1. What is a job?  
a) group of tasks  
b) group of commands  
c) group of processes  
d) group of signals

Answer: c  
Explanation: A job is a name given to a group of processes. The easiest way of creating a job is to run a pipeline of two or more commands. If we are using C, Korn or Bash shell we can use the job control commands to manipulate jobs.

2. Which of the following command is used to suspend a job?  
a) ctrl-Z  
b) ctrl-Q  
c) bg  
d) $

Answer: a  
Explanation: Suppose we invoke a command and the prompt hasn’t returned even after a long time then we can suspend that job by pressing Ctrl-Z. The point needed to be focused here is, the job hasn’t terminated yet; it’s only suspended or stopped.

3. Which command will push the current foreground job to the background?  
a) bg  
b) fg  
c) ctrl-Z  
d) kill

Answer: a  
Explanation: If we’d suspended a job using ctrl-Z then after that we can use the bg command to push the current foreground job to the background. For example,

$ bg

[1] sort abd.index > mash.index & // this job has been sent to the background. [1]

// indicates job number 1.

4. \_\_\_\_ command will bring the background jobs to the foreground.  
a) bg  
b) fg  
c) ctrl-Z  
d) kill

Answer: b  
Explanation: We can use the fg command to bring any of the background jobs to the foreground. To bring the most recent background job to the foreground we can use the following command:

$ fg

5. The command fg %1 will bring the first background job to the foreground.  
a) True  
b) False

Answer: a  
Explanation: fg command can be used with the job number, job name or a string as arguments prefixed with the % symbol. For example,

$ fg %3 // brings the third job to foreground

6. The command bg %2 is valid.  
a) True  
b) False

Answer: a  
Explanation: bg command can also be used with the job number, job name or a string as arguments prefixed with the % symbol. For example, if we’ve suspended two jobs but now we want to send the second job to the background, use the following command:

$ bg %2

7. What does the following command do?

$ kill %2

a) kills job number 2  
b) kills the second background job  
c) invalid command  
d) kill all foreground & background jobs

Answer: b  
Explanation: We can use the identifiers like job number, job name or a string of arguments with kill command to terminate a job. Thus kill %2 will kill the second background job.

10. Which command is used to list the status of jobs?  
a) fg  
b) JOBS  
c) jobs  
d) fg

Answer: c  
Explanation: We can use the jobs command to list the status of the jobs. This command tells the state of the job along with job number. For example,

$ jobs

[3] Running wc -l fitr?? > word\_count &

[2] Running sort abd.index > mash.index &

11. We can schedule a job to run at a specified time of day using \_\_\_\_\_\_\_ command.  
a) batch  
b) at  
c) cron  
d) jobs

Answer: b  
Explanation: UNIX provides sophisticated facilities to schedule a job to run at a specified time of day using at command. at command takes as its arguments the time the job is to be executed and displays the > prompt. The input then has to be supplied from standard input. For example,

$ at 19:07

at> file02.sh

[Ctrl-D]

Commands will be executed using usr/bin/bash

Job 1016171818.a at Sun Jan 15 19:07:00 2018

In the above command the script file named file02.sh goes to the queue and it will be executed at the specified time.

3. We cannot find out the name of the program scheduled using at command.  
a) True  
b) False

Answer: a  
Explanation: Unfortunately, the major drawback in at command is that we cannot find out the name of the program scheduled to be executed. This may create a problem when we are unable to recall whether a specific job has actually been scheduled for later execution.

4. Which command permits to schedule jobs for later execution, as soon as the system load permits?  
a) at  
b) %  
c) batch  
d) cron

Answer: c  
Explanation: The batch command also allows us to schedule jobs for later execution but unlike at command, it executes the jobs as soon as the system load permits. For example,

$ batch < emp.sh

7. What is a daemon?  
a) process whose parent has died  
b) process who has completed its execution but still has an entry in the process table  
c) process which is running infinitely  
d) process which runs automatically without any user interaction

Answer: d  
Explanation: Daemons are processes which run automatically without requiring any user interaction. These are designed to help the user by performing tasks which are commonly done. For example, checking for an e-mail.

8. What is cron?  
a) a simple process  
b) an orphan process  
c) a daemon  
d) a zombie process

Answer: c  
Explanation: cron is a daemon which runs on UNIX systems. It executes programs at regular intervals. It lets us to schedule jobs so that they can be scheduled repeatedly. It executes cron jobs created by the crontab. For example, a cron job can be scheduled for processing logs every evening.

11. To find out how efficiently a program a used the system resources, which command is used?  
a) sys  
b) time  
c) crontab  
d) daemon

Answer: b  
Explanation: When there are multiple versions of a program and we want to find out how efficiently they’ve used the system resources we can use the time command. The time command accepts the entire command line to be timed as its argument. It executes the command and also displays the time usage on the terminal. For example,

$ time sort -o emp.txt invoi.txt

real 0m19.811s // time elapsed from the invocation of command until its termination

user 0m1.851s // time spent in executing itself

sys 0m9.761s // time spent by the kernel in doing work on behalf of the user process