

- b. rpm -ivh packagename
- c. rpm -qi packagename
- d. rpm -e packagename

Câu 25: Trong Ubuntu, công cụ tương tự với gói cài đặt tự động yum trên CentOS là

- a. sudo
- b. su
- c. get
- d. apt-get

Câu 26: Để cấu hình địa chỉ IP cho một card mạng được đặt tên là ifcfg-eth0, ta dùng lệnh:

- a. nano /etc/sysconfig/ifcfg-eth0
- b. nano /etc/network-scripts/ifcfg-eth0
- c. nano /etc/sysconfig/network-scripts/ifcfg-eth0
- d. nano /etc/network-scripts/sysconfig/ifcfg-eth0

Câu 27: Lệnh nào cho phép khởi động lại dịch vụ mạng?

- a. restart service network
- b. network restart
- c. network service restart
- d. service network restart

Câu 28: Những Package được đóng gói, khi đặt tên sẽ có cấu trúc:

- a. Tên package-số hiệuphiên bản.kiến trúc.rpm
- b. Tên package-số hiệu.kiến trúc.rpm
- c. Tên package-phiên bản.kiến trúc.rpm
- d. Tên package-phiên bản-số hiệu.kiến trúc.rpm

Câu 29: Tiến trình cha của tất cả các tiến trình khác trên hệ thống Linux có tên?

- a. pm
- b. kthreadd
- c. bash
- d. init

Câu 30: Khi thực hiện đưa một tiến trình từ tiền cảnh vào hậu cảnh, ta thêm vào cuối câu lệnh dấu:

- a. *
- b. &
- c. \$
- d. @

Câu 31: Trong các trạng thái của tiến trình, trạng thái zombie có nghĩa là:

- a. Tiến trình dừng, nhưng chưa kết thúc hẳn vì còn chờ phản hồi của tiến trình cha.
- b. Tiến trình đang chờ đợi tài nguyên
- c. Sự thực thi của tiến trình kết thúc
- d. Tiến trình có trong bộ nhớ, nhưng không làm gì cả.

Câu 32: Khi thực hiện lập lịch tại một thời điểm định trước, ta dùng cú pháp:

- a. crontab
- b. batch
- c. at
- d. Các câu trên đều đúng.

Câu 33: Khi thực hiện gọi các tham số vào shell script, ta dùng \$#, ý nghĩa của \$# là:

- a. Danh sách tham số đầy đủ
- b. Tên tập tin lệnh
- c. Tổng số tham số
- d. Các giá trị của tham số

Câu 34: Run level nào cho phép khởi động vào chế độ đa người dùng, có dịch vụ mạng và không có giao diện người dùng?

- a. 6
- b. 5
- c. 3
- d. 0

Câu 35: Hãy cho biết kết quả của câu lệnh: echo hihihih | sed 's/hi/Hi/2'

- a. hihihih
- b. HiHiHiHi
- c. hiHiHihi
- d. hihihih

Câu 36: cho biết kết quả xuất ra của lệnh sau x=10 && [\$x -gt 0] && echo "Hello" || echo "World" || echo "Hello World"

- a. Hello
- b. World
- c. Hello World
- d. Hello và Hello World

Câu 37: Trong câu lệnh awk, ký tự FS là ký tự:

- a. Phân cách cột
- b. Phân cách dòng
- c. Tổng số dòng có trong file

d. Các câu trên không đúng

Câu 38: Lệnh nào cho phép đăng nhập vào một máy tính ở xa với kết nối được mã hóa?

- a. Telnet
- b. Route
- c. SSH
- d. Netstat

Câu 39: Trong các tiến trình, giá trị nice nào sau đây thiết lập tiến trình mang độ ưu tiên cao nhất:

- a. 0

Phần tự luận (6 đ)

Câu 1 (2 đ): Viết 1 shell script, cho phép người dùng nhập vào một số nguyên dương. Kiểm tra số người dùng vừa nhập có chính xác là số nguyên dương không. Nếu đúng xuất ra số ngũ phân của số nguyên dương này. Nếu không đúng, yêu cầu nhập lại cho đến khi đúng thì thôi.

Vì dụ:

```
[gv@localhost ~]$ bash DoiNhiPhan.sh
input: -10
Bạn đã nhập sai, đề nghị nhập lại.
input: 128
output: 10000000
```

Câu 2 (4 đ): Cho file quản lý dữ liệu DNS có tên data.txt, có cấu trúc như hình bên dưới, mỗi dòng gồm có 2 thông tin (địa chỉ IP và domain tương ứng), phân cách nhau bởi 1 khoảng trắng.

```
210.18.4.5 abc.net
31.9.0.120 xyz.com
110.20.1.65 a1.org
21.38.100.4 bha.us
```

- a. Hãy viết hàm nhập vào 1 địa chỉ IP, kiểm tra xem địa chỉ IP này có tồn tại trong file dữ liệu hay chưa, nếu tồn tại trả về số 1, không tồn tại trả về số 0 (hàm này sinh viên tự đặt tên) (1.5 đ).
- Trên chương trình chính cho chạy hàm vừa nhập.
- b. Nếu địa chỉ IP này chưa tồn tại, hãy cho nhập vào tên miền và cập nhật vào file dữ liệu trên. Xuất toàn bộ file dữ liệu sau khi cập nhật (1.5 đ).
- c. Nếu địa chỉ IP đã tồn tại, hãy cho biết tên miền tương ứng với IP đó. (1 đ).

Khi thực hiện chạy scripts, người dùng sẽ gọi:

```
[gv@localhost ~]$ bash DNS.sh data.txt
```

Lưu ý: SV chỉ thực hiện các yêu cầu trên, không cần kiểm tra các dữ liệu đầu vào nếu đề thi không yêu cầu

-HẾT-

- b. 19
- c. -20
- d. -30

Câu 40: Câu lệnh sau đây có giá trị là bao nhiêu:

```
n=$((RANDOM
echo $((n%100+10))
```

- a. Xuất giá trị ngẫu nhiên từ 10 - 100
- b. Xuất giá trị ngẫu nhiên từ 10 - 109
- c. Xuất giá trị ngẫu nhiên từ 10 - 110
- d. Các câu trên không đúng.

*read -p "Nhap" num
210.18.4.5 abc.net
31.9.0.120 xyz.com
110.20.1.65 a1.org
21.38.100.4 bha.us
awk '{print \$2}'*

Phân loại:

Câu 1 (2 đ): Viết 1 shell script, cho phép người dùng nhập vào một số nguyên dương. Kiểm tra số người dùng vừa nhập có chính xác là số nguyên dương không. Nếu đúng xuất ra số nhị phân của số nguyên dương này. Nếu không đúng, yêu cầu nhập lại cho đến khi đúng thì thôi.

Ví dụ:

```
[gv@localhost ~]$ bash DoiNhiPhan.sh
```

```
input: -10
```

Bạn đã nhập sai, đề nghị nhập lại.

```
input: 128
```

```
output: 10000000
```

Câu 2 (1 đ): Cho file quản lý dữ liệu DNS có tên data.txt, có cấu trúc như hình bên dưới.

output: 10000000

Câu 2 (4 đ): Cho file quản lý dữ liệu DNS có tên data.txt, có cấu trúc như hình bên dưới, mỗi dòng gồm có 2 thông tin (địa chỉ IP và domain tương ứng), phân cách nhau bởi 1 khoảng trắng.

210.18.4.5 abc.net

31.9.0.120 xyz.com

110.20.1.65 a1.org

21.38.100.4 bha.us

- a. Hãy viết hàm nhập vào 1 địa chỉ IP, kiểm tra xem địa chỉ IP này có tồn tại trong file dữ liệu hay chưa, nếu tồn tại trả về số 1, không tồn tại trả về số 0 (hàm này sinh viên tự đặt tên) (1.5 đ).

Trên chương trình chính cho chạy hàm vừa nhập.

- b. Nếu địa chỉ IP này chưa tồn tại, hãy cho nhập vào tên miền và cập nhật vào file dữ liệu trên. Xuất toàn bộ file dữ liệu sau khi cập nhật (1.5 đ).

- c. Nếu địa chỉ IP đã tồn tại, hãy cho biết tên miền tương ứng với IP đó. (1 đ).

Khi thực hiện chạy scripts, người dùng sẽ gọi:

[gv@localhost ~]\$ bash DNS.sh data.txt

Lưu ý: SV chỉ thực hiện các yêu cầu trên, không cần kiểm tra các dữ liệu đầu vào nếu đề thi không yêu cầu.

—HẾT—