PH302 - Computing Skills
Introduction to Programming
Convenor: Dr. Jingqi Miao
Lecturer: Dr. Dean Sayle

Timothy Kinnear
tk218@kent.ac.uk
Room 104 Ingram

Demonstrators:
Paul Cornwall
Chrysa Avdellidou
Ricky Hibbert
Jimmel Stewart

Aims

- To gain a familiarity with Unix based Operating Systems (Linux)
- To learn the Fortran 90 programming language to a basic level
- To appreciate some of the uses of programming within the context of science and data analysis
- To understand some of the principles of how computers work behind GUIs/etc

What are Programming Languages For?

- Need to be able to tell a computer what to do
- Computer operates in machine language, direct binary instructions
- Programming languages provide a level between human understanding, and machine instructions
- Each language has a specific *syntax*
- Then use a *compiler* to convert the language text into an executable program

How are Programming Languages Used?

- Code written in standard ASCII text file (`.f90`)
- Compiler is run on the file
- An *object* file created (`.o`)
- Object file is *linked* to relevant libraries (`.a`/`.so`/`.dll`)
- Executable Binary is created

What Can You Do With Programming?

- Data processing
- Interaction/visualisation of data
- Computation solutions to analytical/mathematical problems ('Numerical Methods/Analysis')
- Computational Simulation
  - Video - '*AREPO*' Fluid Dynamics Code[1];
    Kelvin-Helmholtz Instability and Rayleigh-Taylor Instability.
    - Kelvin-Helmholtz cloud image[2].
    - Jupiter atmosphere animation[3].
  - Video - In-house Astrophysics Fluid Dynamics Code[4];
    Fragmentation of a Molecular Hydrogen Cloud at an HII Boundary Region.

[1] Springel, V. 2010, MNRAS, 401, 791
[2] John Bennett, http://cloudappreciationsociety.org/gallery/photo-07739/, accessed 07/12/12
[3] NASA Multimedia, credit: University of Arizona
[4] Code Described in Miao, J. et al, 2006, MNRAS, 369, 143

Weekly Topics - First Half

Week 13 - The *PCSPS07* Server, Unix and Your First Programs
Week 14 - Using Input and Output in Programs
Week 15 - Conditional Logic and `IF`/`CASE` Statements
Week 16 - Loops and the `DO` Statement
Week 17 - Arrays
Week 18 - Arrays/Subroutines and Functions

Weekly Topics - Second Half

Week 19 - Class Test + Subroutines and Functions
Week 20 - Characters, Strings and Formatting
Week 21 - Advanced Functionalities and Concepts of Fortran
Week 22 - Introduction to Scientific Programming and Numerical Methods
Week 23 - Mini-project
Week 24 - Class Test

Workshop Sessions

	Monday	Tuesday	Wednesday	Thursday	Friday
9:00-10:00		Group 2 Weeks 13-18 & 20-23 ECT1 CA			
10:00-11:00		Group 1 Weeks 13-18 & 20-23 ECT1 CA			
11:00-12:00		Class Tests Week 19 : Eliot Hall Week 24 : Sports Hall Check Email/Moodle Group 3 Weeks 13-18 & 20-23 ECT1 CA			
12:00-13:00				Group 3 Weeks 13-24 ECT1 PC	
13:00-14:00					
14:00-15:00					Group 3 Weeks 13-24 ECT1 RH, JS
15:00-16:00				Group 1 Weeks 13-24 ECT1 PC	Group 1 Weeks 13-24 ECT1 RH, JS
16:00-17:00					Group 2 Weeks 13-24 ECT1 RH, JS
17:00-18:00	Lecture Week 13 RLT1 All			Group 2 Weeks 13-24 ECT1 PC	

Assessment

Week				
Uni	Course	Programs	Questions	Date due in
13	1	0	No	
14	2	2	Yes	Friday (31/01/13)
15	3	2	Yes	Friday (07/02/13)
16	4	2	Yes	Friday (14/02/13)
17	5	2	Yes	Thursday (27/02/13, week 18)
18	6	0	No	
19	7	2	Yes	Friday (07/03/13)
20	8	1	Yes	Friday (14/03/13)
21	9	2	Yes	Friday (21/03/13)
22	10	2	Yes	Friday (28/03/13)
23	11	1	No	Friday (11/04/13, week 24)
24	12	0	No	

Assessment

- Mid-term class test:
  Tuesday, 4th March, 11:00-12:00, Eliot Hall
- End of term class test:
  Tuesday, 8th April, 11:00-12:00, Sports Hall

Creating Your Account

- Use 'Putty' to ssh to the server '`PCSPS07.kent.ac.uk`'
- Log in with your ITS username and initial password '`[ITS username]_[university number]`' (remember your student card!)
- e.g. for ITS username '`aa000`' and student number '`987654321`', password would be:

  `aa000_987654321`
- Follow instructions presented to reset your password
- Putty will close after this is done, your password will be set to whatever you selected, and your account will be ready for subsequent logins

Using PCSPS07 via Xming

- Run the 'Xming X Launch Wizard' application (search within Start menu)
- Select 'One Window' mode
- Select 'Open session via XDMCP'
- Enter '`PCSPS07.kent.ac.uk`' as the server
- If desired, save the configuration to create a shortcut to do the above automatically in future
- Enter username and password to log in

Exiting Xming

- **Do not** exit Xming by selecting a 'log out' or 'exit' option within the virtual desktop
- **Do** close Xming directly using the 'x' close window icon in the upper right, or the Alt-F4 keyboard shortcut

Some Terms

- *Terminal* - The bit you type in
- *Shell* - The language and 'interface' between actions in the terminal, and what is performed by the computer
- '*Bash*' - The shell which you will be using for this course; '**B**ourne **A**gain **SH**ell'
- '*ssh*' - **S**ecure **SH**ell, protocol which lets you interact with a Unix machine remotely
- '*X*' or '*X11*' - The set of instructions with which graphics are displayed on screen in most Unix versions
- *Directory* - Equivilent to 'folder' on Windows machines

Structure of Terminal Commands

- `command [-o|-p] [argument1] [argument2] [argument3]`
- `ls` → lists contents of current directory
- `ls ./mysubdir/` → lists contents of the sub directory '`mysubdir`'
- `ls -l` → lists contents of current directory in 'long' mode
- `ls -l ./mysubdir/` → lists contents of the sub directory '`mysubdir`' in 'long' mode
- `ls -l -a` → lists contents of current directory in 'long' mode, including 'hidden' files

Structure of Terminal Commands

- `command myinputfile`
  - produces → `myinputfile.out`

Structure of Terminal Commands

- `command myinputfile`
  - produces → `myinputfile.out`
- Additional option '`-o`' to specify what the output name should be

Structure of Terminal Commands

- `command myinputfile`
  - produces → `myinputfile.out`
- Additional option '`-o`' to specify what the output name should be
- `command -o myoutputfile myinputfile`
  - produces → `myoutputfile`

Structure of Terminal Commands

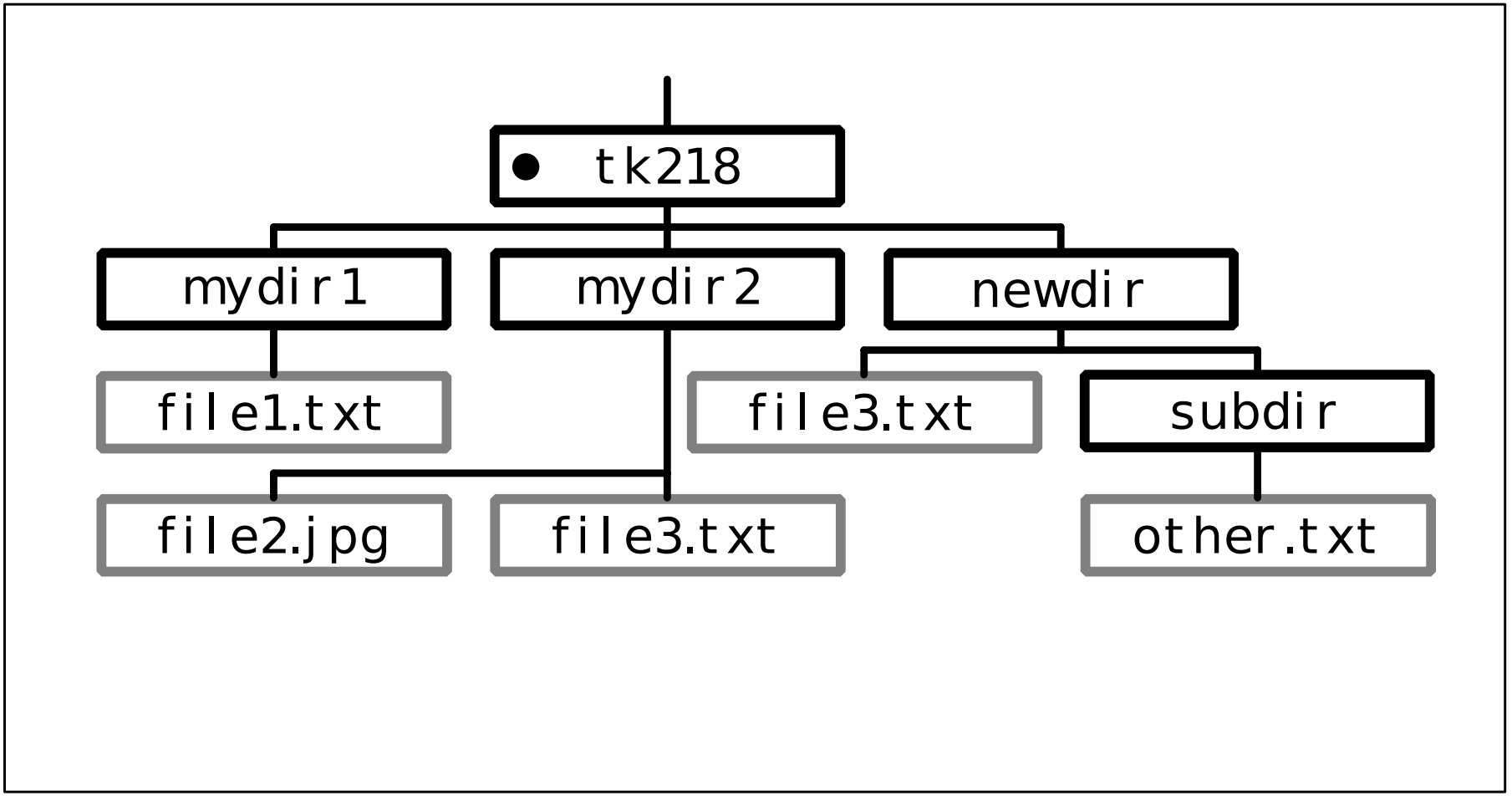
- `command myinputfile`
  - produces → `myinputfile.out`
- Additional option '`-o`' to specify what the output name should be
- `command -o myoutputfile myinputfile`
  - produces → `myoutputfile`
- `command [ -a ] [ -b [option_argument] ] [command_argument]`

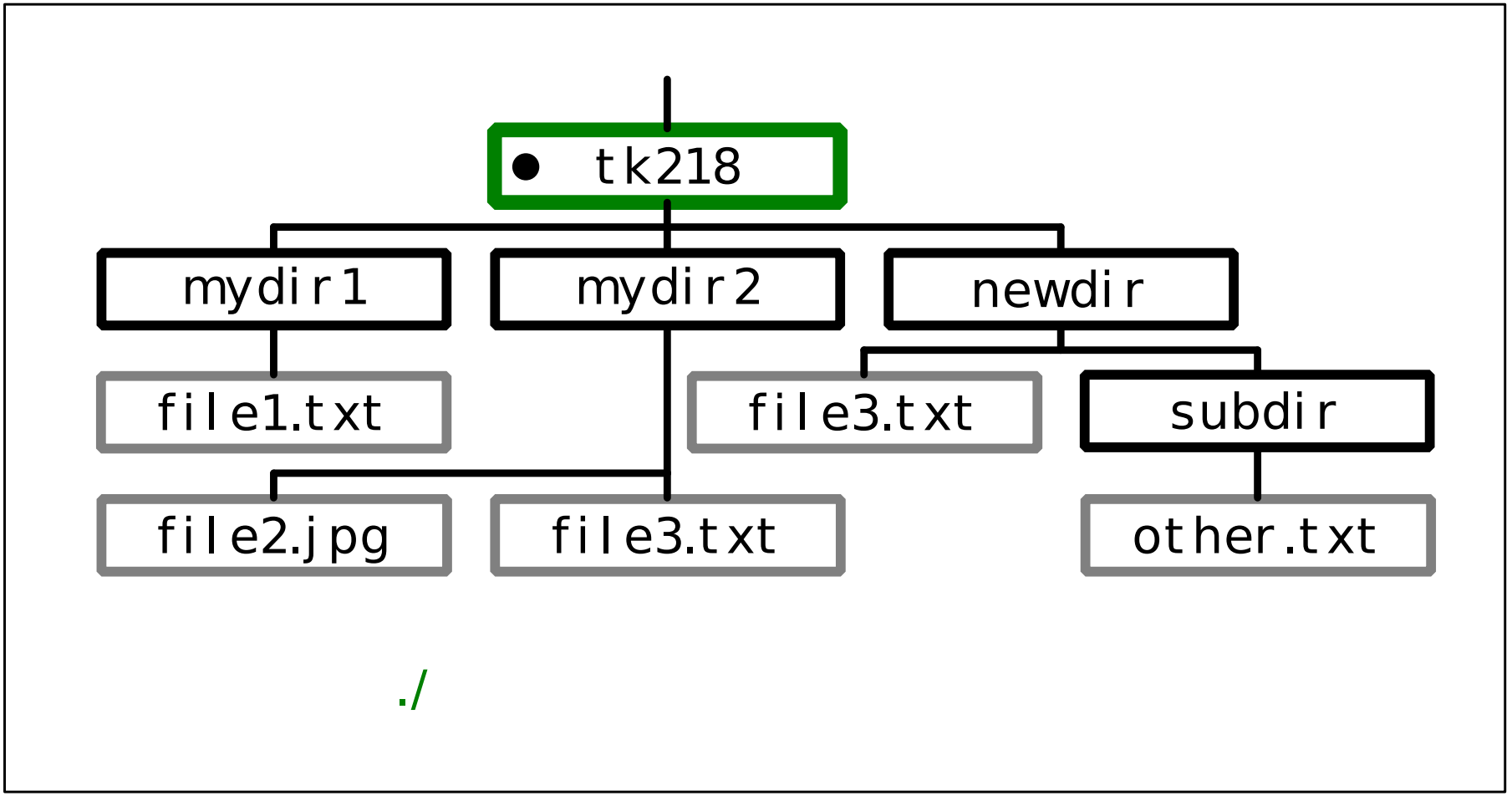
Structure of Terminal Commands

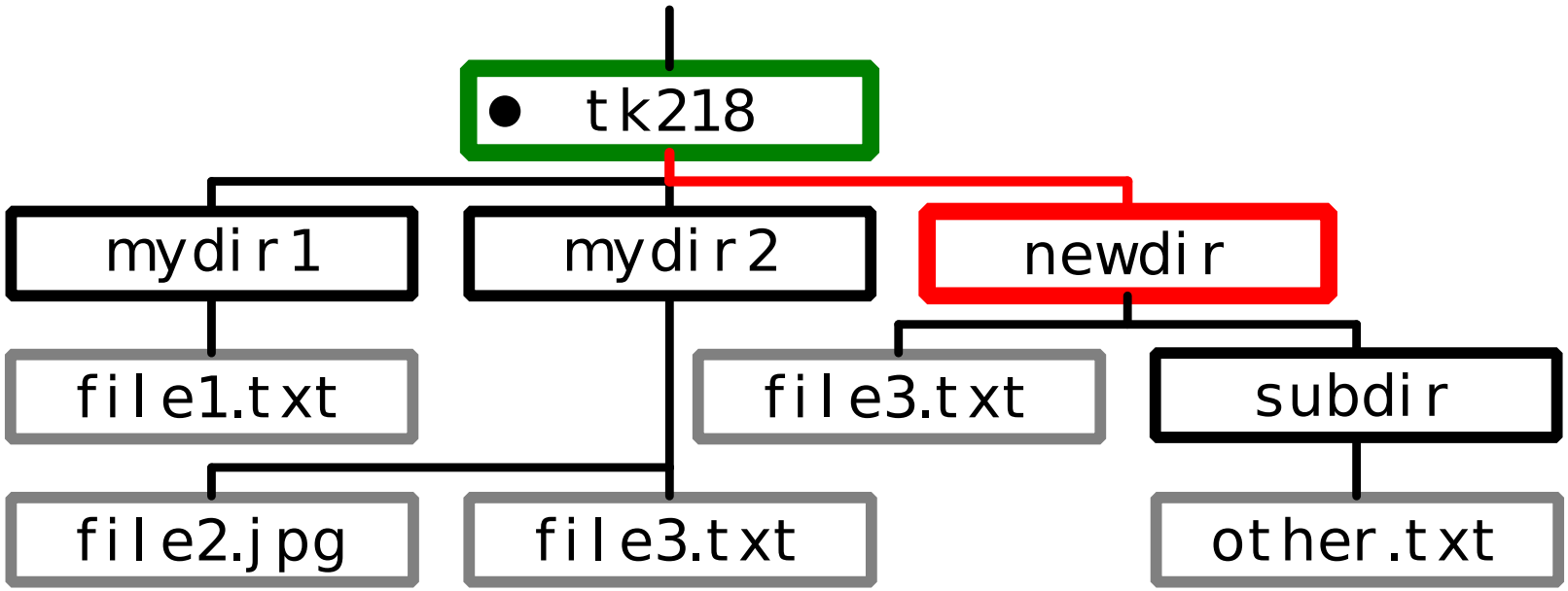
- `command myinputfile`
  - produces → `myinputfile.out`
- Additional option '`-o`' to specify what the output name should be
- `command -o myoutputfile myinputfile`
  - produces → `myoutputfile`
- `command [ -a ] [ -b [option_argument] ] [command_argument]`
- Like mathematical functions:
  - $f\left(x, g\left(y\right), z\right)$
  - $\mathrm{Command}\left(a, b\left(\mathrm{optarg}\right), \mathrm{comarg}\right)$

Important Commands

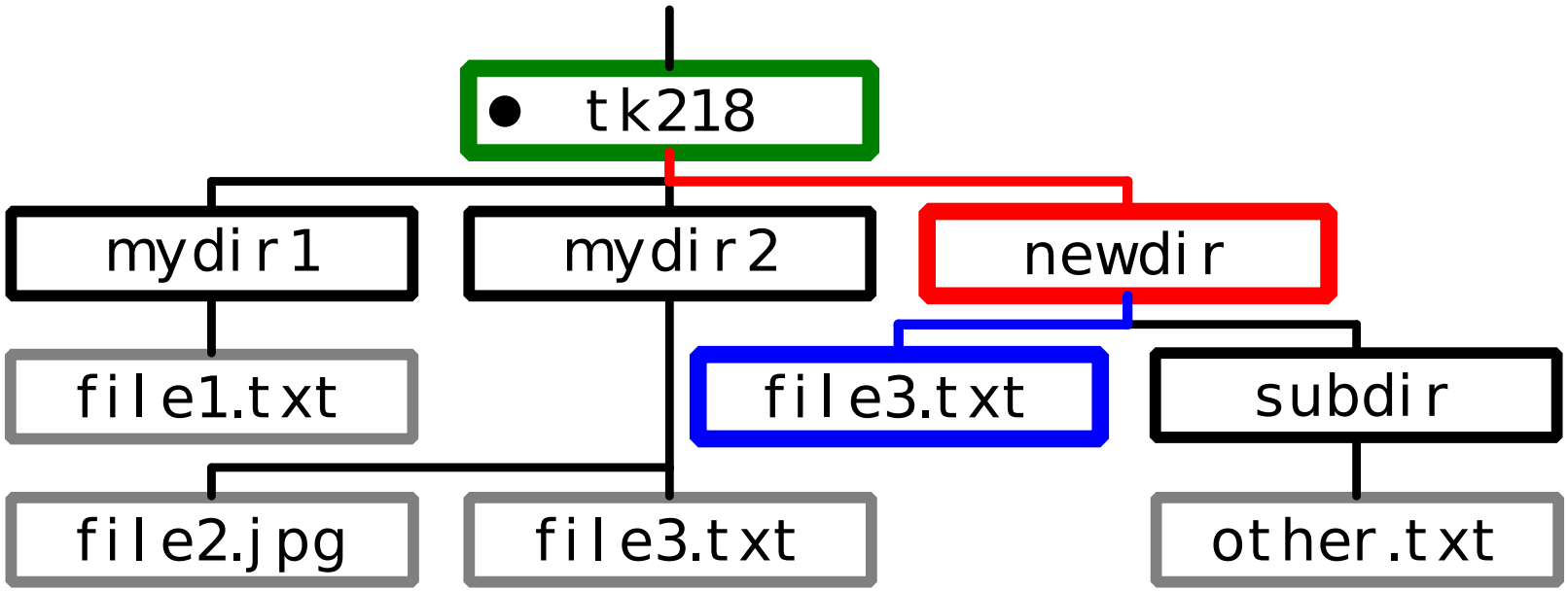
- `man commandname` → **MAN**ual page for the command '`commandname`'
- `ls` → **L**i**S**t, lists contents of a directory
- `cd targetdir` → **C**hange **D**irectory, moves to directory specified by '`targetdir`'
- `mkdir dirname` → **M**a**K**e **DIR**ectory, makes a directory named '`dirname`'
- `cp from to` → **C**o**P**y, copies the file '`from`' to the location/name '`to`'
- `mv from to` → **M**o**V**e, moves the file '`from`' to the location/name '`to`'
- `rm filename` → **R**e**M**ove, deletes the file '`filename`'
- `rm -r dirname` → '**R**ecursive' option, deletes the directory '`dirname`'


tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt

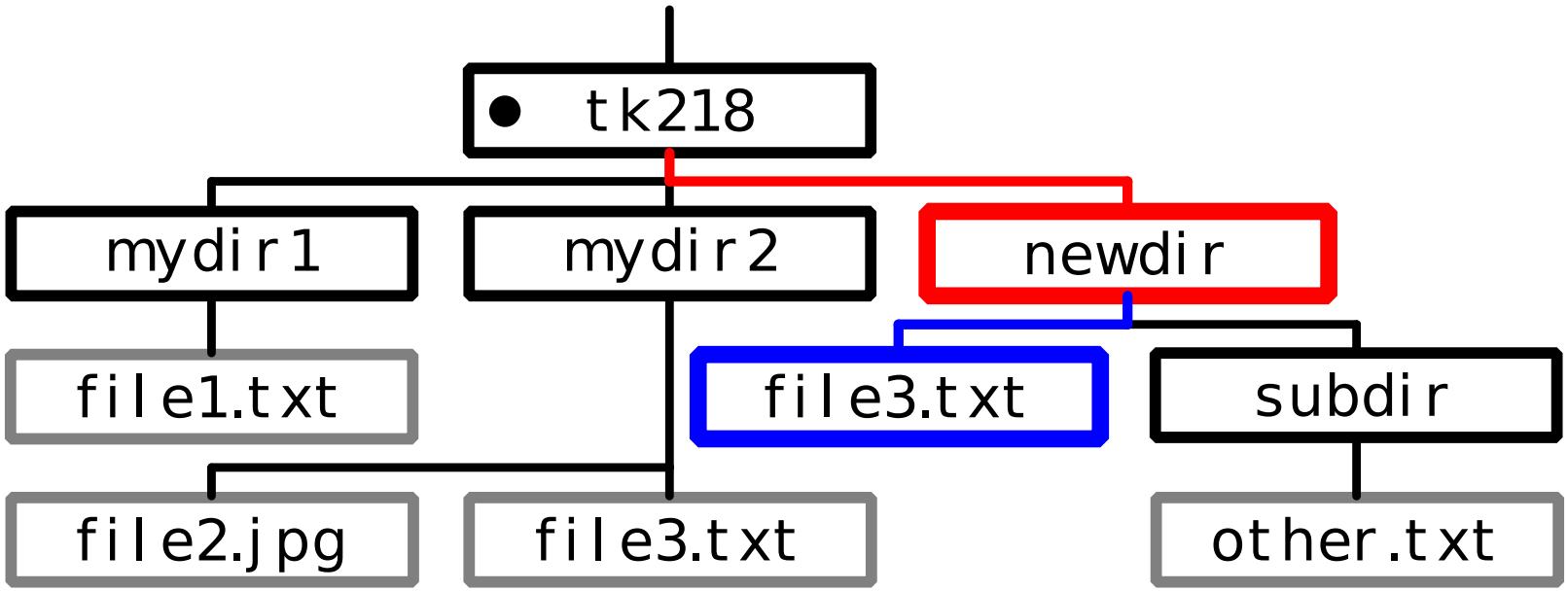

tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt
./


tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt

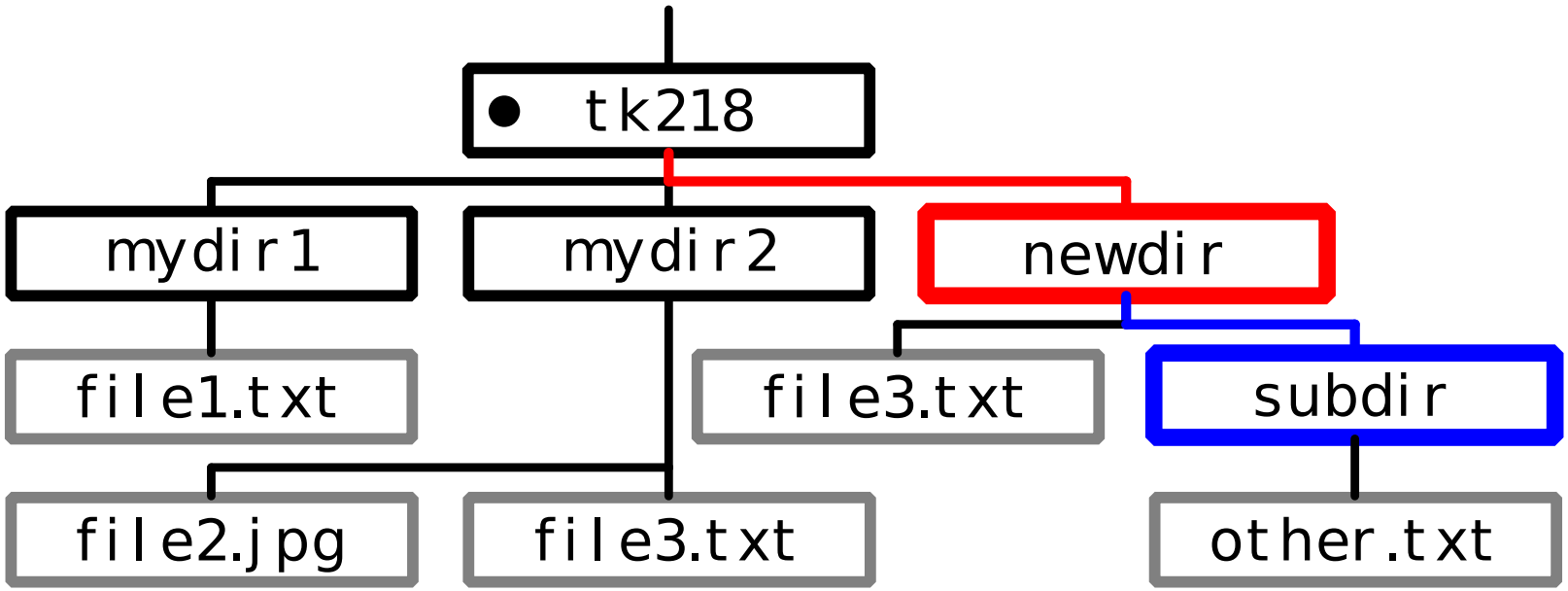
./newdir/



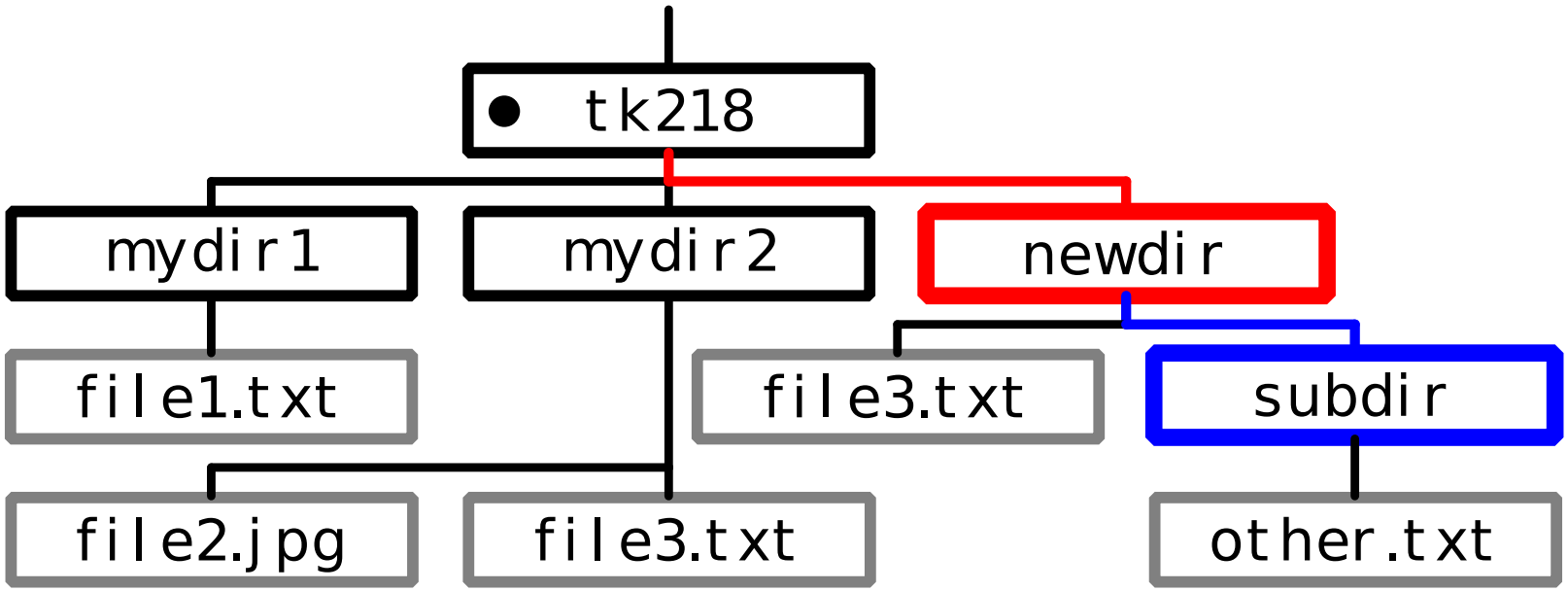
./newdir/file3.txt



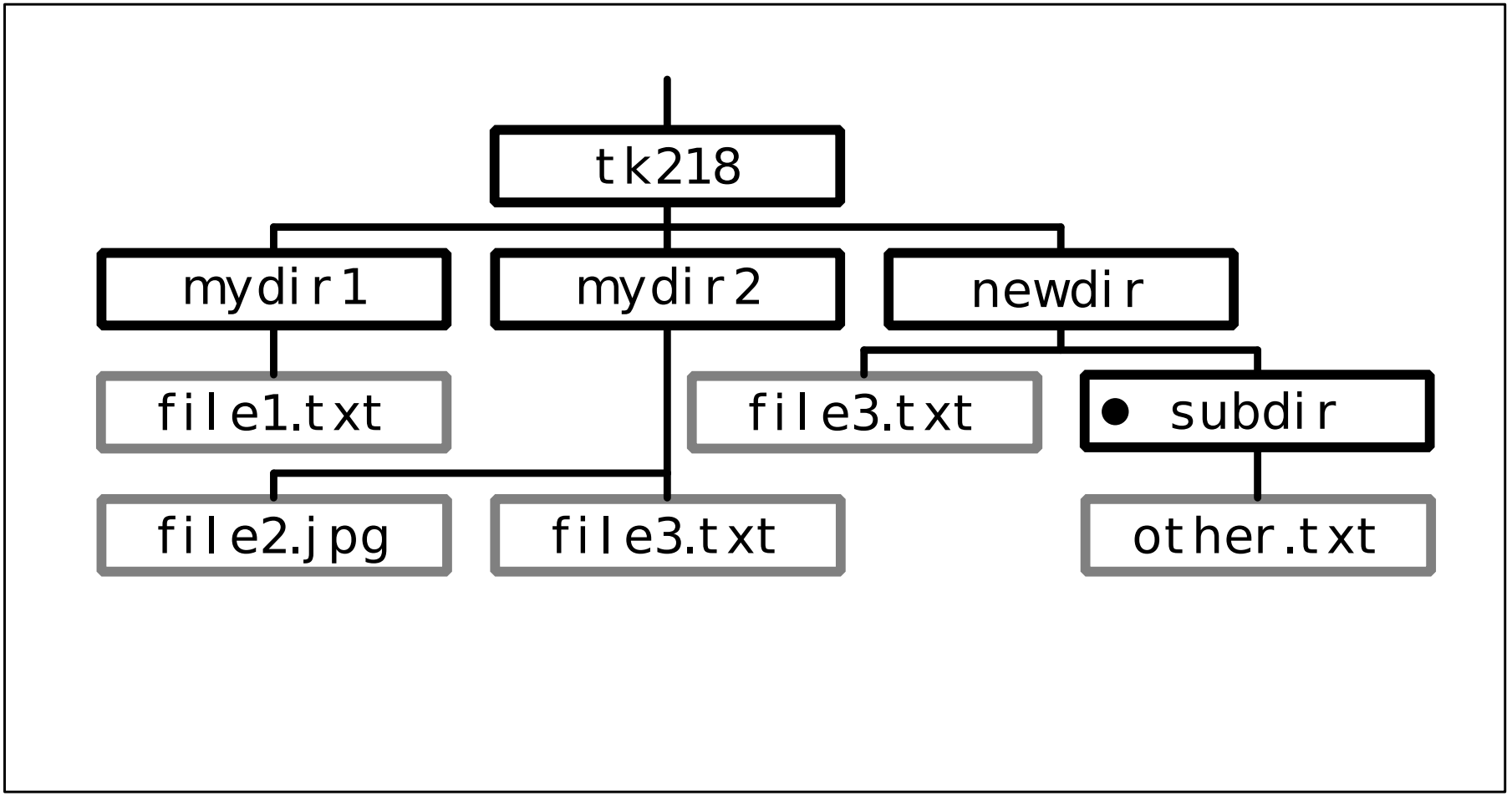
newdir/file3.txt

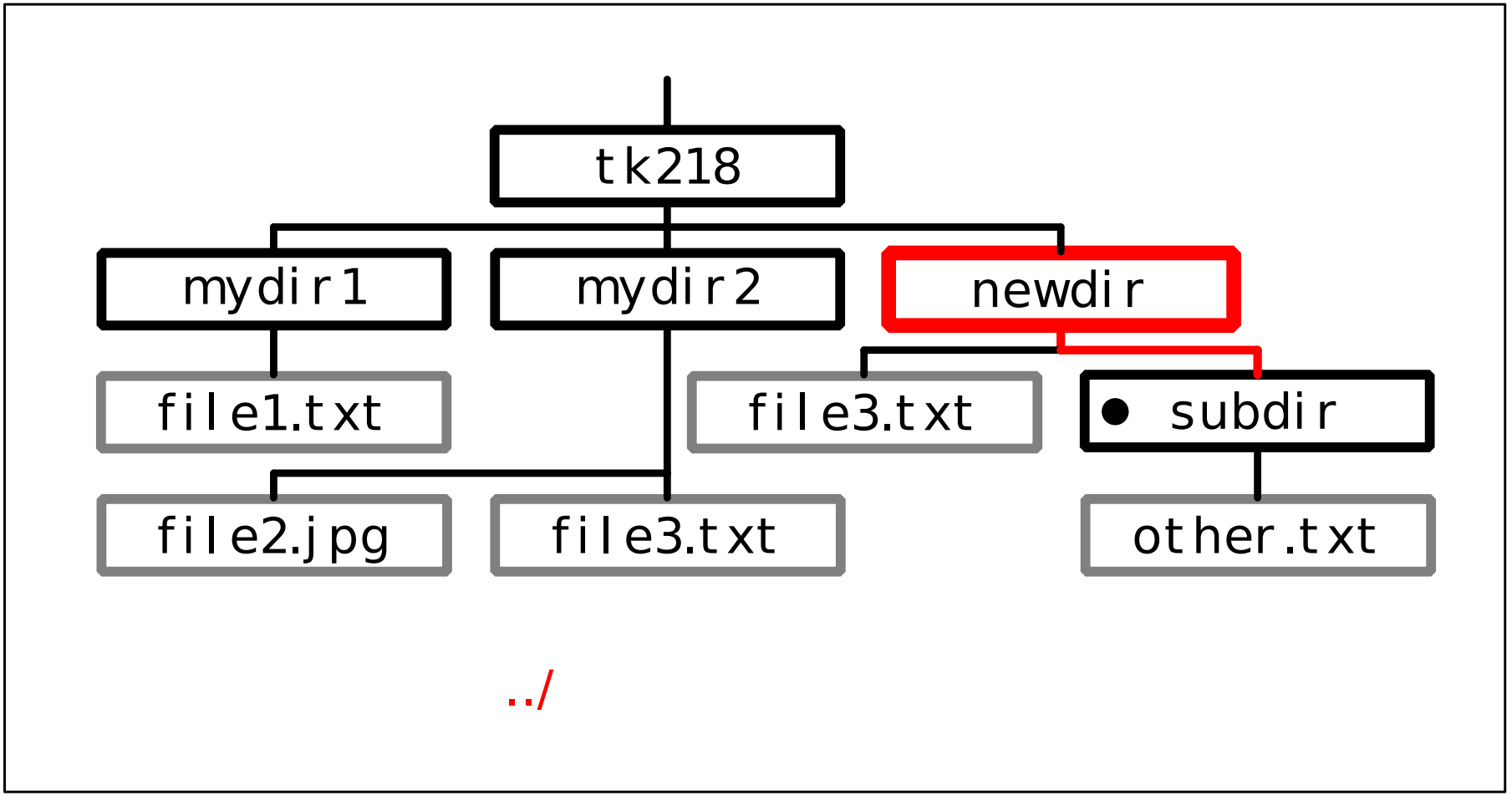


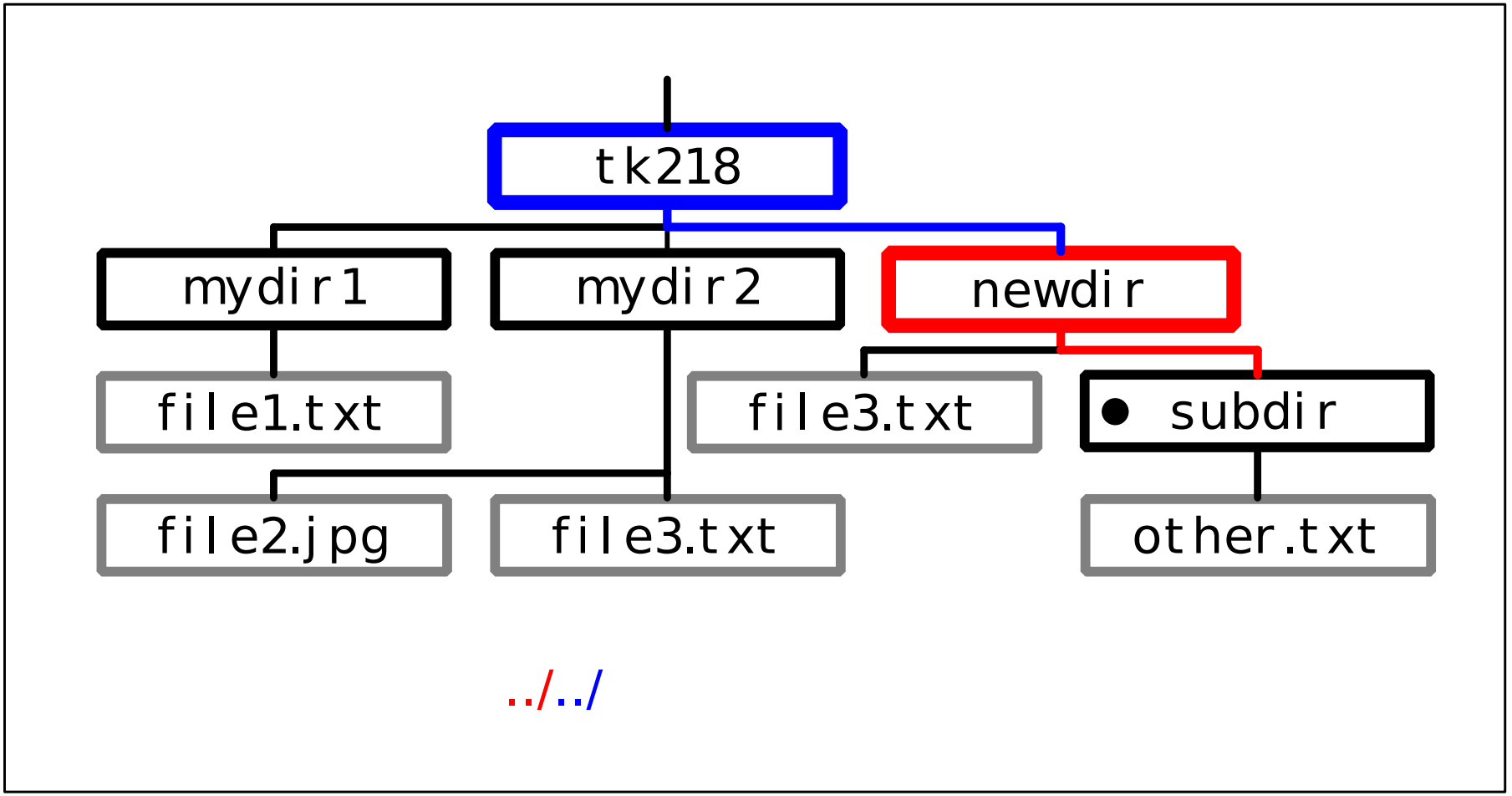
newdir/subdir/

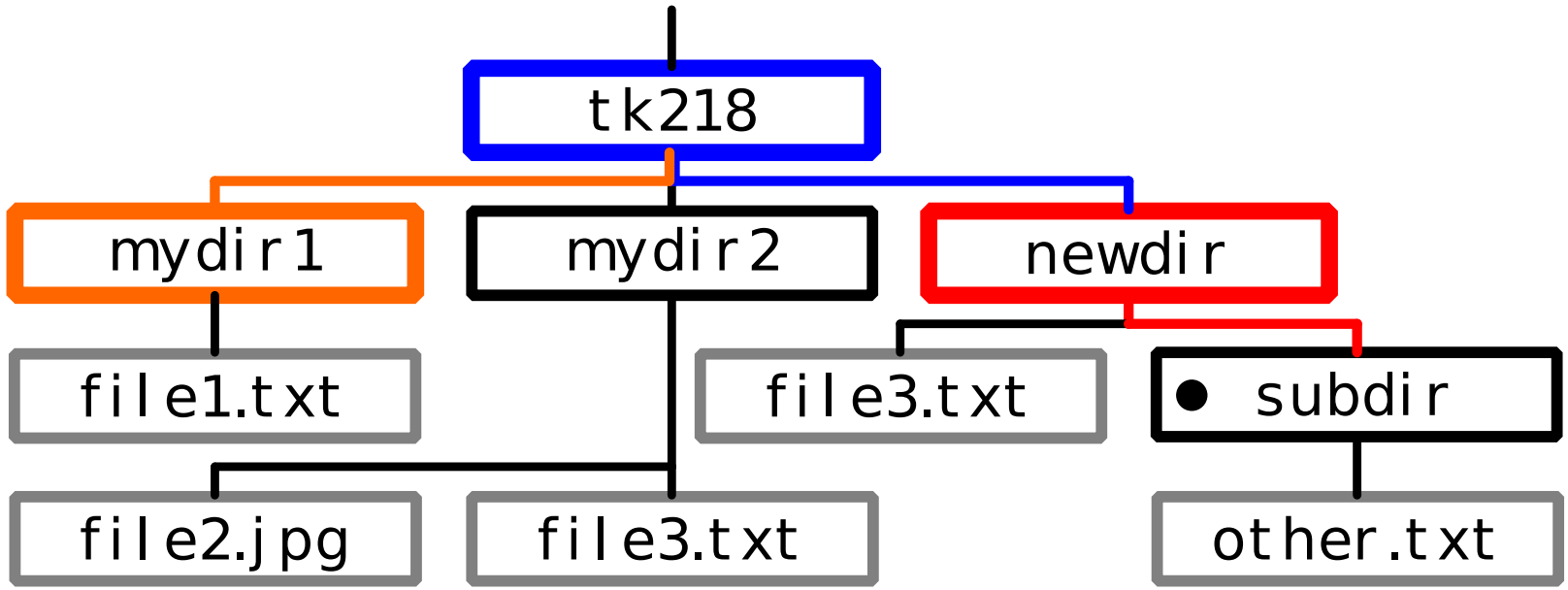


cd newdir/subdir/

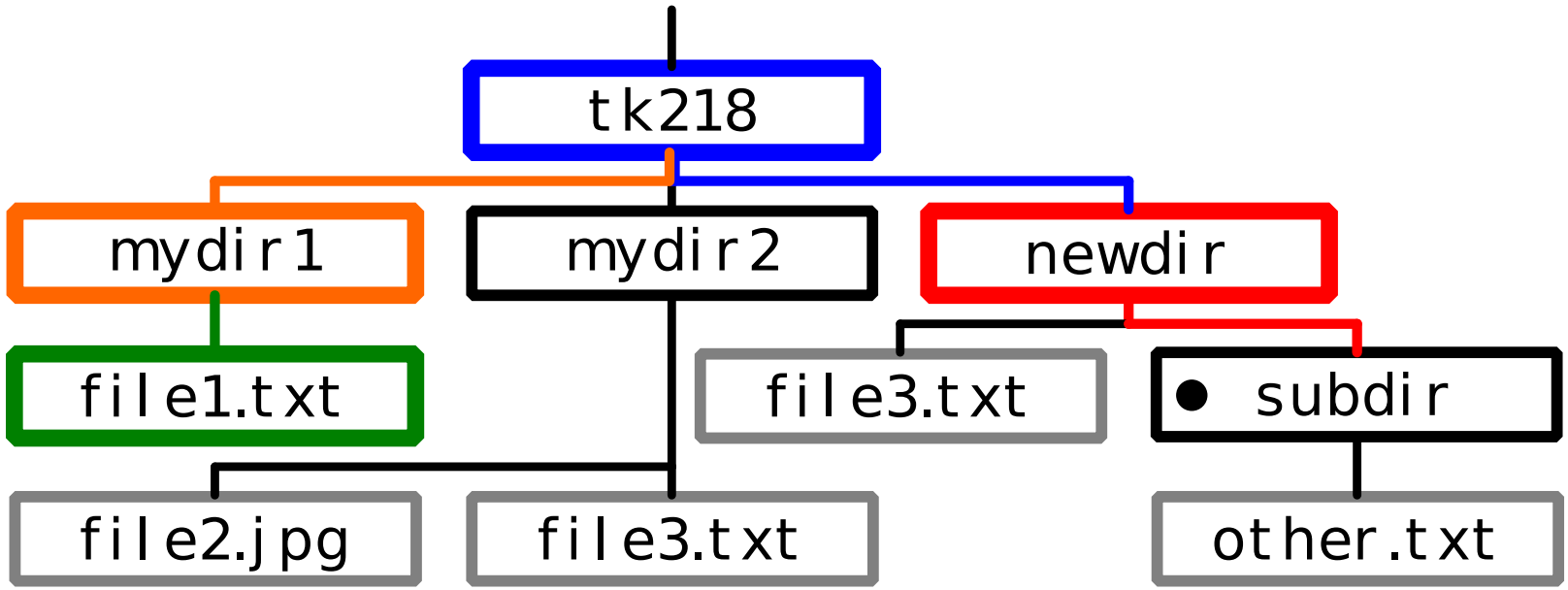

tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt


tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt
../

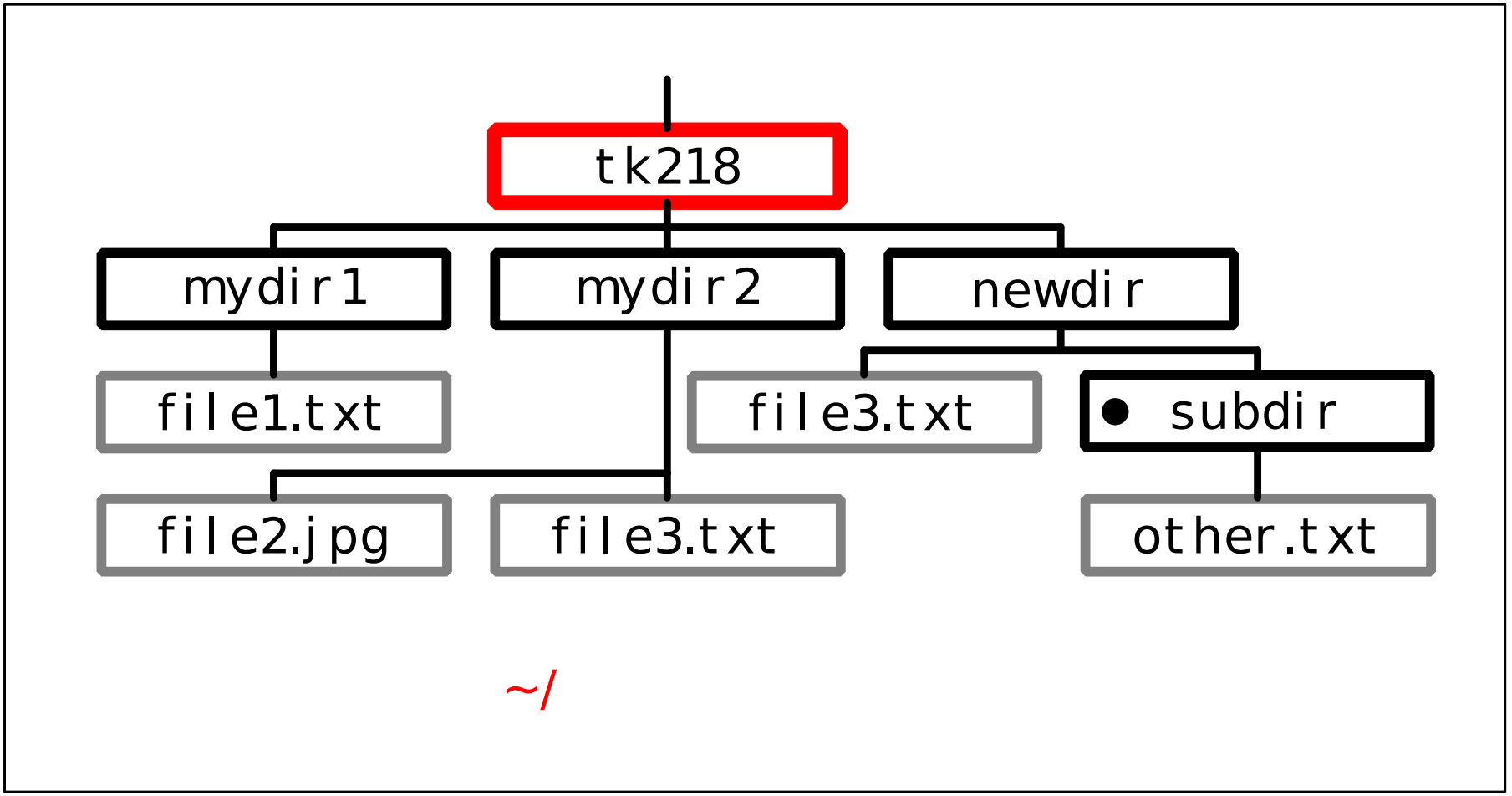

tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt
../../

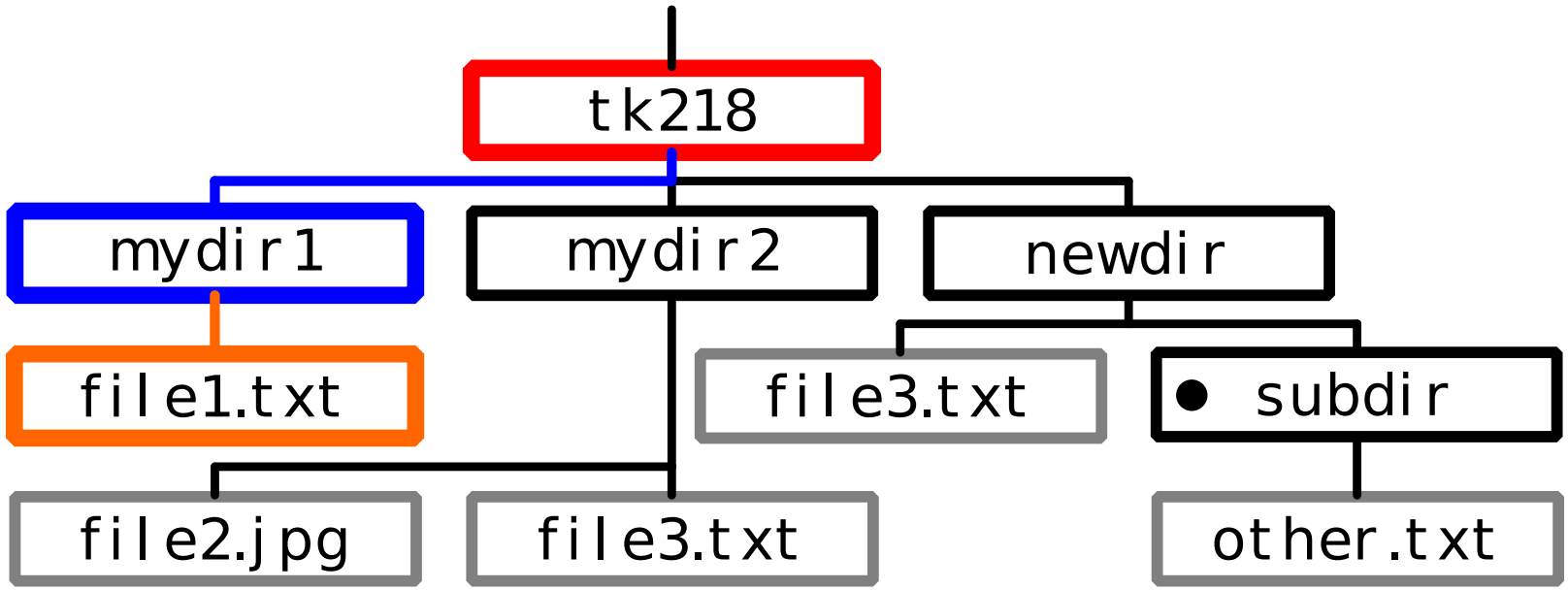

tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt

../../mydir1/

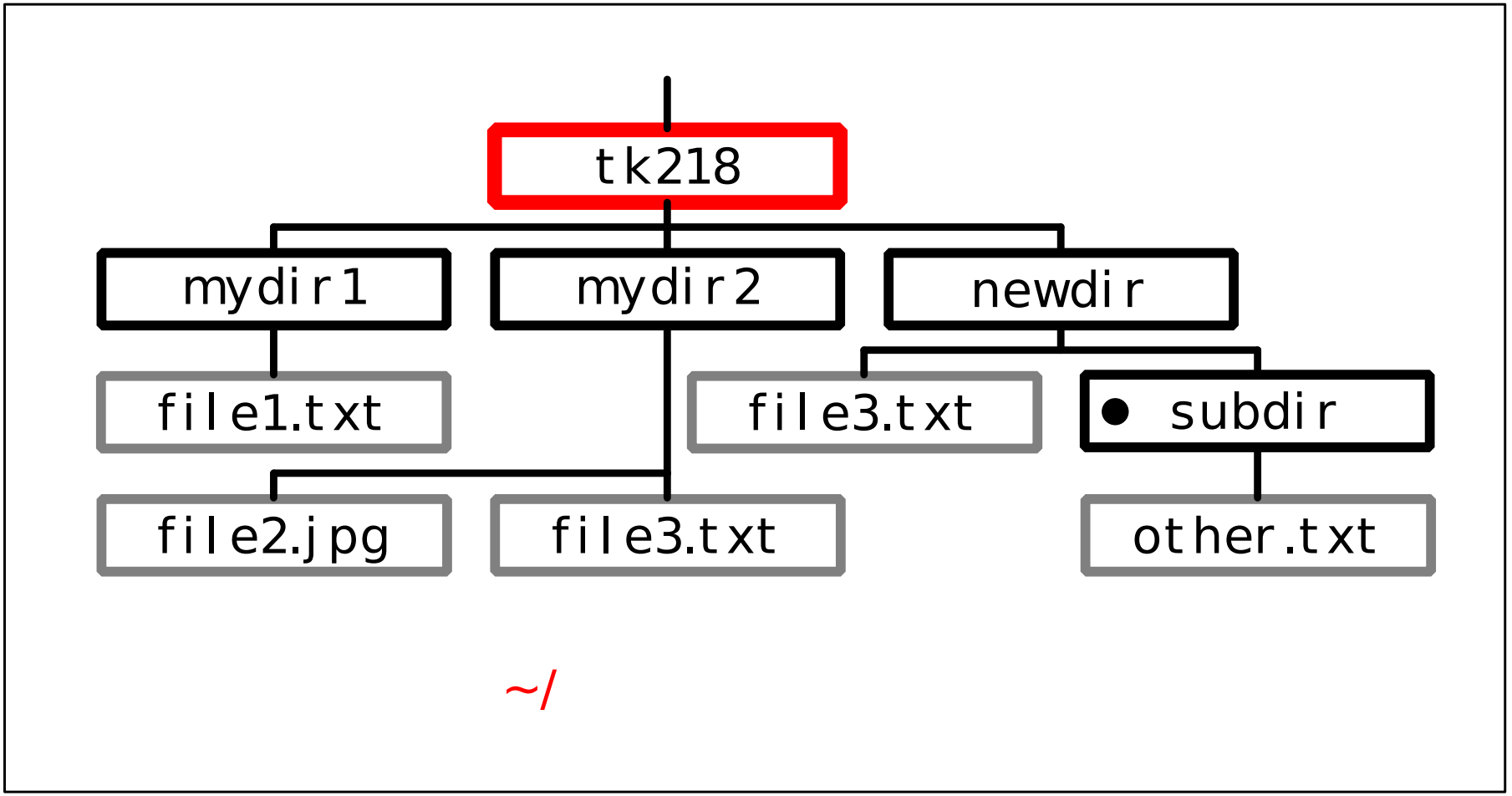


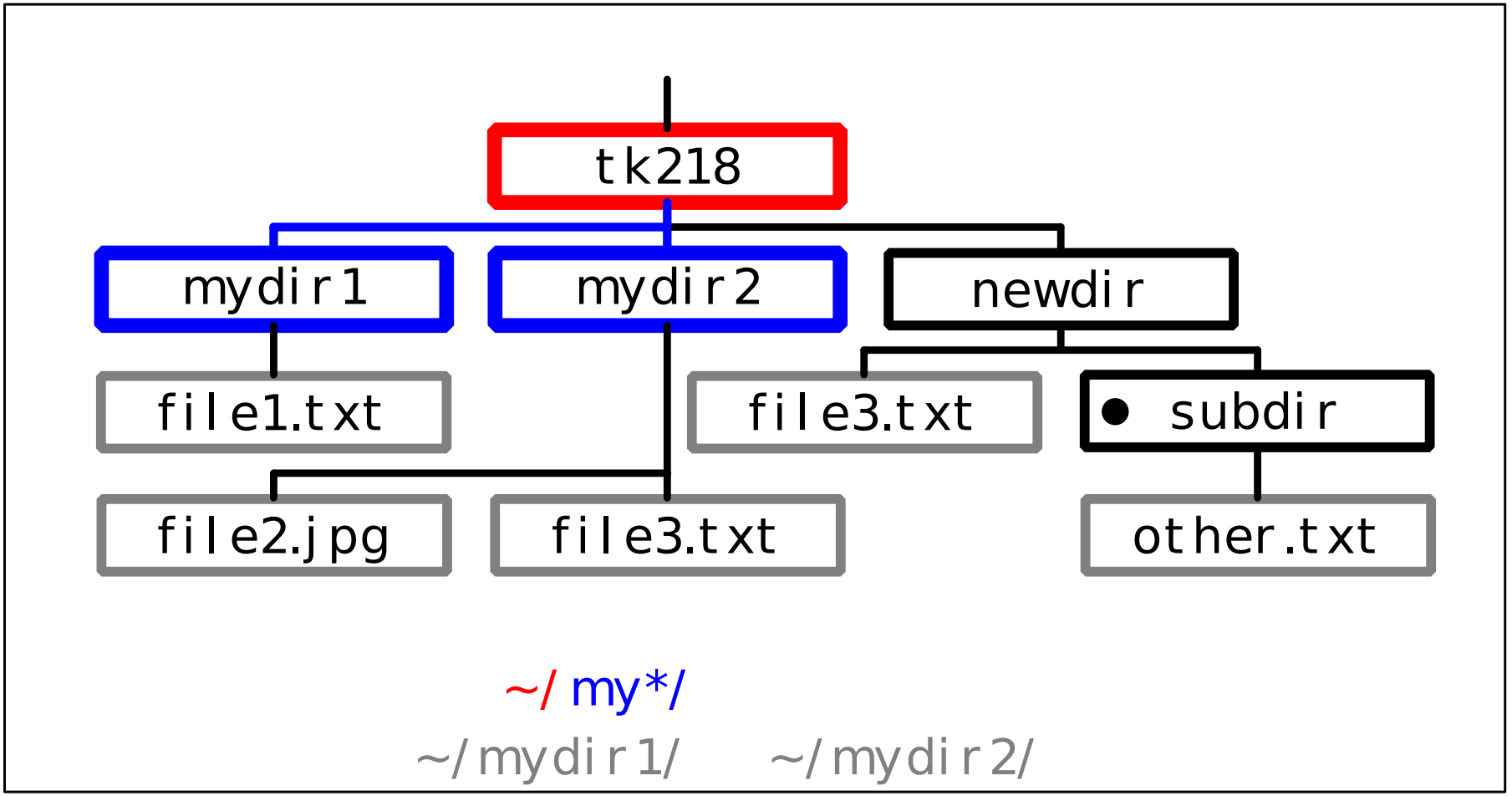
../../mydir1/file1.txt

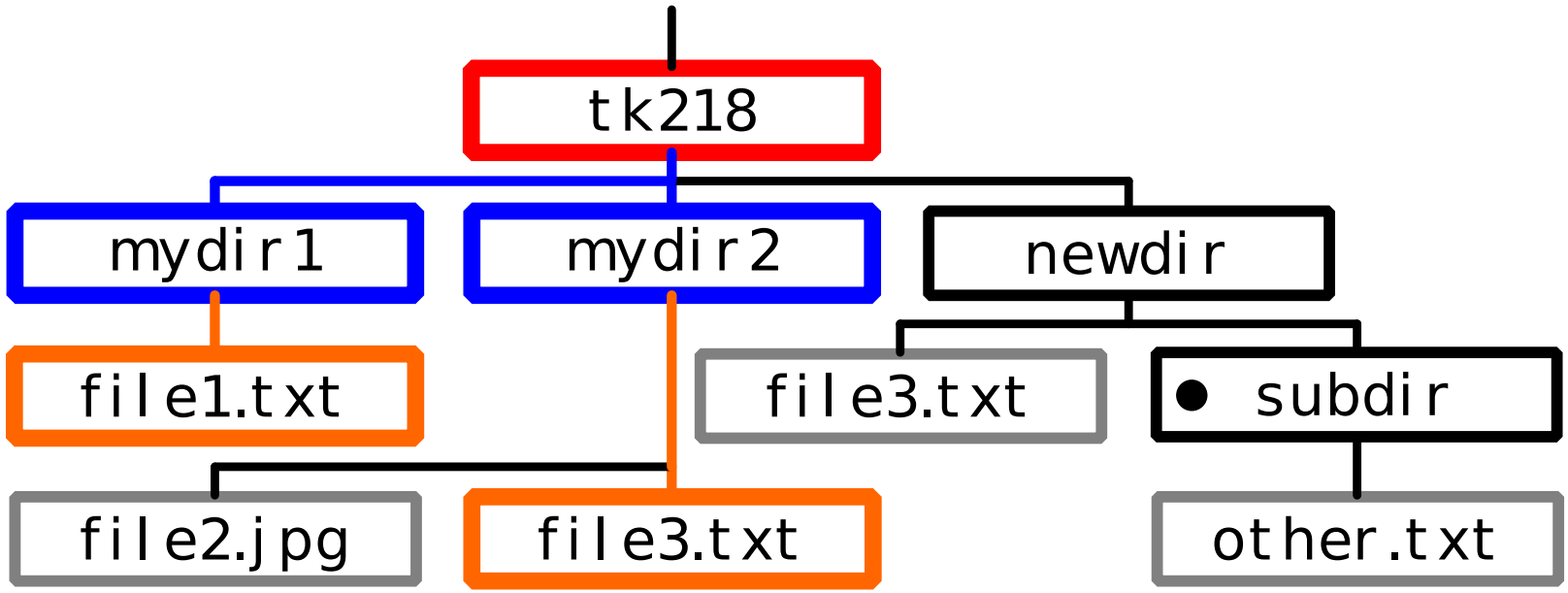

tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt
~/



~/mydir1/file1.txt

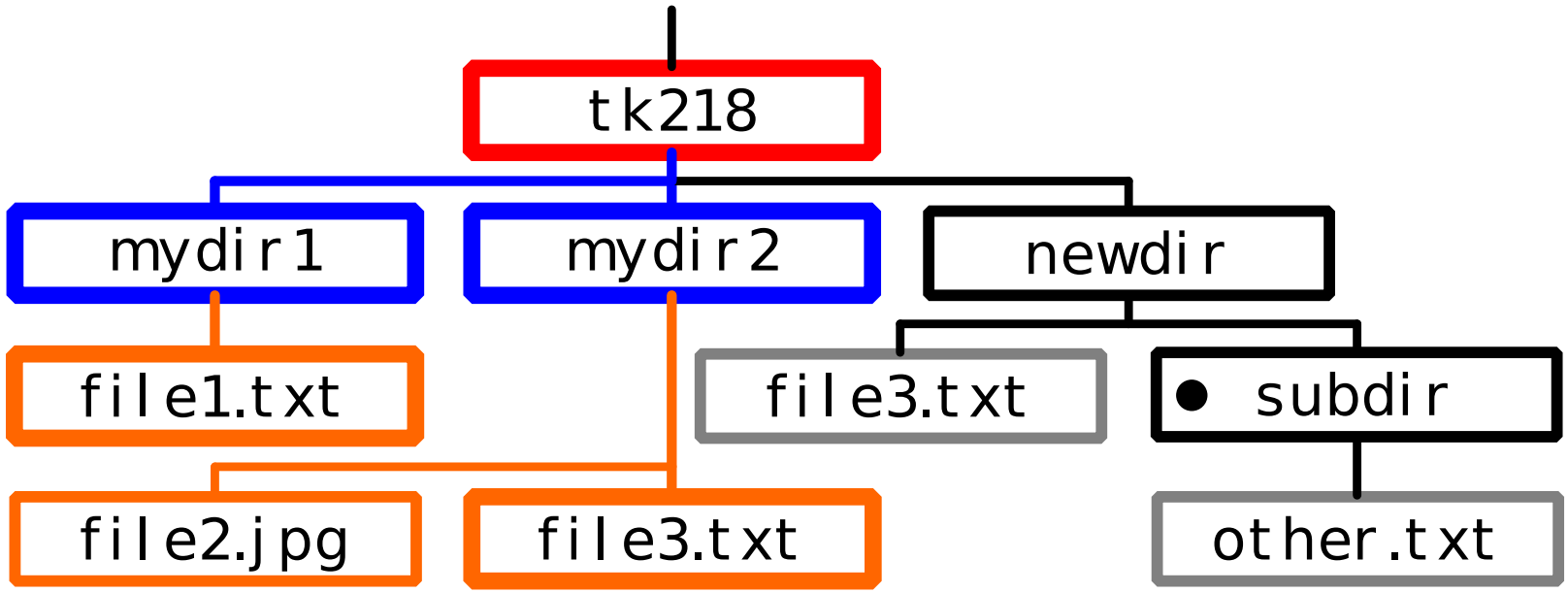

tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt
~/


tk218
mydir1
mydir2
newdir
file1.txt
file3.txt
subdir
file2.jpg
file3.txt
other.txt
~/my*/
~/mydir1/
~/mydir2/



~/my*/*.txt

~/mydir1/file3.txt ~/mydir2/file1.txt



~/my*/file*

~/mydir1/file3.txt ~/mydir2/file2.jpg ~/mydir2/file1.txt

What is Fortran?

- 1957, First Version of FORTRAN
- 1962, FORTRAN IV
- 1966, FORTRAN 66
- 1977/1980, FORTRAN 77
- 1991, Fortran 90
  - Many modernised features
  - 'free form'
  - Still fully backwards compatible with F77
- 1997, Fortran 95
  - Minor changes to F90
  - Some obselete features removed permanently - not fully back-compatible with F77
- 2004, Fortran 2003
  - Still propriatory - No free compilers
- 2010, Fortran '2008'
  - Minor changes, similar to F90 $\rightarrow$ F95

Hello World Code

```
PROGRAM helloworld
  IMPLICIT NONE
  !print out the phrase "hello world" to the terminal
  WRITE(*,*) 'Hello World'
END PROGRAM helloworld
```

Compiling and Running Code

Compiling:
```
:> gfortran -o helloworld helloworld.f90
```
(Compiling:
```
:> gfortran [-o (helloworld)] [helloworld.f90] )
```
Running:
```
:> ./helloworld
```
Output:
```
Hello World
```
Empty Prompt:
```
:> _
```

References

Using GNU Fortran; for GCC version 4.5.0. Free Software Foundation, 2008.
Springel, V. 2010, MNRAS, 401, 791
John Bennett, http://cloudappreciationsociety.org/gallery/photo-07739/ accessed 07/12/12
Animation of Jupiter, NASA Multimedia, University of Arizona
Miao, J. et al, 2006, MNRAS, 369, 143
Standards Documents. http://gcc.gnu.org/wiki/GFortranStandards

Any Questions?