

# Paths and Files

CS 272 Software Development

# Java IO, NIO, NIO.2

- Package **java.io** (input/output) was originally introduced in JDK 1.0 in 1996
- Package **java.nio** (non-blocking I/O) was originally introduced in J2SE 1.4 in 2002
- Package **java.nio.file** (new I/O) was originally introduced in Java SE 7 in 2011

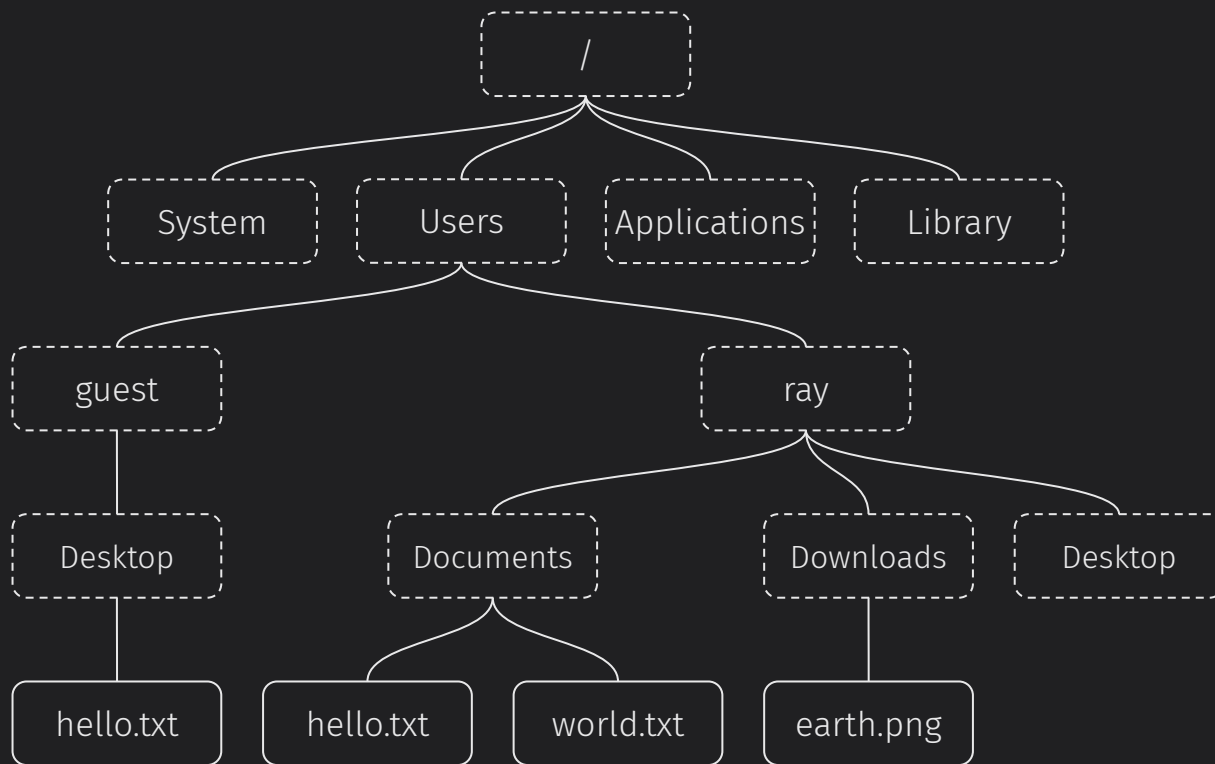
<https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/doc-files/coll-index.html>

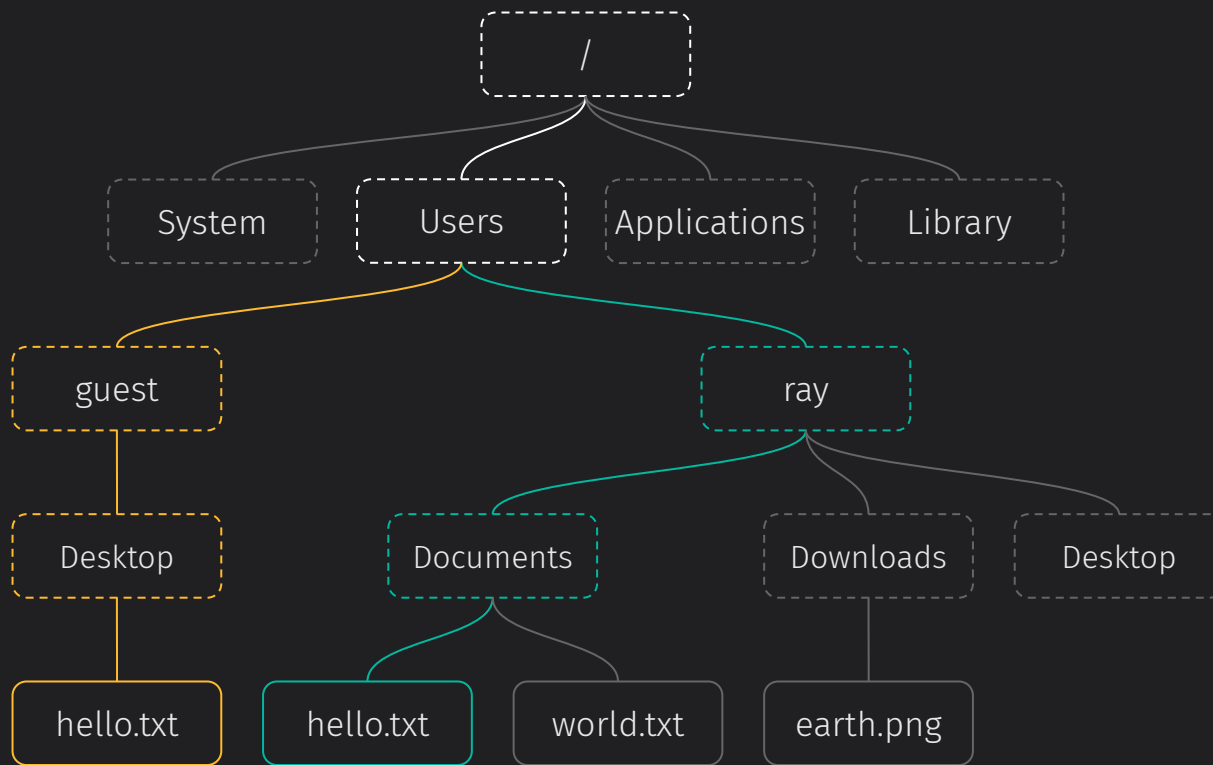


# Terminology

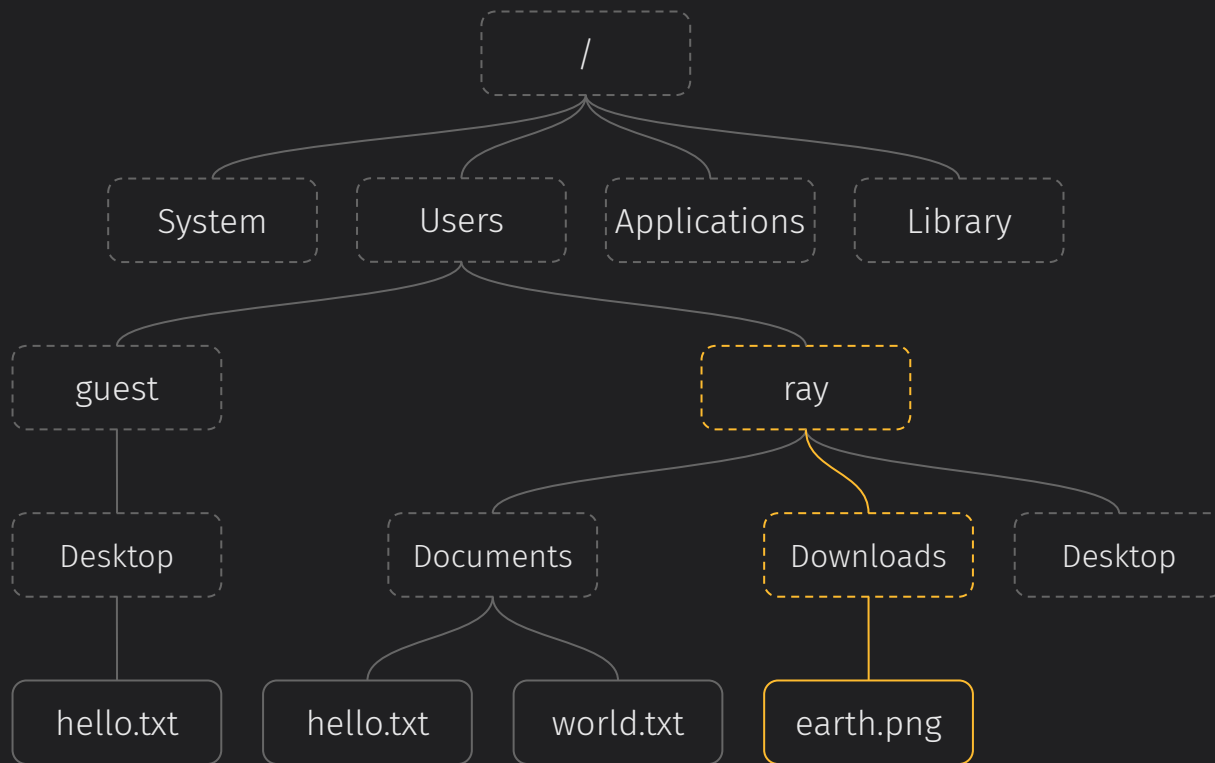
- File systems are **hierarchical tree** structures
  - Has a **root** node (`/` on \*nix or `C:\` on Windows)
  - Nodes may have children (**directories** or **folders**)
- A **path** is a location in the file system
  - Slash separates levels (`/` on \*nix or `\` on Windows)
  - May be **absolute** (starts with root) or **relative**



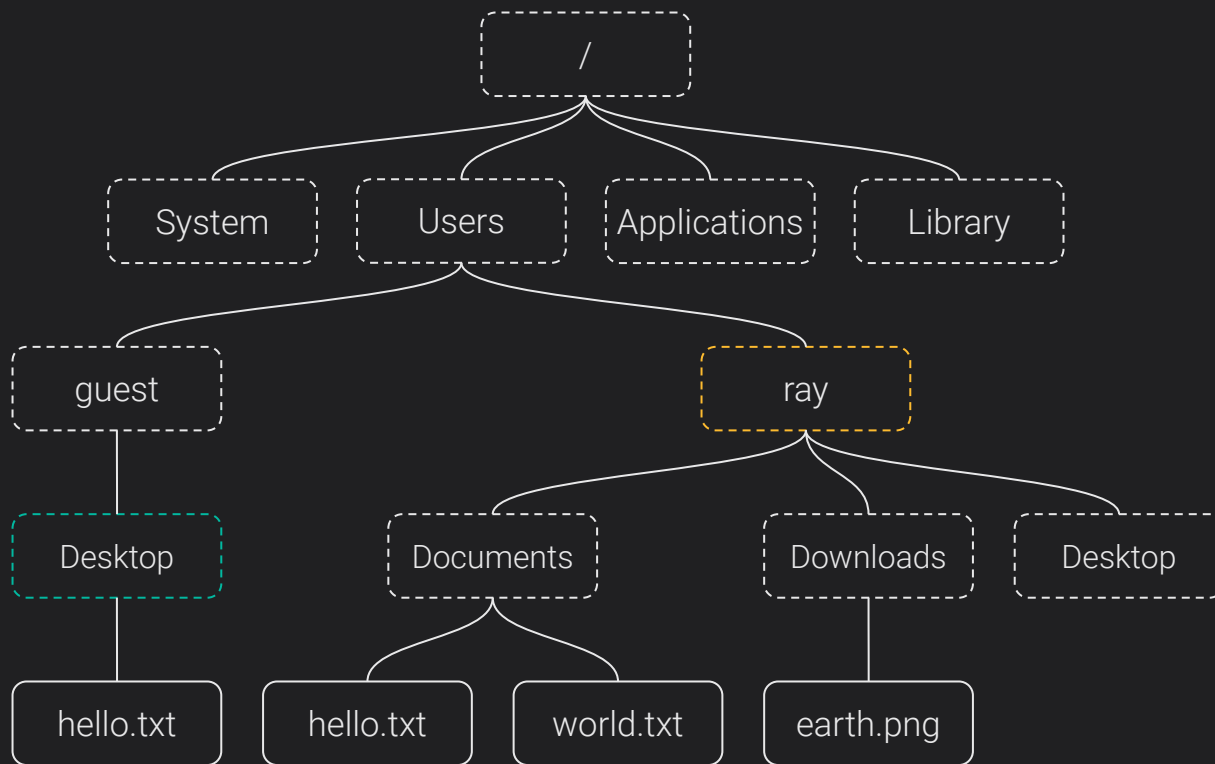




**/Users/guest/Desktop/hello.txt** vs **/Users/ray/Documents/hello.txt**



**~/Downloads/earth.png vs Downloads/earth.png vs /Users/ray/Downloads/earth.png**



**Desktop vs ./Desktop vs ../guest/Desktop**

# Java IO vs NIO.2

Package `java.io`

Package `java.nio.file`

Manipulating Paths





# Java IO vs NIO.2

## Package `java.io`

---

`java.io.File`

---

`File.getParent()`

---

`File.getAbsolutePath()`

---

`File.exists()`

---

`File.canRead()`

## Package `java.nio.file`

---

`java.io.Path`

---

`Path.getParent()`

---

`Path.toAbsolutePath()`

---

`Files.exists()`

---

`Files.isReadable()`

Manipulating Paths



# Java IO vs NIO.2

**Class** `java.io.File`

**Package** `java.nio.file`

---

---

---

---

Listing Directories



# Java IO vs NIO.2

**Class** `java.io.File`

---

`File.list()`

---

`File.listFiles()`

**Package** `java.nio.file`

---

`Files.list()*`

---

`Files.walk()*`

---

`Files.walkFileTree()`

---

`Files.newDirectoryStream()`

Listing Directories



# Java IO vs NIO.2

**Class** `java.io.File`

**Package** `java.nio.file`

---

---

---

---

Reading and Writing Files



# Java IO vs NIO.2

## Class `java.io.File`

---

```
BufferedReader in =  
    new BufferedReader(  
        new FileReader(  
            new File("hello.txt")));
```

## Package `java.nio.file`

---

```
Files.readString()
```

---

```
Files.lines()
```

---

```
Files.readAllLines()
```

---

```
Files.newBufferedReader()
```

Reading and Writing Files



# Replacements for `java.io.File`

- Use **Path** to represent and manipulate a location
- Use **Files** to learn more about what is at a **Path**
- Use **Files** to read or write small files
- Use **Files** to create **BufferedReader/Writer** and **DirectoryStream** objects for other operations



# Questions?

