Complementi di Programmazione

File I/O

CdL Informatica - Università degli studi di Modena e Reggio Emilia
AA 2023/2024

Filippo Muzzini

File I/O

Sulle unita' di memorizzazione (dischi/chiavette etc) i dati sono registrati in blocchi contenenti sequenze di lunghezza fissa composte da byte.

E' una componente del Sistema Operativo che fornisce una “visione” comoda per accedere ai dati, ovvero una visione a file, contenuti nelle directory.

File I/O

Ogni file ha un nome (o piu' precisamente almeno un nome).

Per individuare un file nel file system occorre un pathname (nome del percorso)

A partire dalla radice si possono nominare tutte le directory da attraversare per “raggiungere” il file.

File I/O

Per evitare di specificare sempre il percorso a partire dalla radice ogni processo ha una directory di lavoro.

Di norma un processo eredita la stessa directory di lavoro del processo che l'ha attivato.

File I/O

Per aprire un file si usa l'istruzione “open”, per chiuderlo si usa il metodo “close”.

Open ha due parametri, il primo e' il pathname del file, il secondo e' il modo di apertura:

- 'r' per leggere
- 'w' per scrivere (azzera il contenuto del file)
- ‘a’ per aggiungere ad un file già presente
- ‘b’ (da aggiungere ad uno dei precedenti) per aprire il file in modalità binaria

File I/O

Si apre il file cosi', il valore di uscita della open e' il descrittore del file da usare per individuare il file aperto.

– fd=open(“miofile”,”r”)

Si puo' leggere l'intero file in una stringa:

– filecontents=fd.read()

Si puo' leggere una riga (e poi le successive con identico metodo)

– fileline=fd.readline()

Si puo' fare un ciclo usando il file come sequenza, a ogni iterazione “line” sara' istanziato ad una riga del file

– for line in fd:

Alla fine il file si chiude cosi':

– fd.close()

File I/O

Si apre il file cosi', il valore di uscita della open e' il descrittore del file da usare per individuare il file aperto.

– fd=open(“miofile”,”w”)

Si puo' scrivere una stringa:

– fd.write(s)

Si puo' usare il parametro file= della print (solo in Python3)

– print(“hello world”, file=fd)

Alla fine il file si chiude cosi':

– fd.close()

File I/O

In modalità binaria si salvano bytes grezzi. E’ compito del programmatore saperli poi interpretare:

```
prova = open('prova.bin', 'wb')

prova.write(bytes([1, 2, 3]))

prova.close()

prova2 = open('prova.bin', 'rb')

a = prova2.read()

a = list(a)
```

File I/O

L'istruzione with consente una sintassi alternativa.

with open(“file”,”r”) as fd:

    ....

Nel blocco dipendente dall'istruzione “with”, fd e' il descrittore del file aperto. Se avviene un errore il blocco non viene eseguito e comunque il file viene chiuso alla fine del blocco.

Pickle

Pickle è un modulo che permette di salvare oggetti arbitrari su file.

Serializza il dato.

La serializzazione è molto comune per trasmettere e salvare dati.

Pickle

pickle.dump(obj, file) -> salva l’oggetto su file

pickle.dumps(obj) -> restituisce la serializzazione come stringa.

pickle.load(file) -> legge il file e restituisce l’oggetto

pickle.loads(data) -> legge data (stringa) e restituisce l’oggetto

Pickle

Cosa può essere salvato:

- built-in;
- integers, floating-point numbers, complex numbers;
- strings, bytes, bytearrays;
- tuples, lists, sets, and dictionaries contenenti oggetti salvabili;
- funzioni (no `lambda`);
- classi
- istanze che abbiano il metodo __getstate__ salvabile (di default ritorna il __dict__) (utile per salvare solo alcuni campi)

Notare che funzioni e classi vengono salvate utilizzando solo il loro nome (non il codice per motivi di sicurezza). Questo implica che quando vengono deserializzate è necessario importarle preventivamente.

Pickle

Pickle può essere pericoloso!

Potreste deserializzare un oggetto che contiene codice malevolo.

Deserializzare solo oggetti di cui vi fidate!