddress PhysAddress,
        ifAdminStatus INTEGER,
        ifOperStatus INTEGER,
        ifLastChange TimeTicks,
        ifInOctets Counter,
        ifInUcastPkts Counter,
        ifInNUcastPkts Counter,
        ifInDiscards Counter,
        ifInErrors Counter,
        ifInUnknownProtos Counter,
        ifOutOctets Counter,
        ifOutUcastPkts Counter,
        ifOutNUcastPkts Counter,
        ifOutDiscards Counter,
        ifOutErrors Counter,
        ifOutQLen Gauge,
        ifSpecific OBJECT IDENTIFIER
    }

ifIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "A unique value for each interface. [...] between 1 and ifNumber."
    ::= { ifEntry 1 }

ifDescr OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "information about the interface. [...]"
    ::= { ifEntry 2 }

ifMtu OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The size of the largest datagram which can be sent/received on the interface"
```

```

 ::= { ifEntry 4 }

ifInOctets OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION "The total number of octets received on the interface"
    ::= { ifEntry 10 }

```

### 3 Extrait du sous-arbre ip de la MIB-2

```

ipNetToMediaTable OBJECT-TYPE
    SYNTAX SEQUENCE OF IpNetToMediaEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION "[...] used for mapping from IP addresses to physical addresses."
    ::= { ip 22 }

```

```

ipNetToMediaEntry OBJECT-TYPE
    SYNTAX IpNetToMediaEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION "Each entry contains one IpAddress to 'physical' address equivalence."
    INDEX { ipNetToMediaIfIndex,
            ipNetToMediaNetAddress }
    ::= { ipNetToMediaTable 1 }

```

```

IpNetToMediaEntry ::=
    SEQUENCE {
        ipNetToMediaIfIndex INTEGER,
        ipNetToMediaPhysAddress PhysAddress,
        ipNetToMediaNetAddress IpAddress,
        ipNetToMediaType INTEGER
    }

```

```

ipNetToMediaIfIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "[...] the same as identified by ifIndex."
    ::= { ipNetToMediaEntry 1 }

```

```

ipNetToMediaPhysAddress OBJECT-TYPE
    SYNTAX PhysAddress
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "The media-dependent 'physical' address."
    ::= { ipNetToMediaEntry 2 }

```

```

ipNetToMediaNetAddress OBJECT-TYPE
    SYNTAX IpAddress
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "The IpAddress corresponding to the media-dependent 'physical' address."
    ::= { ipNetToMediaEntry 3 }

```

```

ipNetToMediaType OBJECT-TYPE
    SYNTAX INTEGER {
        other(1), invalid(2), dynamic(3), static(4)
    }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION "The type of mapping."
    ::= { ipNetToMediaEntry 4 }

```