PAU CASESNOVES
Centre Integrat de Formació Professional

Desenvolupament d'aplicacions multiplataforma

Accés a dades

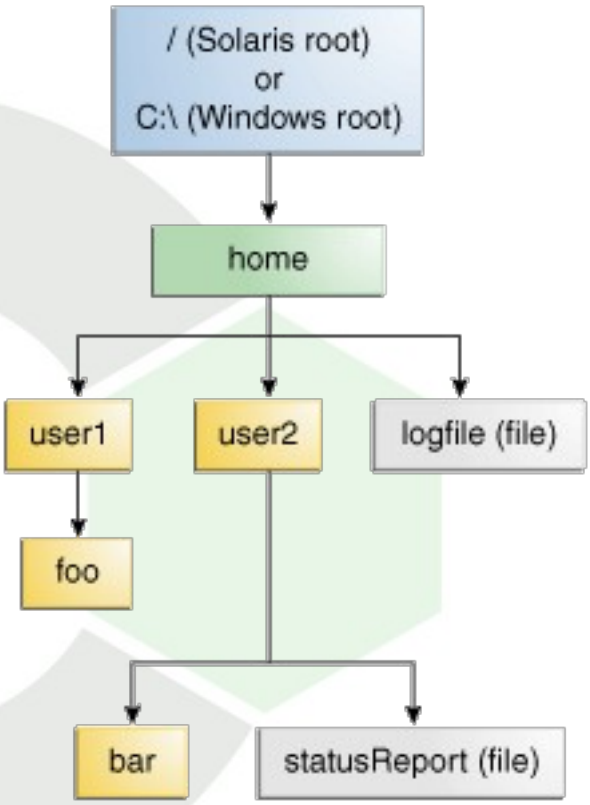
UT 2.1. Sistemes de fitxers.

Joan Pons Tugores

Índex de continguts

Sistemes de fitxers

El sistema de fitxers és un mètode per emmagatzemar i organitzar fitxers d'ordinador i les dades que contenen per tal de facilitar-ne la localització i accés.



Generalment un sistema d'arxius té directoris que associen noms d'arxius amb arxius, normalment connectant el nom d'arxiu a un índex en una taula d'assignació d'arxius d'algun tipus, com ara FAT en sistemes d'arxius MS-DOS o els inodes dels sistemes Unix.

En alguns sistemes de fitxers els noms d'arxius són estructurats, amb sintaxis especials per a extensions d'arxius i números de versió. En altres, els noms d'arxius són simplement cadenes de text i les metadades de cada arxiu són allotjades separadament.

Rutes

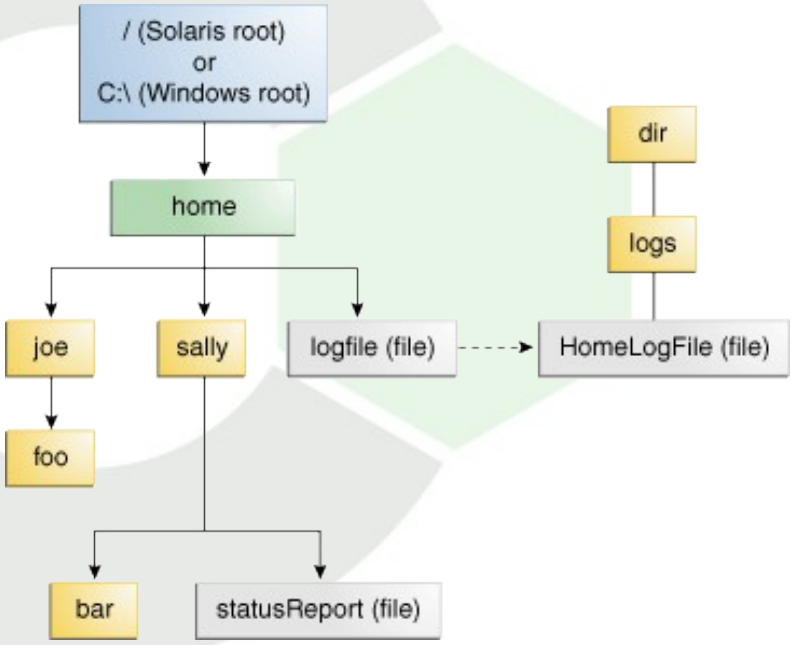
En sistemes d'arxius jeràrquics, normalment, es declara la ubicació precisa d'un arxiu amb una cadena de text anomenada "ruta". La nomenclatura per a rutes varia lleugerament de sistema en sistema, però mantenen en general una mateixa estructura. Una ruta ve donada per una successió de noms de directoris i subdirectoris, ordenats jeràrquicament d'esquerra a dreta i separats per algun caràcter especial que sol ser una barra ('/') o barra invertida ('\') i pot acabar amb el nom d'un arxiu present en l'última branca de directoris especificada.

Un fitxer s'identifica de forma única amb la seva ruta. Una ruta pot ser absoluta, quan conté l'arrel del sistema, o relativa quan conté només part de la ruta, o l'inici de la ruta no és l'arrel.

Per exemple:

En un sistema linux:

```
/home/sally/statusReport
```



Si comença amb / és una ruta absoluta a partir de l'arrel del sistema de fitxers.

En canvi, si el primer caràcter no és / es tracta d'una ruta relativa al directori on ens trobam actualment.

Per exemple, si estam dins la carpeta *home* la ruta relativa al mateix fitxer seria

```
sally/statusReport
```

En un sistema window, una ruta és absoluta si comença amb el nom de launitat seguida de : i \ Per el mateix fitxer, la ruta absoluta seria

```
c:\home\sally\statusReport
```

Les rutes relatives seran totes les que no incloguin el nom de la unitat.

```
\home\sally\statusReport  //relativa a l'arrel

sally\statusReport  //Si estam situats al directori home
```

Tant en sistemes linux com windows podem utilitzar els caràcters . i ..

- . significa el directori actual. Si estam a la carpeta *home*

```
./sally/statusReport
```

- .. significa el directori superior, el que inclou l'actual. Per exemple, si ens trobam al directori *joe*, per accedir al fitxer *statusReport* podem posar:

```
../sally/statusReport
```

Al sistema de fitxers podem trobar arxius especials que serveixen de referència a altres arxius, els enllaços simbòlics. Qualsevol operació feta sobre l'enllaç es porta a terme amb el fitxer o directori que enllaça, excepte l'esborrat o el canvi de nom. Normalment són transparents a l'usuari.

Comandes de consola bàsiques

Canviar de directori

cd (Change Directory) tant en sistemes linus com windows

```
cd /home/sally //ruta relativa

cd ./sally          //des de home

cd ..               //Va al directori home si estam dins el directori joe
```

Veure el contingut del directori actual

```
ls en sistemes linux

dir         en sistemes windows
```

La classe Path

Aquesta classe representa una ruta dins del sistema de fitxers. Pot fer referència a un arxiu, a un directori o no existir. Utilitza la notació pròpia del sistema de fitxers que estem utilitzant.

Crear un objecte de la classe Path

La forma més senzilla és utilitzant el mètode estàtic *of*. Rep com a paràmetre una cadena amb la ruta que volem utilitzar

```
Path p1 = Path.of("/home/sally/statusReport");

Path p2 = Path.of("home","sally","statusReport"); //Junta les cadenes per
                                                  formar la ruta.

Path p3 = Path.of(URI.create("file:///home/sally/statusReport"));
```

Per a cada nivell de la ruta guarda un element que el representa.

Recuperar informació del Path

Si tenim el path */home/joe/foo* llavors

- **.toString()** torna *"/home/joe/foo"*
- **.getNameCount()** torna quants elements hi ha al path.
- **.getName(index)** torna el *Path* de l'element de la ruta d'aquesta posició. getName(0) torna *home*.
- **.getFileName()** torna el Path que representa el darrer element de la ruta, tant si és un fitxer com un directori. En aquest cas *foo*

- **.subpath(inici, fi)** torna el path entre la posició inici i la fi. SubPath(0,2) torna home/joe

La classe *Files*

És una classe d'utilitat (una classe abstracte amb mètodes de classe) que ens permet llegir, escriure i manipular fitxers i directoris.

Consultes

- **Files.isRegularFile(path)**: *true* si el path que rep com a argument és un fitxer, *false* si és un directori o un enllaç.
- **Files.isDirectory(path)**: *true* si el path que rep com a argument és un directori.
- **Files.isReadable(path)**
- **Files.isWriteable(path)**
- **Files.isExecutable(path)**
- **Files.isSameFile(path1, path2)**: *true* si els dos paths que rep com a arguments representen el mateix objecte.
- **Files.size(path)**: mida del fitxer en bytes.
- **Files.isHidden(path)**
- **Files.get/setLastModifiedTime(path)**
- **Files.get/setOwner(path)**

Crear directori

- **Files.createDirectory(Path)**: Crea el directori definit per el paràmetre. La ruta on volem crear el directori ha d'existir. És a dir si volem crear /home/joe/proves la ruta /home/joe ha d'existir

- **Files.createDirectories(Path)**: Crea el directori definit per el paràmetre. Si la ruta on volem crear el directori no existeix la crea. És a dir si volem crear /home/joe/proves i no existeix el directori *joe* també el crea.

Esborrar fitxers o directoris:

- **Files.delete(Path)**: Esborra l'objecte representat per el path.

Copiar fitxers o directoris:

- **Files.copy(origen, destí, opcions):** Origen i destí són *Paths* i *opcions* és un vararg per a les constants StandardCopyOption.REPLACE_EXISTING, COPY_ATTRIBUTES, LinkOption.NOFOLLOW_LINKS.

Moure fitxers o directoris:

- **Files.move(origen, destí, opcions)**: Origen i destí són *Paths* i *opcions* és un vararg per a les constants StandardCopyOption.REPLACE_EXISTING, StandardCopyOption.ATOMIC_MOVE.

Llegir i escriure continguts del fitxer:

Aquest mètodes ens serviran quan necessitem manipular fitxers senzills. Per altres casos serà molt millor utilitzar els streams que veurem pròximament.

- **Files.write(Path, bytes[], opcions)**: Escriu el contingut de l'array de bytes al fitxer. Dins les opcions podem decidir com s'obre el fitxer:
  - StandardOpenOption.WRITE: Obre el fitxer per escriptura.
  - StandardOpenOption.READ: Obre el fitxer per lectura.
  - StandardOpenOption.CREATE: L'obre si existeix i si no el crea.
  - StandardOpenOption.APPEND: Juntament amb els anteriors l'obre al final per afegir dades.

  El següent codi obre per escriptura el fitxer /home/joe/doi.txt Si no existeix el crea. Afegeix els bytes de la cadena de text al final del fitxer.

```
Files.write(Path.of("/home/joe/doi.txt"),"Aixo era i no era".getBytes(StandardCharsets.UTF_8), StandardOpenOption.CREATE, StandardOpenOption.APPEND);
```

- **Files.readAllBytes(path)**: Llegeix el contingut del fitxer com a bytes. Torna byte[].

```
try {
    byte[] bytes = Files.readAllBytes(Path.of(fitxers.toString() + "/doi.txt"));
    System.out.println(Arrays.toString(bytes));
} catch (IOException e) {
     System.out.println("Error:"+e.getMessage());
}
```

Si llegim el fitxer escrit anteriorment veurem per pantalla:

```
[65, 105, 120, 111, 32, 101, 114, 97, 32, 105, 32, 110, 111, 32, 101, 114, 97]
```

- **Files.readAllLines(Path)**:

```
try {
        List<String> linees = Files.readAllLines(Path.of(fitxers.toString() +
"/doi.txt"));
        System.out.println(linees);
} catch (IOException e) {
        System.out.println("Error:" + e.getMessage());
}
```

Si llegim el fitxer escrit anteriorment veurem per pantalla

```
[Això era i no era]
```

Aquest mètode està sobreescrit amb una versió que accepta com a paràmetre el joc de caràcters en el que està escrit el text.

Per exemple, si volem especificar que el fitxer conté caràcters en UTF-8:

```
List<String> linies = Files.readAllLines(Path.of(fitxers.toString() + "/doi.txt"),
StandardCharsets.UTF_8);
```

O si està escrit en el joc de caracters que utilitza normalment windows:

```
List<String> linies = Files.readAllLines(Path.of(fitxers.toString() + "/doi.txt"),
StandardCharsets.ISO_8859_1);
```

Llegir el contingut d'un directori:

- **Files.newDirectoryStream()**: Torna un stream on cada element és un path de dins del directory.

```
Path dir = ...;

try (DirectoryStream<Path> stream = Files.newDirectoryStream(dir)) {

    for (Path file: stream) {

        System.out.println(file.getFileName());

    }

}
```