A) Log files.

Type wrong password in login and try to find info about that in log files. Check if there is more info about startup. Try with head- and tail command. Use Google to find out what is registered in different log files.

:~$ cd /var/log
:/var/log$ sudo less syslog - system-related info and general messager
:/var/log$ sudo dmesg | less - info about drivers
:/var/log$ sudo tail syslog - to see last 10 log messages statically
:/var/log$ sudo tail -f syslog - to see last 10 log messages dynamically, new files will be added consecutive. To end that process, press Ctrl^c

:/var/log$ sudo less auth.log - here comes info about login / and authentication log

:/var/log$ sudo head faillog | less - info about unsuccessful login
:/var/log$ sudo less daemon.log - info about processes running at background
:/var/log$ sudo head daemon.log - info about first 10 processes started at background

B) Cron-jobs.

A cron task runs a given command at precise time, for example each night at 2 am.
In our case we run command each minute to skip waiting. Cron jobs can be set up with command crontab -e.

:~$ crontab -e

You opened an editor where you can add so called cron line which defines your cron task. A such line for example can be:

***/1 * * * * echo hello there > /dev/pts/0 - prints “hello there” to a terminal each minute.**

:~$ tty
/dev/pts/0 - to determine a terminal nr.

*/1 * * * * echo “hello there” > /dev/pts/0

...adding this line at the end of crontab file...
Ctrl^o
Ctrl^x
crontab: installing new crontab

ik0v@debian:~$ hello there
ik0v@debian:~$ hello there - get this message each minute on my screen

Check log to see a status for cron jobs, /var/log/syslog with tail command. You can start a watch with seconds indicator with command
xclock -update 1 &

:~$ sudo tail -f /var/log/syslog
:~$ xclock -update 1& - this command opens a new window with watch

Try your own cron tasks, for example write today's date to a file each minute:

* * * * * echo $(date +"%d-%m-%y") > ~/Desktop/date.txt

...writing that command at the end of crontab file...

each minute write status of a chosen process to a file:

* * * * * ps auxf | grep crontab | tail >> ~/Desktop/date.txt

each minute remove all files from a chosen folder:

* * * * * rm -r ~/Desktop/deleteme/*

Now you have several cron jobs running each minute. Remember to delete them with command crontab -r**. Make sure these commands were really deleted with command crontab** -l**.**

:~$ crontab -l

:~$ crontab -r
:~$ crontab -l
no crontab for ik0v

Create a simple cron job which prints out some message on screen each 5 minutes. You can also add a date and time.

:~$ crontab -e

*/5 * * * * echo "hello from crontab, $(date)" > ~/dev/pts/0

On http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger
you can find many examples about cron.

C) Transferring files with scp and sftp.

We need some files for that exercise. Use Google and save some pictures to a folder ~/Pictures. Get about 8-10 files.
Find out whether you have commands scp and stfp firstly. Install them otherwise.

:~$ which scp
/usr/bin/scp

:~$which sftp
/usr/bin/sftp

Cooperate with somebody and send a file from your pc to that person's pc. Remember to login with your user account. Check if it worked. Try to do same operation with sftp command. Use manual or Google to find out how to do it.

:~$ cd Pictures/test
:~/Pictures/test$ touch pct1
:~/Pictures/test$ scp pct1 user@64.52.85.143:~
user@64.52.85.143's password:

pct1 100% 0 0.0KB/s 00:00

:~/Pictures/test$ touch pct1
:~/Pictures/test$ sftp pct1 user@64.52.85.143:
user@64.52.85.143's password:

sftp>ls
… here come files from user pc current folder
sftp>lls
… here come files from my folder (pct1)
sftp>put pct1
Uploading pct1 to /home/user/bilde1
sftp>get pct1 - to send a file from user's pc back to my pc.

D) tar command.

a) File prosjekt.tar.gz is attached here.
This tar file contains a files and folders.
a) Use tar-command with correct option and check files located inside this tar-file.

:~$ tar -tf prosjekt.tar.gz

prosjekt/
prosjekt/prosjektkoder.txt
prosjekt/pro02/
prosjekt/pro02/plan02.sxw
prosjekt/pro02/prospekt.sxw
prosjekt/pro02/brosjyre.pdf
... – og andre maper og filer

b) Pack out that files to a subfolder in your home directory.

:~$ gunzip prosjekt.tar.gz
:~$ ls -lh
...
prosjekt.tar - has a size of 1.9MB. When it was compressed size was 1.5MB.

:~$ mkdir test
:~$ mv prosjekt.tar test/
:~/test$ tar -xf prosjekt.tar

c) Delete file prosjekt.tar.gz and pack then all unpacked files to a new tar-file. Call this tar-file for myproject. Check that package contains all files.

:~/test$ rm prosjekt.tar

:~/test$ tar -cf myproject.tar prosjekt/
:~/test$ tar -tf myproject.tar

prosjekt/
prosjekt/prosjektkoder.txt
prosjekt/pro02/
... – and other files and folders, all files are here.

d) Pack your whole Dropbox-folder to a tar-file. Check that all files are added.

:~$ tar -cf dropbox.tar Dropbox/
:~$ tar –tf dropbox.tar

e) We spoke about tar-file, but it actually was a gz-file. Explain a connection between these.

Tar file is so called ‘box’ in which we put some files. When we are done with that we can compress that box with gzip command, so that package, box with files takes less space.