

Lab 1: Setup

12:00 PM, Sep 10, 2017

Contents

1	Your friendly lab TAs	1
2	Pair programming	1
3	Welcome to lab	2
4	The file system	2
5	Intro to terminal	3
6	What is DrRacket?	5
6.1	Setup (language levels and the CS 17 teachpack)	5
6.2	Writing expressions in DrRacket	6
6.3	Loading and saving files	7
6.4	Let's talk parentheses!	7
6.5	Comments	8
7	Handing in your work	9
8	Piazza	9
9	Feedback	10
10	Important reading	10
10.1	Collaboration policy	10
10.2	Style guide	10
11	Your partner's account	10

Objectives

By the end of this lab you will understand:

- the department file system

- rudimentary Racket constructs
- our expectations for this semester

By the end of this lab you will be able to:

- navigate the file system using the Linux terminal
- use the DrRacket programming environment
- hand in CS 17 assignments
- use Piazza
- give feedback on labs, homeworks, lectures, and more

1 Your friendly lab TAs

Welcome to your first CS 17 lab! Labs are an important component of this course—you will learn things here in lab that will help clarify lecture content and assist you in your independent work.

You may have noticed that a few of the CS 17 TAs are present for today's lab. TAs will be at all lab sessions to guide you as you need it. To ask a question, raise your hand (but remain in your seat), and a TA will be with you as quickly as possible.

2 Pair programming

Task: Read the pair programming guide available on the course website and discuss it briefly with a neighboring student. Pair programming is required on many labs (like this one) and on most projects.

During today's lab, you'll be working with this student. You'll complete your work on one person's account, and once lab is done, you can set up what you need to on the other person's account.

3 Welcome to lab

As a CS student, you have access to a number of computer labs in which to do your work. In CS17, labs will always take place in one of these rooms so that you can do your work on a department machine.

Department machines run an operating system called *Linux*. Broadly speaking, an *operating system* is the software that determines how your computer works. Examples of other operating systems are Windows and Mac. If you're used to working with operating systems like Windows or Mac, you'll find Linux a bit different. However, by the end of this lab, you'll know everything you need to know about Linux for CS 17.

Task: Login to the department machine using your Banner login and password.¹

¹You can complete this lab on your account or your partner's account.

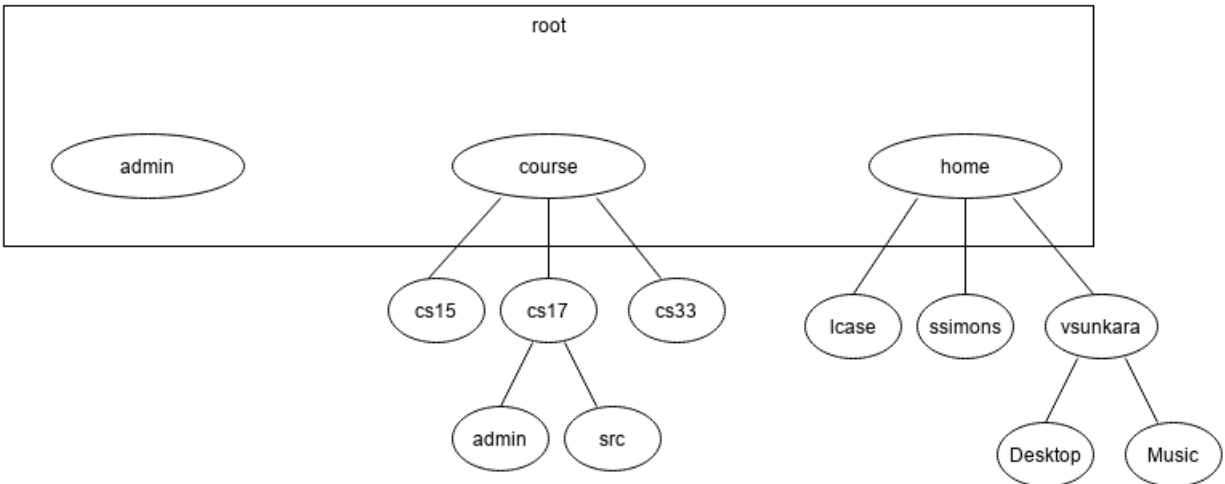


Figure 1: A visualization of the file system.

4 The file system

Once you've logged into your account, you should see your `Desktop`. On your `Desktop`, you should find the File System icon and click on it. The *File System* contains directories, or folders, that all the department machines need. One of the important directories that lives on the File System is called `home`. All CS accounts live in the `home` directory. CS accounts are actually subdirectories themselves within the `home` directory, and the name of the directory associated with your account is your login! ²

In your own directory, there are lots of subdirectories like your `Desktop`, `Documents`, and `Pictures`. You can view these directories by clicking the bar with your login on the left side of the File Manager window.

It's useful to think about the File System as a tree. The **root** of the tree is where the outermost directories, like `home`, live. Another important directory in the root of the tree is `course`. Inside of `course` is where all of the course specific directories live, like the CS 15 and CS 17 course folders.

We can talk about the location of different files and directories using *paths*. Let's say we wanted to describe where the CS 17 folder is on the file system. We would write:

```
/course/cs0170/
```

This means that to find the `cs0170` directory, we would start at the root, click on the `course` folder, and then see the `cs0170` directory. Let's say we wanted to describe where HTA Veda's `Desktop` directory is. We would write:

```
/home/vsunkara/Desktop/
```

²Don't worry, no one can access the directory associated with your account but you.

This means that to find Veda's Desktop, we would start at the root, navigate to home, click on vsunkara (Veda's login), and then see Veda's Desktop directory.

Notice that each path we've written begins with the character `/`. This means that the path you are specifying begins at the root.

Let's say you wanted to describe the location of a file **relative** to the directory you're currently in. For instance, let's say you're in the home directory and want to describe where HTA Veda's Desktop is relative to home. You would write:

```
vsunkara/Desktop/
```

Notice that we did NOT include the character, `/`, at the beginning of this path. This is because we are not starting from the root. We call these paths *relative paths*, and paths that start at the root *absolute paths*.

5 Intro to terminal

You're now going to open something called `terminal`. A *terminal* is a window that allows you to type commands for your computer to perform. Previously, you were navigating the file system graphically. You're now going to learn how to navigate the file system using terminal commands!

To get started, click on the Applications menu and select `terminal`. If you don't see `terminal`, navigate to the `accessories` submenu, and it should be there.

On the left hand side of the terminal window, `terminal` shows you where you are in the File System. You should see something like: `/gfps/main/home/<your login>`

You're now going to create a directory for all your coursework this semester using a terminal command. When you run this command, a directory with lots of subdirectories will be made in whatever directory you're currently in. This course directory needs to be inside of your personal home directory (the directory whose name is your CS login), so if your terminal doesn't display a path like the one above, call a TA over!

Running the following command will create a `course` directory within your personal home directory, and within this `course` directory, a `cs0170` directory. Additionally, inside the `cs0170` directory, subdirectories called `labs`, `homeworks`, `projects`, and `exams` will be created, and inside those subdirectories even more specific subsubdirectories.

Task: In the terminal window, type `/course/cs0170/bin/cs017_student_setup` at the prompt, and hit `Enter`. You should then see a message that says:

```
All done.
```

If you see some extra text before `All done.`, that's fine.

To see a visual display of the structure you just created, type `tree`.

You're now going to learn how to change directories into the `course` directory you just created.

First, to list the contents of your own home directory, you should type:

```
ls
```

`ls` is short for "list". The first character of this command is the letter "l" not the number one. After doing so, you should see many subdirectories like `Applications` and `Desktop`.

You should also see the directory you just created, `course`. To change directories from your home directory to the `course` directory, you should write `cd course`³

Notice that you wrote the relative, not absolute, file path to your `course` directory. This is because you were in your own home directory and therefore did not need to specify the absolute path to `course`.

You should now list the contents of your `course` directory by typing `ls`. You should then change directories into the `cs0170` directory by typing `cd cs0170`⁴. Then, list the contents of your `cs0170` directory.

Let's say you want to go back to the `course` directory. To move up into the directory that houses the directory you're currently in, you should write:

```
cd ..
```

If you type this in your `cs0170` directory, you will be brought back to the `course` directory.

If you type `cd ..` within your `course` directory, where will terminal take you? Try it out and see!

In general, using `cd ..` will bring you up one level above the directory you're currently in.

The `course` directory you've been navigating is **DISTINCT** from the `course` directory at the root of the file system. To navigate to the `course` directory at the root, you have two options:

- 1) Keep changing directories upwardly (using `cd ..`) until you arrive at the root, and then `cd` into the `course` folder.
- 2) Type `cd /course`

In option 2, notice that we typed an absolute path not a relative path. When you need to move to one directory that's pretty far from the directory you're currently in, *using an absolute path is generally preferable*.

Task: Take a look at the paths below. Determine which path leads to directories that belong to you.

path A: `/course/cs0170/src/`

path B: `/gpfs/main/home/<yourlogin>/course/cs0170`

You are now going to create and edit a text file. You are going to be editing this file in a text editor called `atom`. To open up `atom`, you should navigate to your `lab01` directory and type the following simple command: `atom`

Once `atom` has opened, you should write your favorite joke or quote. Then, you should save

³`cd` is short for change directory.

⁴Notice that you do not need the final `/` when changing directories. That is, `cd cs0170` is the same command as `cd cs0170/`.

your file by navigating to the `File` bar and pressing `Save File As`. You should save this file as `my-favorite.txt`, and importantly, you should save this file within **your** `lab01` directory.

Task: Be ready to show a TA the file you just created.



| You've reached a checkpoint! Please call over a lab TA to review your work.

6 What is DrRacket?

We're now going to shift gears and talk about the first programming language you'll be using this semester and how you'll be interacting with it. A **programming environment** is a workspace in which programmers create and run programs. The first programming language we use in CS 17 is called **Racket**, and the corresponding programming environment is called **DrRacket**.

6.1 Setup (language levels and the CS 17 teachpack)

Task: To open DrRacket, navigate to the terminal and type `drracket &`.

Note: The `&` tells the shell to open DrRacket "in the background." In other words, if you don't include `&`, you won't be able to use the terminal window until you close DrRacket.

When DrRacket is finished loading, a large window will appear on your screen. The top half of this window is called the **Definitions** window, and the bottom half is called the **Interactions** window.

When you click on the `Run` button (located at the top right), DrRacket will process everything you wrote in the definitions window, and the values it computes will appear in the interactions window. If it's a helpful analogy, you can think of DrRacket as a person who understands the Racket language, and when you press `Run`, you're telling this person to read and execute your Racket code.

You can also type expressions into the interactions window directly and then press `Enter` to evaluate them immediately!

DrRacket implements a number of "language levels," which we use to introduce Racket little-by-little. Let's start off by selecting the appropriate language level, `Beginning Student`.

Task: We're now going to set the language level.

1. Within the DrRacket window, navigate to the `Language` menu and select `Choose Language`. Open up the option for `How to Design Programs`, and select `Beginning Student`.
2. At the left corner of the `Choose Language` window, find the `Show Details` button and click on it. On the right side of the window, a number of options should appear.
 - (a) Make sure the `Case sensitive` box is checked.
 - (b) The `Insert newlines in printed values` box should also be checked.
 - (c) Make sure that `Constructor` is selected for `Output style`.
 - (d) For `Constant Style`, either option is fine.
 - (e) For `Fraction Style`, there are two possible options: `Mixed fractions` and `Repeating decimals`. You may choose whichever you prefer.

Click OK.

3. Next, you will add the CS 17 teachpack, which contains CS 17-specific code that you'll need throughout the semester. In the Language menu, choose Add Teachpack. Press the Add Teachpack to List button. Find the file:

```
/course/cs0170/src/racket/cs17.ss5
```

Once you've found the file, click Open. In the right half of the window entitled User-installed Teachpacks, select `cs17.ss` and click OK.

4. Finally, in the main DrRacket window, click Run to finish loading the teachpack.

6.2 Writing expressions in DrRacket

Task: Now you're ready to try out DrRacket! Type the following into the definitions (the top) window:

```
(+ 1 17)
```

Click on Run.

You should see the number 18 appear in the interactions window, followed by a prompt (which looks like a “greater-than” sign).

As you can see, in Racket, we write arithmetic with the operator, in this case `+`, first, and the numbers on which we want to operate afterward. This syntax is called *prefix notation*.

Wasn't that fun!? Let's try it again, but this time, in the interactions window (the bottom half of the window).

Task: Type `(+ 1 17)` into the interactions window, and then press Enter. What do you see? Same as before. Pretty cool, eh?

6.3 Loading and saving files

By design, you can save things you type in the definitions window, but you cannot save things you type in the interactions window. Thus, you'll want to use the definitions window for writing programs.

Task: Let's practice loading and saving files in the definitions window.

Add the following expressions into DrRacket's definitions window.

```
17
(+ 4 4)
(* 2 3)
(quotient 17 2)
(remainder 17 2)
```

⁵To find this file, navigate to the root of the file system, NOT your home directory. If you need help finding the file, ask a TA.

When you've finished typing, save the file by pressing the `Save Definitions As` button. Navigate to the following directory:

```
/home/⟨yourlogin⟩/course/cs0170/labs/lab01
```

Then, save the contents of the definitions window in a file named `lab01.rkt`. (Once again, nothing in the interactions window is saved, so be sure everything you want to save is entered in the definitions window!)

Now quit DrRacket by going to the `File` menu and selecting `Quit`.

You can start up DrRacket again and load a file at the same time by running the following command in your terminal:

```
drracket /path/to/your/file.rkt &
```

We're going to open the file you just created, so we'll run:

```
drracket ~/course/cs0170/labs/lab01/lab01.rkt & 6
```

The expressions you typed previously should appear in the definitions window (the upper one).

Click on the `Run` button. DrRacket will then print the values of these expressions in the interactions window.

Note: Whenever you click on the `Run` button, the interactions window is reset. This means that whatever was there before you ran the definitions window will disappear once you run the definitions window.



| You've reached a checkpoint! Please call over a lab TA to review your work.

6.4 Let's talk parentheses!

You just got your first taste of Racket syntax. Let's talk a little more about it! DrRacket relies on parentheses to evaluate the code you write accurately. A convenient way to think about Racket syntax is like this:

```
(verb noun noun)
```

As mentioned before, putting verbs before nouns in expressions is called *prefix notation*.

In Racket, you cannot use spurious parentheses because doing so changes the meaning of an expression. For instance, `(+ 1 1)` is different from `(+ (1 1))`. You'll learn why the designers of Racket made this choice in class. You're now going to take a look at some expressions. With each expression, we pair a desired value.

Task: Reformulate the expressions using Racket syntax so that DrRacket returns the desired value.⁷

- expression: `(- 3 0)`
desired value: `-3`

⁶The tilde is short for `/home/yourlogin`. If you're in your `lab01` directory, you only need to write `drracket lab01.rkt &`.

⁷**Hint:** You may need to change the order of the numbers in the expressions to achieve the desired outputs.

- expression: `(- (10 5))`
desired value: 5
- expression: `(/ 5 (+ 10 15))`
desired value: 5
- expression: `(* 2 (7 2))`
desired value: 28
- expression: `(+ (/ 3 5) (-3 (+ 2 4)))`
desired value: 3.6 or $3\frac{3}{5}$

Task: Save this file in your `lab01` directory as `my-expressions.rkt`.

6.5 Comments

When you write a program, you have to do so in a language that the computer understands. Hopefully, your code will be clear enough that you can understand it too! Even so, it's useful, as a service to your future self and to other readers of your program, to write something *about* the program in a language that *you* understand. Racket lets you do this. Any text that appears after a semicolon on a given line is a **comment**, meaning it is ignored by DrRacket during evaluation. For example, the following two Racket expressions have the same value:

```
17
17 ; this is the number 17
```

Task: Type the following into DrRacket's definitions window, then add semicolons where necessary to separate the comments from the code. ⁸

```
This is a fun but simple program.
(+ (* 2 3) It adds the product of two with three
  (* 4 5)) to the product of four with five
It should evaluate to twenty-six!
```

Task: Save your work in your `lab01` directory as `my-comments.rkt`.



| You've reached a checkpoint! Please call over a lab TA to review your work.

7 Handing in your work

To hand in assignments in CS 17, you'll need to follow a two-step process:

- 1) Navigate to the directory that contains the work you want to hand in.
- 2) Type use the following terminal command: `/course/cs0170/bin/cs017_handin <name of the directory containing what you want to hand in>`⁹

⁸You should NOT do this task in the same file as your previous task. Open up a new definitions window by clicking File and then New.

⁹Because you'll usually need to hand in multiple files at once, you'll always hand in an entire directory rather than singular files.

All the work you've completed today (i.e. `my-favorite.txt`, `lab01.rkt`, `my-expressions.rkt`, and `my-comments.rkt`) should be saved within your `lab01` directory. Therefore, to hand in your work from today, you should:

- 1) Navigate to your `lab01` directory.
- 2) Run the following command: `/course/cs0170/bin/cs017_handin lab01`

Task: Turn in the work you completed today. Only you or your partner needs to hand in your work.

Note: You can run the `handin` command as many times as you like, but we will only see the contents of your most recent `handin`.

Another note: This is the only lab you will be turning in via the command above. From now on, you'll use the `handin` command exclusively for homeworks and projects.

8 Piazza

<https://www.piazza.com/>

In CS 17, we use an online academic forum called Piazza to answer questions outside class and hours. At this point, if you're a registered student in the class, you may have received an email inviting you to join Piazza. If that hasn't happened yet, you can use the link on the announcements page of the course website to sign up.

Piazza is a super useful resource for you to make use of throughout this semester, so we want to acquaint you with its features.

Task: You are now going to ask your first Piazza question! You can ask a question via your account or your partner's account. In other words, you and your partner need not ask separate questions.

1. Navigate to the CSCI 0170 page on Piazza.
2. Click `New Post` on the left-hand side of the screen.
3. Select `Post Type: Question`
4. Select `Post to: Individual Student(s)/Instructor(s)` and then type in `Instructors` so that your question or comment is only visible to the TAs and the professor.
5. Beside `Select Folder(s)` you should select the relevant folder. In this case, that should be `lab01`.
6. Come up with a creative or silly (or serious if you're up to it!) question to ask the instructions. Come up with a summary, use the tag `lab1`, and write your question!

Task: Bookmark Piazza.

9 Feedback

At the bottom of each CS 17/ 18 document,¹⁰ there is a blurb below a horizontal bar. It contains a link to a feedback form. Please use it! It will help us to improve your experience this year and those of students in future years. We welcome your comments, ranging in content from pointing out typos to general recommendations about the class. As indicated on the form, all responses are totally **anonymous**.

If you have a technical question about a specific assignment and want to discuss it with a TA, do not use this link. Instead, email the course staff (at cs017tas@cs.brown.edu) or go to hours.

10 Important reading

Before finishing up lab, you need to read a few documents that will help you perform successfully this semester.

10.1 Collaboration policy

Task: At the beginning of this lab, the TAs distributed copies of the course collaboration policy signature page. If you haven't yet read the CS 17 collaboration policy on the course website, please do so now. Then sign the contract and return it to the lab TAs.

10.2 Style guide

The style guide for CS17 is full of conventions you need to follow so that the TAs (and you) can successfully read your work.

Task: Take a look section 2 of the CS17 style guide located on the course website.

11 Your partner's account

Before finishing today's lab, you need to set up the CS 17-specific directories on your partner's account (i.e. the account you did not use during this lab).

Task: Login into the account you haven't been using (with your Banner credentials) and open up `terminal`. Then, navigate to your personal home directory (the directory whose name is your login). Set up the CS 17-specific directions by typing `/course/cs0170/bin/cs017_student_setup` at the prompt and hitting `Enter`.

Task: On this account, visit <https://www.piazza.com> and bookmark the page.

That's all you need to do on this account.

Once you've finished, congratulations! You've completed your first lab.

Call over a TA to learn about how you receive credit for labs and to ensure you get credit for this one!

¹⁰Including this one! Take a peek!

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS17 document by filling out the anonymous feedback form: <http://cs.brown.edu/courses/cs017/feedback>.