Chapter 23 Answers to In-Chapter Exercises

23.1 No answer required, but the utility of this “dry run” listing is mainly that if you intend to deviate from the installation instructions given, you know ahead of time where and possibly how you will proceed when you reach the point of deviation.

23.2 If you click Next and press Enter, all partitions and data currently on that single disk will be wiped clean. That is assuming that you follow the 4 steps explicitly as shown, and accepted all of the defaults in those steps.

23.3 They all transmit passwords in plain text by default, unless you tunnel them through ssh or change the defaults.

23.4 No answer required. You can use the **more /etc/groups** command to find out the names of groups that already exist on the system.

23.5 No answer required.

23.6 No answer required, but the design and creation of groups and user configurations that you do ahead of time directly affects what you are able to do with **adduser –f filename.** This depends upon your system management strategy and the complexity of your user base.

23.7 No answer required, but we noticed that launching the User Manager from the command line as superuser using **pc-usermanger** is more reliable.

23.8 No answer required.

23.9 For PC-BSD it is listed in /dev as /dev/da0s1.
For Solaris it is listed as /dev/rdsk/c9t0d0p1.

23.10
1. In the following solution we partition, format to zfs, and add the new hard disk as a bootable system disk in preparation for using it as a mirror of the already existing system disk.
a. Become the superuser.
[bob@pcbsd-5867] ~# **su**
Password: **xxx**
[bob@pcbsd-5867] ~#

b. Having previously determined the logical device name to the new hard disk as ada1, you should delete the ZFS metadata associated with all partitions that may have previously existed on it. This also clears the partitions on ada1.
[bob@pcbsd-5867] ~# **zpool labelclear –f ada1**

In the next steps, we will prepare partitions on the new hard disk to accommodate the mirroring process.

c. You then have to add a boot partition to the new hard disk, using the **gpart** command, so that you will be able to boot to it in the event to a failure of the original system disk.
[bob@pcbsd-5867] ~# **gpart add -b 40 -l gptboot -s 512k -t freebsd-boot ada1**

ada1p1 added

d. You then have to add a swap space partition to to the new hard disk.
[bob@pcbsd-5867] ~# **gpart add -s 1G -l swap1 -t freebsd-swap ada1**
ada1p2 added

e. You then have to add the primary partition which will contain the tank1 zpool and all file systems attached to it.
[bob@pcbsd-5867] ~# **gpart add -t freebsd-zfs -l zfs1 ada1**
ada1p3 added

f. You then have to put the bootloader code on the boot partition you created in step c. above.
[bob@pcbsd-5867] ~# **gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1**
bootcode written to ada1

g. You can then examine the partition scheme for all disks attached to your system.
[bob@pcbsd-5867] ~# **gpart show**

```
=>         63  156301425  ada0  MBR  (75G)
           63          63        - free -  (32K)
          126   156301299     1  freebsd  [active]  (75G)
    156301425          63        - free -  (32K)

=>          0   156301299  ada0s1  BSD  (75G)
            0   152086528       1  freebsd-zfs  (73G)
    152086528     4194304       2  freebsd-swap  (2.0G)
    156280832       20467         - free -  (10M)

=>         34   156249933  ada1  GPT  (75G)
           34           6        - free -  (3.0K)
           40        1024     1  freebsd-boot  (512K)
         1064     2097152     2  freebsd-swap  (1.0G)
      2098216   154151751     3  freebsd-zfs  (74G)

=>         34   156249933  diskid/DISK-5RW32T4J  GPT  (75G)
           34           6                        - free -  (3.0K)
           40        1024                     1  freebsd-boot  (512K)
         1064     2097152                     2  freebsd-swap  (1.0G)
      2098216   154151751                     3  freebsd-zfs  (74G)
```

[bob@pcbsd-5867] ~#

2. No answer required.

3. Assuming the new external USB drive is da0, and has no partitions on it-

root@pcbsd-4382:~ # **gpart destroy -F da0**
da0 destroyed

root@pcbsd-4382:~ # **gpart create -s GPT da0**
da0 created

root@pcbsd-4382:~ # **gpart add -t freebsd-zfs -l zfs1 da0**
da0p1 added

root@pcbsd-4382:~ # **gpart show**
=> 34 117231341 ada0 GPT (56G)
34 2048 1 bios-boot (1.0M)
2082 6 - free - (3.0K)
2088 113014784 2 freebsd-zfs (54G)
113016872 4194304 3 freebsd-swap (2.0G)
117211176 20199 - free - (9.9M)

=> 34 312581741 da0 GPT (149G)
34 312581741 1 freebsd-zfs (149G)

root@pcbsd-4382:~ # **zpool create test7 /dev/da0p1**
root@pcbsd-4382:~ # **zpool list**
Output truncated…
test7 55K 144G 19K /test7

root@pcbsd-4382:~ #

23.11 No answer required.

23.12 No answer required.

23.13 Assuming the name of the directory of your choosing is labs-
% cd labs
% tar cvf ~/labs.tar .
[command output]
%
labs.tar

23.14 % cd ~
% tar -tvf labs.tar
[command output]
%

23.15 % mkdir ~/dir.backup
% cp ~/labs.tar ~/dir.backup/mytar
% cd ~/dir.backup
% tar xvf mytar
[command output]
%

23.16 Open a terminal, become superuser, and type-

crontab –e
Select an editor, and type the following-

```
# m  h  dom  mon  dow   command
0     3     *      *     1     sh            /usr/local/bin/backup.sh
```

Save the crontab file. The backup.sh script will now be executed at 3:00 AM every week. Here, /usr/local/bin/backup.sh is the path of backup.sh file. You can change the script path to suit your system requirements. The -e option enabled you to edit the user's
crontab.

23.17 No answer required.

23.18 No answer required.