**1 Minix 3**

* Install : pkgin install [software]

Remove: rm [file]

Update: pkgin update

* Ifconfig is used to view and change the configuration fo the network interfaces of the system.

Adduser is used to create a new user.

Password is used to change the password of users; only root can do it.

**2 Working on a remoter server**

* The default SSH port is 22.
* List and explain the role of each file in the $HOME/.ssh directory:

1. config: stores/sets the configuration of the host
2. id\_rsa: it represents the private key
3. id\_rsa.pub: it represents the public key corresponding to id\_rsa
4. known\_hosts: it contains host keys of SSH servers accessed by the user
5. authorized\_keys: it contains SSH keys that is authorized by the server

* How key –only authentication works in SSH:

There is a pair of keys – private key and public key. Private key is held only by the user, while the public key is copied to the SSH server. The the user can get access to the server by showing its private key.

**3 Basic Bash scripting**

• What should be the first line of a Bash script?

#!/bin/bash

• What are the main differences between sh, bash, csh, and zsh?

Both sh and bash are bourne-compatible shells. Sh is the original Bourne shell without POSIX features. Bash is the GNU Bourne Again shell, which is POSIX compatible and has some extensions. Csh is C shell, which is similar to the C programming language sometimes, which is not Bourne-compatible. Zsh is Z shell, which has some extra features and some Bourne-compatible behaviors.

• How to define and access variables?

Define: variable = value

Access: $variable

• What is the meaning of $0, $1,…, $?, $!?

$0: ./script.sh, i.e. the name of the file

$1: first argument to the script

$?: exit code from the previous command

$!: last process number

• How to define arrays and access or assign elements?

Define: declare –a arrayname

Assign: arrayname[index] = value

• How to perform if and switch statements? Provide an example.

if [ $a = “a” ] ; then

echo aaa

fi

case $i in

1. echo a

;;

1. echo b

;;

\*) echo c

esac

• What are the various syntaxes of a for loop? For each type write a sample code.

for ((i=0; i<2; i++)) ; do

echo $i

done

for i in seq 1 9 ; do

echo $i

done

array = ( “a” “b” “c” )

for i in “$[arr[@]]”

do

echo $i

done

line = “a b c”

for i in $line

do

echo “$i”

done

• How to write a while loop?

while [ expression ] ; do

statements;

done

• What is the use of the PS3 variable? Provide a short code example.

PS3 variable is used to design a custom prompt for the select loop inside a shell script.

Code:

PS3 = “Select a course (1-4): ”

select i in ve482 ve370 ve281 ve311

do

case $i in

ve482) echo “ve482 selected”

ve370) echo “ve370 selected”

ve281) echo “ve281 selected”

ve311) echo “ve311 selected”

esac

done

• What is the purpose of the iconv command, and why is it useful?

iconv is used to convert text from one form of encoding to another.

Computer stores text as combination of 0 and 1 and character encoding informs the computer how to interpret these 0 and 1 to characters we know. Using iconv we can choose the right way to decode.

• Given a variable $temp what is the effect of ${#temp}, ${temp%%word}, ${temp/pattern/string}.

${#temp}: the length of $temp

${temp%%word}: if word occur at the end of $temp, ${temp%%word} is $temp removing word; otherwise ${temp%%word} is $temp

${temp/pattern/string}: the first [pattern] occur in $temp is replaced by [string]; if [pattern] does not occur in $temp, ${temp/pattern/string} is ${temp}

• Search what are “regular expressions” and how to use them in a grep or find command. Give some simple examples based on files and keywords used in exercise 2 of assignment 2.

A regular expression is a special text string for describing a search pattern. Regular expressions are used in grep or find command to support various kinds of search pattern. For example, *grep “it$” a.txt* finds all the lines in a.txt ending with “it”.

Example: find / “.\*keymap.\*”

grep /usr/src –regex “shift.\*F7”

* Sed is a stream editor. It can filter and transform text from a file or input from a pipeline on a line-by-line basis.

Awk is used to match certain pattern of the text.

For example, *echo “awtermelon” | sed s/aw/wa* will substitute aw with wa in the input from the pipeline “awtermelon”. *awk ‘/root/’ a.txt* prints all the lines containing toot in a.txt.

Sed is useful for character-based processing, while awk is useful for delimited field processing.