

NOM Prénom No Étudiant :

Université Paris Diderot

Année 2015–2016

Programmation Réseaux - Contrôle Continu 1

Durée : 45 minutes
Aucun document n'est autorisé

1. Donnez une commande unix pour connaître le nom de la machine sur laquelle on se trouve

Réponse :

2. Donnez une commande pour connaître l'adresse IP de la machine `www.paris.fr`

Réponse :

3. Sur combien d'octets sont codées les adresses IPv6 ?

Réponse :

4. Quel est le service associé classiquement au port 7 sous les systèmes unix ?

Réponse :

5. Quelle commande lancer depuis le terminal pour se connecter à un service sur le port 4343 actif sur la machine `monjetas.informatique.univ-paris-diderot.fr` sous un système Unix (au sein des machines de l'ufr) ?

Réponse :

6. Quel est le protocole de la couche transport permettant de communiquer en mode par paquets ?

Réponse :

7. Quel est le problème principal du morceau de code suivant :

```
Socket so=new Socket(''lucien'');
PrintWriter pw=new PrintWriter(
    new OutputStreamWriter(so.getOutputStream()));
pw.print(''Hello\n'');
```

Réponse :

8. Quel est le problème du morceau de code suivant :

```
ServerSocket serv=new ServerSocket(7676);
InputStream is=serv.getInputStream();
```

Réponse :

9. Qu'affiche le programme suivant ?

```
public static void main(String [] args){
    try {
        Process process = Runtime.getRuntime().exec("ls");
        process.waitFor();
    }
    catch (Exception e) {
        e.printStackTrace();
        System.exit(-1);
    }
}
```

Réponse :

10. Qu'affiche le code suivant ?

```
public class ServiceTest implements Runnable {
    String st;
    public ServiceTest(String _st){
        this.st = _st;
    }
    public void run(){
        System.out.println("Hello_" + st);
    }
}
```

```

    }
}

public class Prog {
    public static void main(String [] args){
        try{
            for(int i=15;i>=10;i--){
                String s=Integer.toString(i);
                Thread t=new Thread(new ServiceTest(s));
                t.start();
            }
        }
        catch(Exception e){
            System.out.println(e);
            e.printStackTrace();
        }
    }
}

```

Réponse :

On considère le code suivant :

```

public class Serveur{
    public static void main(String [] args){
        try{
            ServerSocket server=new ServerSocket(6565);
            while(true){
                Socket socket=server.accept();
                PrintWriter pw=new PrintWriter(
                    new OutputStreamWriter(
                        socket.getOutputStream()));
                pw.println("Bonjour!");
                pw.close();
                socket.close();
            }
        }
        catch(Exception e){
            System.out.println(e);
            e.printStackTrace();
        }
    }
}

public class Client{

```

```

public static void main(String [] args){
    try{
        Socket socket=new Socket("monjetas",6565);
        BufferedReader br=new BufferedReader(
            new InputStreamReader(
                socket.getInputStream()));
        String mess=br.readLine();
        PrintWriter pw=new PrintWriter(
            new OutputStreamWriter(
                socket.getOutputStream()));
        pw.println("Salut!");
        System.out.println("Recu_du_serveur_:"+mess);
        pw.close();
        br.close();
        socket.close();
    }
    catch(Exception e){
        System.out.println(e);
        e.printStackTrace();
    }
}
}

```

11. Sur quelle machine doit-on exécuter le programme de la classe Serv pour la faire marcher avec notre client ?

Réponse :

12. Comment testeriez-vous le serveur sans lancer le programme du client ? (Détaillez votre réponse)

Réponse :

13. Qu'affichent le client et le serveur lorsqu'on les exécute ensemble (en mettant le serveur sur la bonne machine) ?

Réponse :

14. Modifiez le serveur pour lui faire afficher un message textuel envoyé par le client.

Réponse :