

OLCF Crash Course in Supercomputing, Summer 2011

UNIX, VI editor, Compiling and Make, and C
Programming Language



Arnold Tharrington
Scientific Computing Group
Oak Ridge Leadership Computing Facility

Oak Ridge, Tennessee
June 16, 2011



U.S. DEPARTMENT OF
ENERGY



OAK RIDGE NATIONAL LABORATORY

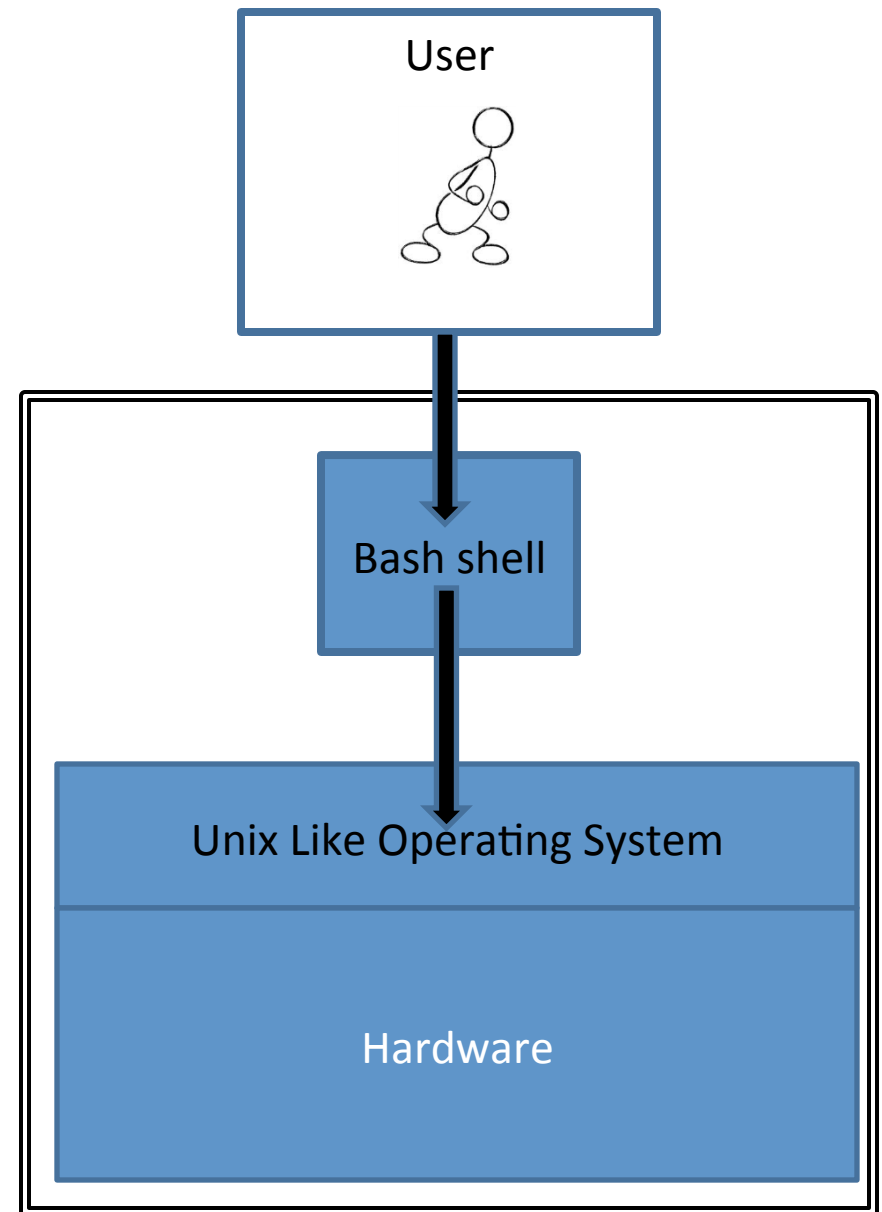
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Course Outline

- Introduction to UNIX
 - Basic navigation, manipulations of files and directories, job control, and job output redirection.
- VI editor
- C Programming
- Make

UNIX

- The operating systems of nearly all HPC clusters are derivatives of Unix.
- Bash shell
 - The user interacts with the operating system by means of a Unix shell - we will use the **Bash** shell
 - Command-line interpreter
 - Programmable
 - <http://www.gnu.org/software/bash/manual/bashref.html>

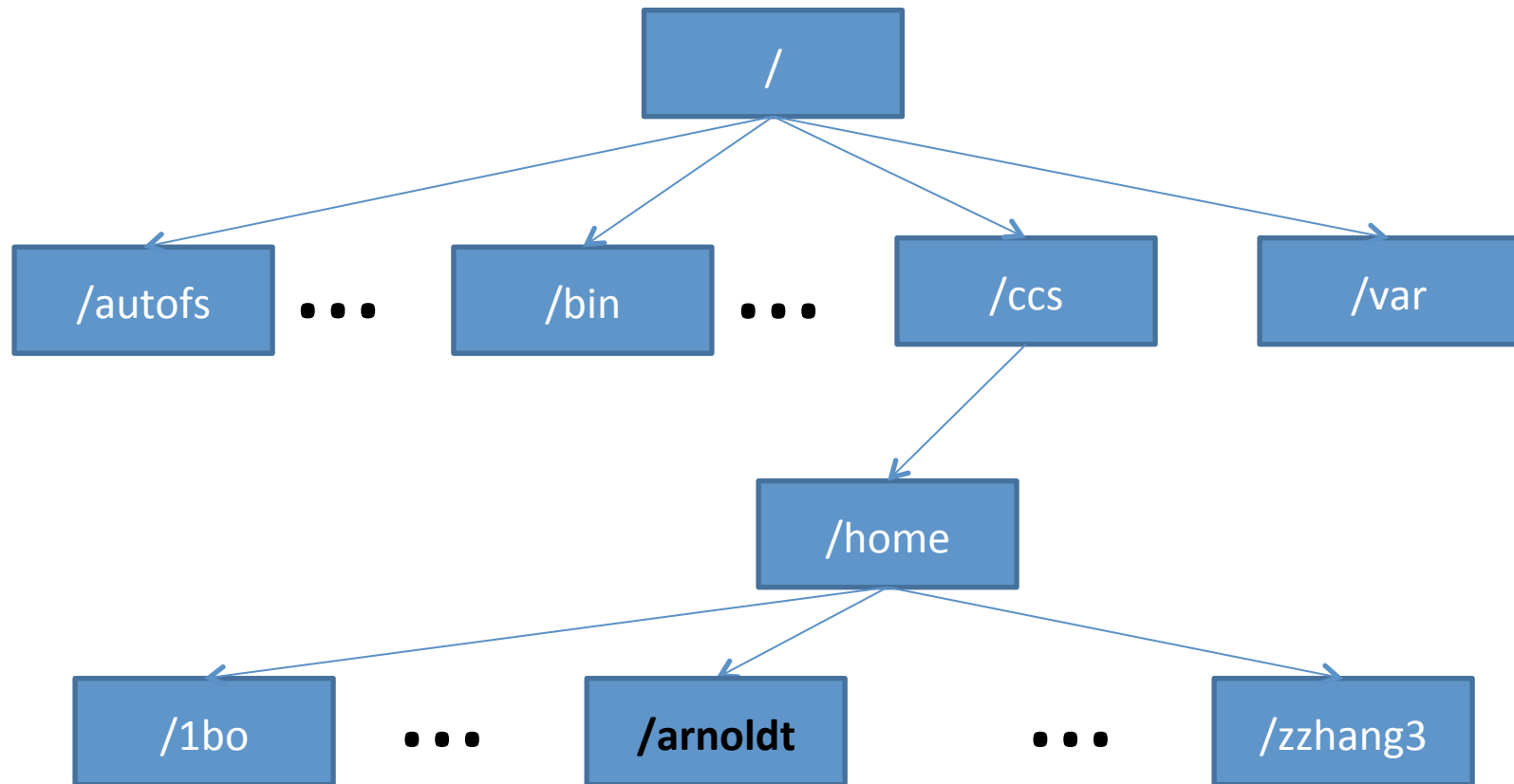


Smoky

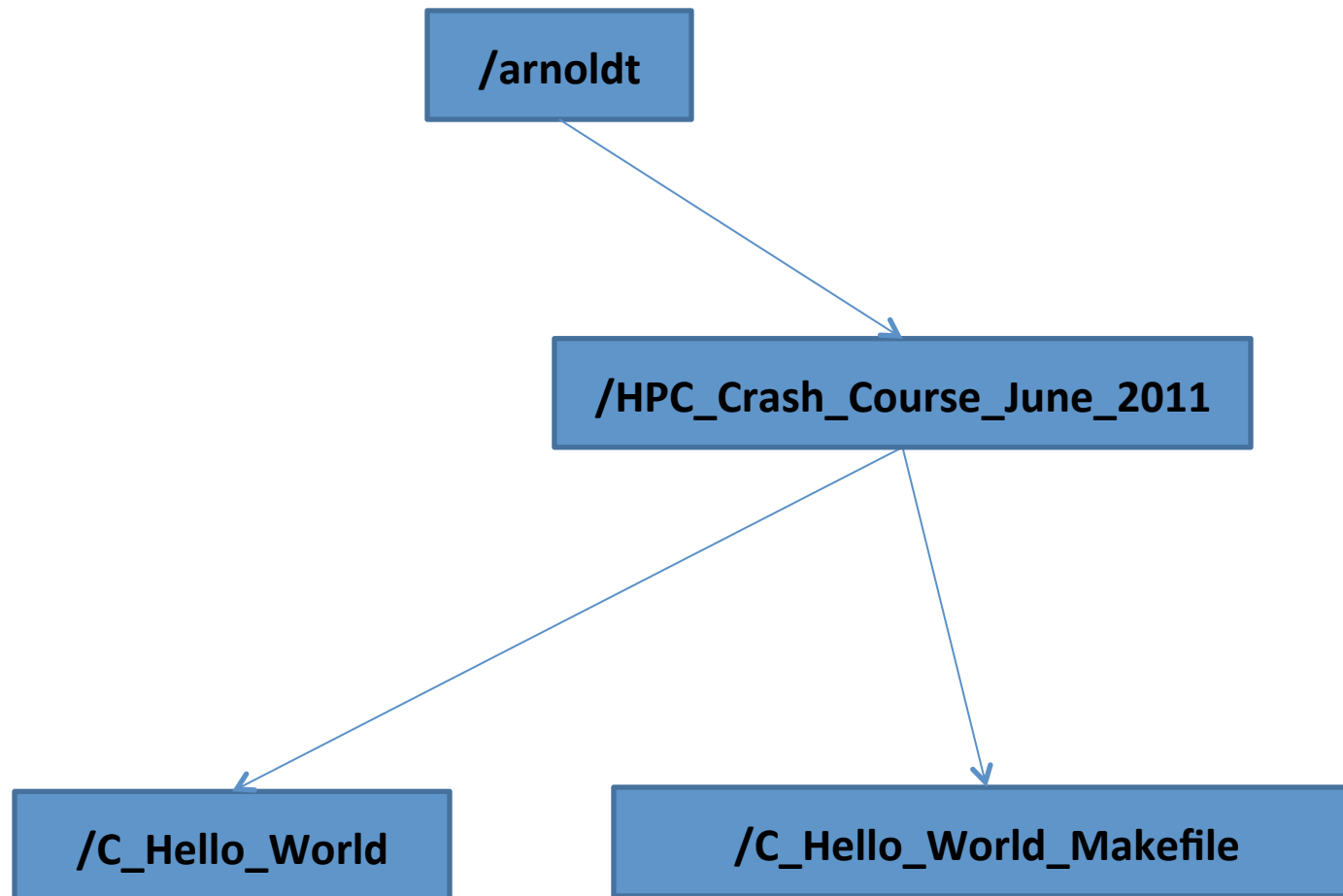
1. Basic Navigation Within The Unix File System

- Log on to Smoky
 - ssh arnoldt@smoky.ccs.ornl.gov
- Copy a directory to your home directory
 - cp -rf ~arnoldt//HPC_Crash_Course_June_2011 ./
- Run the following commands that are in bold:
 - **pwd**
 - pwd: print name of current/working directory
 - **echo \${USER}**
 - **echo \${HOME}**
 - echo - display a line of text
 - USER and HOME are environmental variables defined by the operating system

Basic Navigation Within The Unix File System (cont.)



Basic Navigation Within The Unix File System (cont.)

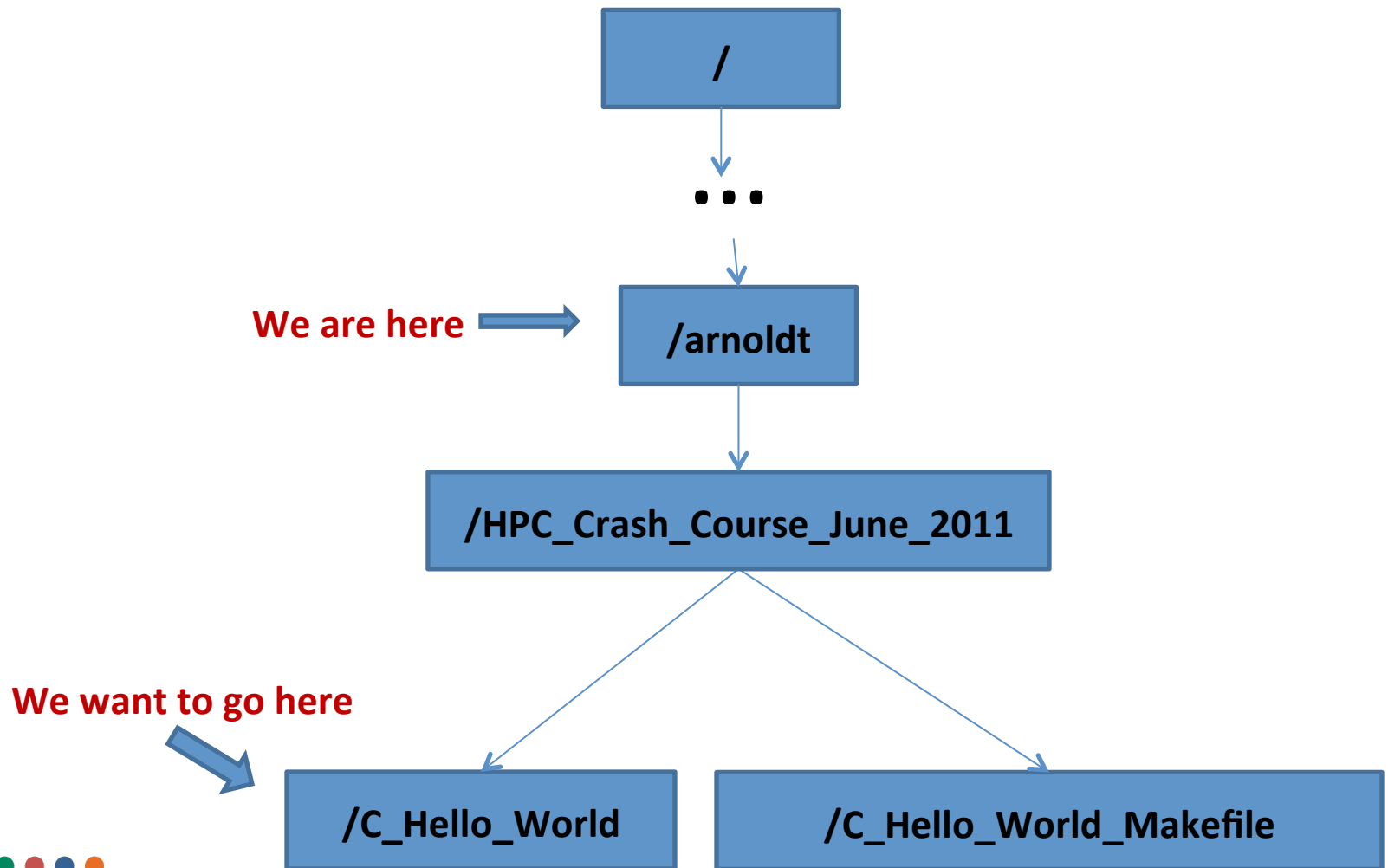


Basic Navigation Within The Unix File System (cont.)

- `cd ..`
 - Move up 1 level
- `cd ../../`
 - Move up 2 levels
- `cd`
 - Move to your home directory
- `cd -`
 - Move to the previous directory location
- `cd ${HOME}`
 - Move to directory defined by HOME

Basic Navigation Within The Unix File System (cont.)

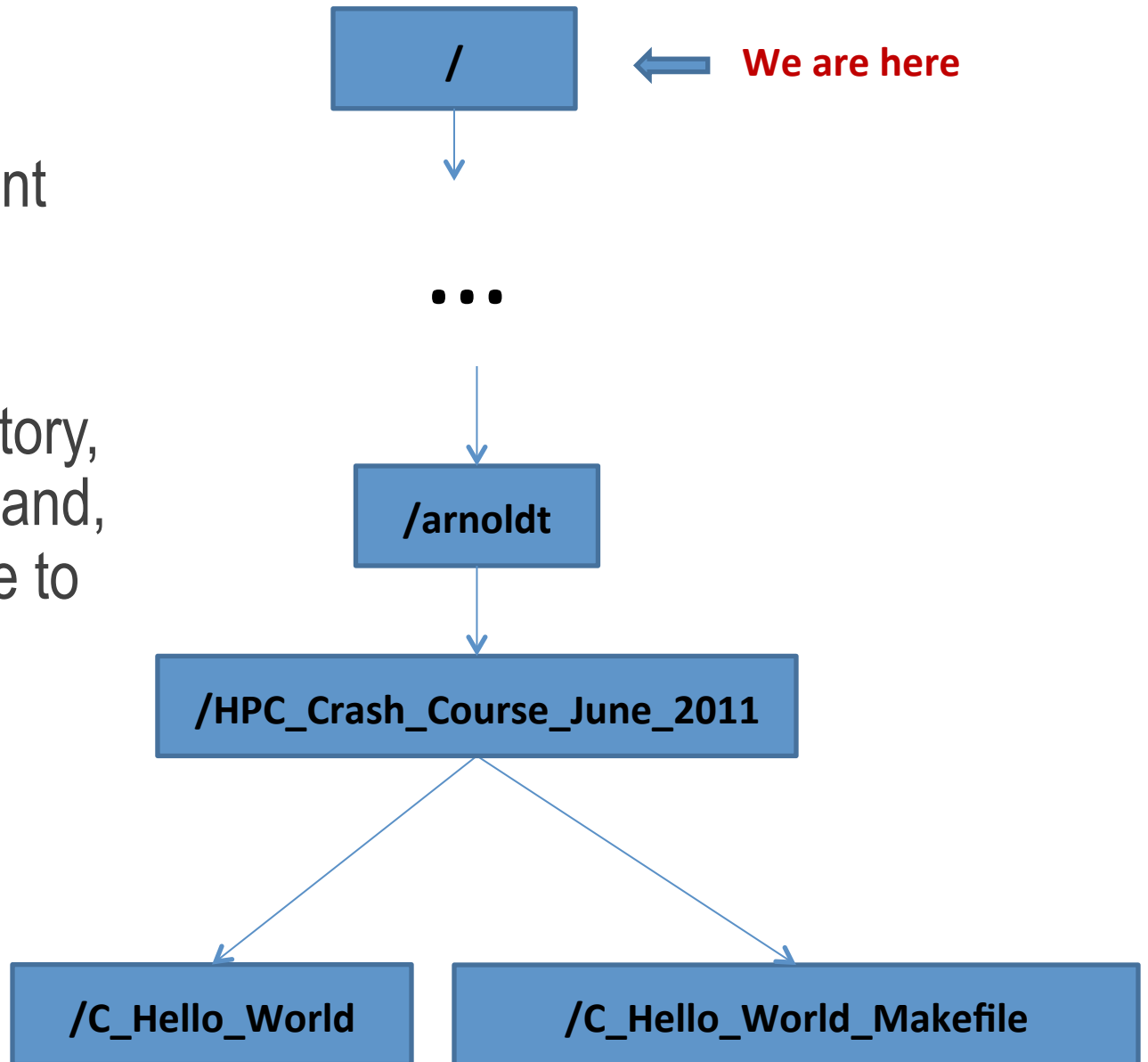
- `cd ./HPC_Crash_Course_June_2011/C_Hello_World`



Exercise

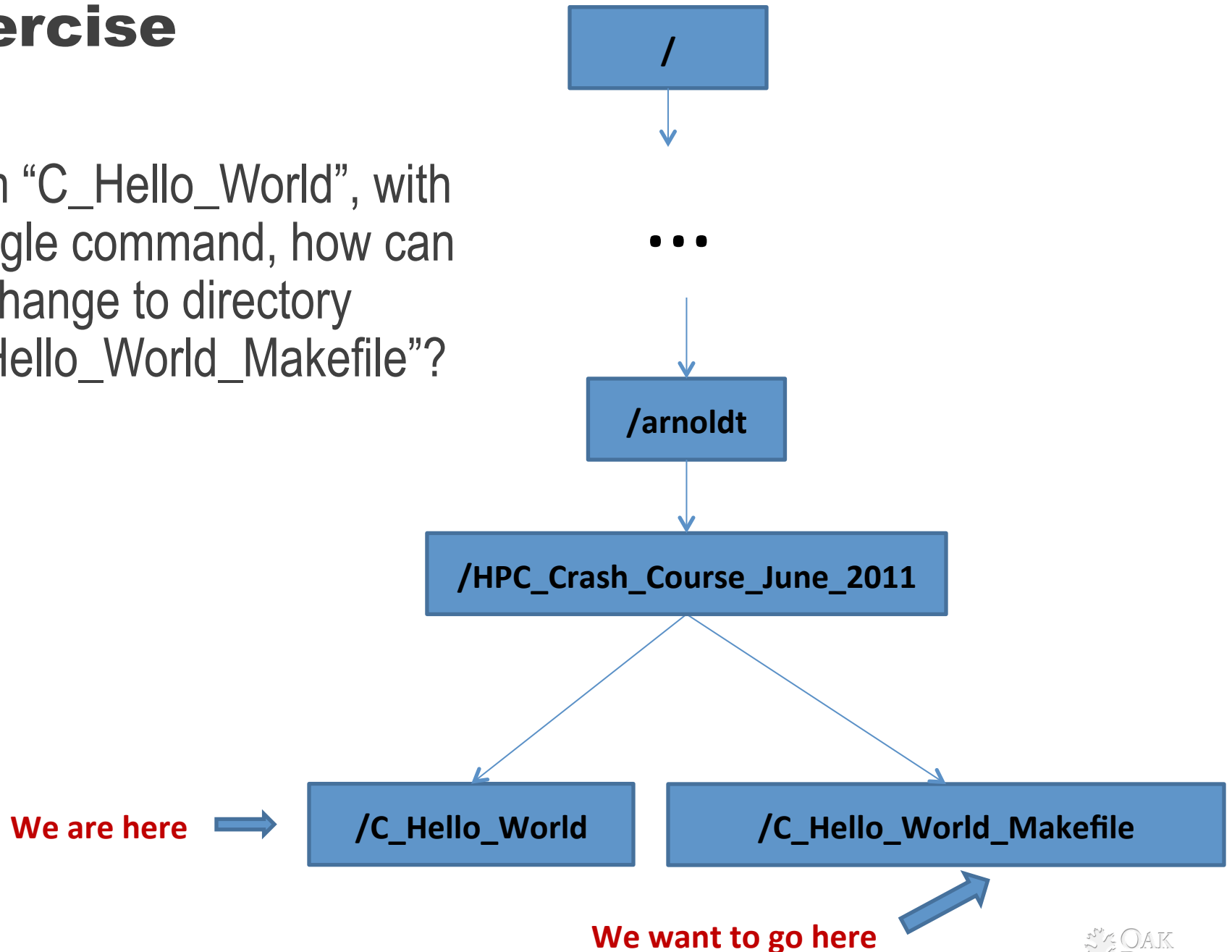
- Change your present location to the root directory
- From the root directory, with a single command, how can we change to directory “C_Hello_World”?

We want to go here



Exercise

- From “C_Hello_World”, with a single command, how can we change to directory “C_Hello_World_Makefile”?

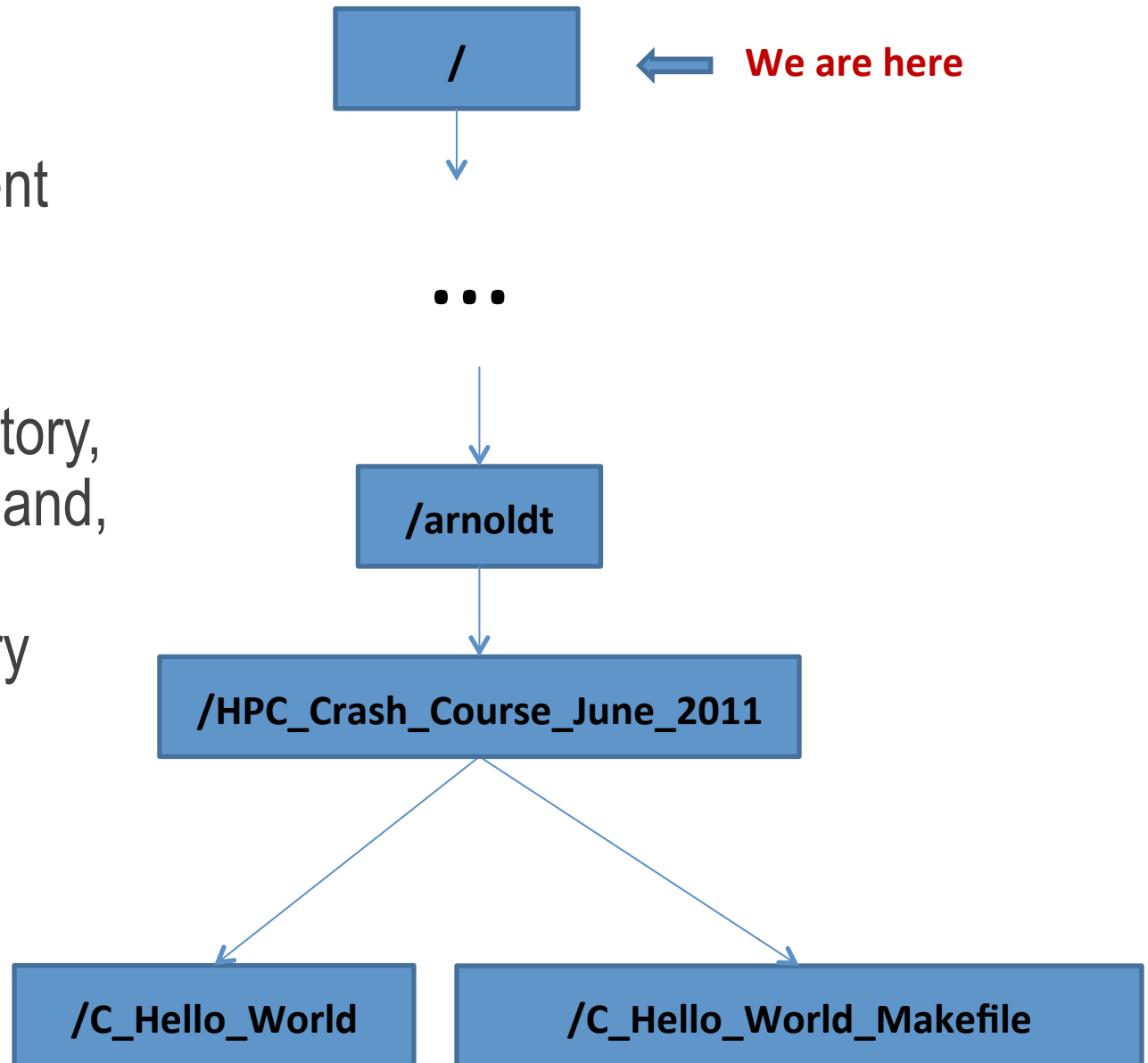


2. Viewing Directories and File Contents

- `ls`
 - list directory contents
- `ls ..`
 - List contents 1 level up
- `ls ../../`
 - List contents 2 levels up
- `ls ./` or `ls`
 - List contents of present directory
- `ls ${HOME}`
 - List contents of HOME directory

Exercise

- Change your present location to the root directory
- From the root directory, with a single command, how can we list the contents of directory “C_Hello_World”?



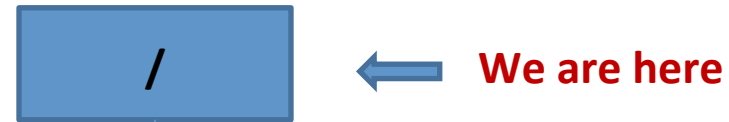
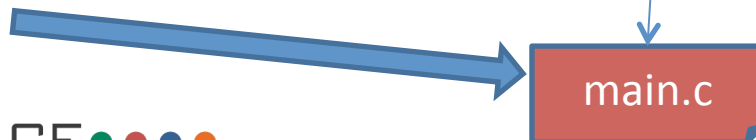
Viewing file contents

- less
 - file perusal filter for crt viewing

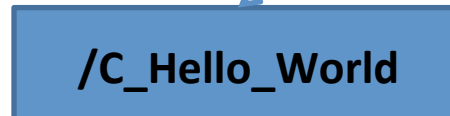
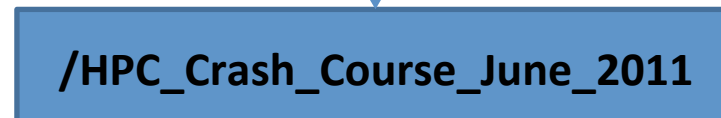
Exercise

- Change your present location to the root directory
- From the root directory, with a single command, how can we view the contents of file main.cpp in directory “C_Hello_World”?

We want to view contents of file here



...



Creating Files and Directories

- `mkdir <path_to_directory>`
- Copying a directory
 - `cp -rf <path_to_original> <path_to_copy>`
- Copying a file
 - `cp -f <path_to_original> <path_to_copy>`
- Removing a directory
 - `rm -rf <path_to_directory>`
- Removing a file
 - `rm -f <path_to_file>`

VI editor

- **Starting the VI Editor**

`vim <path_to_filename>`

- **Exiting the VI Editor**

- Hit the key labeled escape and type one of following depending
 - `:wq` save and quit
 - `:q` quit
 - `:q!` force quit

- **The Two Modes of VI - command and insert**

- The command mode allows the entry of commands to manipulate text.
- The insert mode puts anything typed on the keyboard into the current file.

Unix Job Control

- Job Control commands
 - **jobs**
 - Lists the jobs running in the background, giving the job number.
 - **fg, bg**
 - The fg command switches a job running in the background into the foreground. The bg command restarts a suspended job, and runs it in the background. If no job number is specified, then the fg or bg command acts upon the currently running job.
 - **Control-Z**
 - Stop a foreground job
 - **Control-C**
 - Kills a foreground job
 - **kill %<job_spec>**
 - Kill a job, by specifying its job number after the percent sign

The C language

Every C program has a unique main program

```
int main()
```

```
int main(int argc, char *argv[])
```

C functions are declared as follows

```
return-type function-name(parameter-list,...) { body... };
```

GNU Make

- Make enables the end user to build and install your package without knowing the details of how that is done -- because these details are recorded in the makefile that you supply.
- Make figures out automatically which files it needs to update, based on which source files have changed. It also automatically determines the proper order for updating files, in case one non-source file depends on another non-source file.
- As a result, if you change a few source files and then run Make, it does not need to recompile all of your program. It updates only those non-source files that depend directly or indirectly on the source files that you changed.