Cours 5 :

Fichiers

M. BONNEFOI
2014 / 2015

Plan

Introduction :
- Concept : (rappel) Le système de Fichier
- Concept : (rappel) Stockage de masse et arborescence de fichier

Répertoires :
- Concept : API
- Technique : fonctions liés aux répertoires

Fichier :
- Concept : (rappel) Fichier texte / binaire
- Technique : Lecture / Ecriture de fichiers

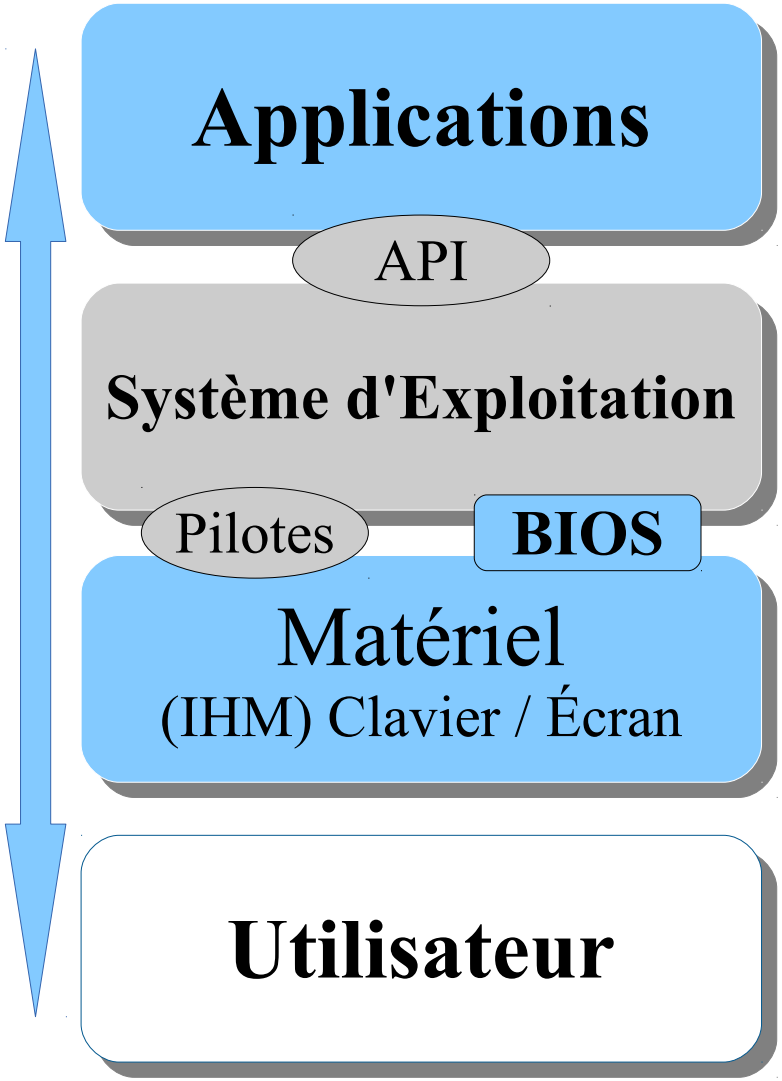
Chaînes de caractères :
- Concept : (rappel) les conteneurs iterateurs
- Technique : Fonctions (méthodes) des chaines de caractères

Systèmes de fichiers : les supports

Mémoire persistante : Périphérique de stockage de masse

- Disque Dur
- Clés USB
- cartes SD etc.
- CD/DVD
- Bandes magnétiques
- etc...

Système d'exploitation



Exemple d'interaction :

1- utilisateur interagit avec le matériel
(ex : appuie sur une touche du clavier)
2- le matériel génère un signal
3- le signal est transmis au pilote
4- le pilote transmet l'info au S.E.

5- le SE transmet l'info à une application
6- l'appli. modifie son état et l'indique au S.E.
(ex : telle image dans telle fenêtre/pixels)

7- le S.E. communique la nouvelle info au driver
8- le driver communique l'info au matériel
9- le matériel modifie sont état
(ex:affichage de la nouvelle image)

Le système de fichiers

Système de fichiers :

Un système de fichiers permet d'organiser la lecture et l'écriture de données sur un périphérique de stockage de masse (comme un disque dur) sous forme de fichiers rangés dans des répertoires.

Il transforme de la mémoire brute (une suite de bits) en une arborescence de fichiers et répertoires.

Ils permettent aussi :

- la gestion des droits entre utilisateurs (qui est propriétaire, qui a le droit d'écrire etc.)
- la cryptographie
- la journalisation (un historique de modifications)
- des liens symboliques ou pas entres fichiers (un même fichier à plusieurs endroits) ('raccourcis')

Ils peuvent aussi être réseau ou virtuel.

Ex : Microsoft : FAT/FAT32 – NTFS
Mac : HFS/HFS+
Unix/Linux : ExtFS – ReiserFS – ZFS...

2 Parties :

- La structure : La table d'allocation des fichiers qui décrit l'arborescence est en début de partition,
- Les données : L'ensemble des fichiers et de leur contenu.

→ possibilité d'effacer l'arborescence ou de retrouver un fichier effacé

Répertoire

Répertoires et chemins

Deux manière d'indiquer l'adresse d'un fichier dans un système de fichier
chemin absolu (démarre depuis la racine du S.F. indépendant de l'emplacement courant)
chemin relatif (relatif au répertoire actuel)

Exemple de chemin absolu :
Windows :

C:\Python34\Lib\idlelib\idle.bat
L:\IN14\TP5\

Unix - Linux – MacOS - Android etc.

/User/bonnefoi/Documents/IPSA/IN14
/Volumes/BOOTCAMP/

Exemple de chemin relatifs :

. → répertoire courant
.. → répertoire parent
../../tpsrc/ → remonte de 2 répertoires puis entre dans tpsrc/

Répertoires : api

Le module os, interface de programmation du S.E.
(in english : OS API)

```
import os
```

`os.curdir`	→ indique le répertoire courant
`os.path.abspath(path)`	→ donne le chemin absolu de path (str)
`os.path.basename(path)`	→ donne le nom sans le chemin (str)
`os.listdir(path)`	→ liste le contenu d'un répertoire (liste)
`os.chdir(path)`	→ change le répertoire de travail
`os.mkdir(path)`	→ créer un répertoire
`os.path.isfile(path)`	→ vrai si path est un fichier
`os.path.isdir(path)`	→ vrai si path est un répertoire

path → chaîne de caractère contenant le chemin d'un fichier ou d'un répertoire

Répertoires - exemple

Le module os est l'API avec l'os :

```
print("Répertoire courant: ", os.curdir)
print("Répertoire courant: ", os.path.abspath('.'))
print("Liste des fichiers :", os.listdir())
print("Changement de répertoire")
os.chdir("./TP05src/tests/")    # mac linux android
os.chdir(".\\TP05src\\tests\\") # windows
print("Répertoire courant: ",
os.path.basename(os.path.abspath(os.curdir)))
```

```
Répertoire courant:  .
Répertoire courant:  /Users/fabien/Documents/Trabajo/2014_IPSA/
2014/IPSA_2014_IN1/IN14/S05_Fichiers/TPsrc
Liste des fichiers : ['.DS_Store', 'F1_openShow.py',
'gpstrace1.json', 'gpstrace2.json', 'gpstrace3.json',
'JsonToCSV.py', 'tests', 'TP5_p1.py', 'TP5_p2.py']
Changement de répertoire
Répertoire courant:  tests
```

...

Fichiers

Les Fichiers

2 catégories :

- les fichiers 'textes' → utilisent un codage de caractères
- les fichiers 'binaires' → utilisent un codage spécifique

Un fichier Texte :

- peut servir à stoker des documents, codes sources de programme, scripts, fichiers de configuration, dessins vectoriels, stockage de données, etc.
- manipulable avec un simple 'éditeur de texte'
- exemples :
  .txt, .csv, .html .php .js .css .svg, .py, .c .cpp .java ……

Les 'fichiers binaires' pour tout usage :

Quelques exemples :

Fichiers image : .bmp .jpg .gif .png ...
Fichiers audio : .wav .mp3 .acc .ogg ...
Fichiers video : .avi .mov . mp4 .mkv …
Fichiers exécutables : .exe .app
Fichier compressés : .zip .rar .part .7z ...
Fichiers documents : .doc, .docx, .odt, .pdf ...

Les Fichiers

Fichier 'Texte' :

Un fichier texte est constitué d'une suite de caractères codés suivant une norme de codage de caractères qui décrit les codes, sur plusieurs bits de données, de chacun des caractères.

Codages (ou tables) de caractères 'standards' :

ASCII (ansi): 128 caractères → 7bits (std depuis 1960)
chiffres, lettres, majuscules, ponctuation et 30 commandes (retour à la ligne, bip, etc.)
Iso-Latin1 (ou ISO-8859-1): (1986) 8 bits – 1octets – 256 caractères
Introduction entre autre des accents
UTF8 : (années 2000) 1 à 4 octets →> 1million de caractères
Internationalisation

127 premiers → toujours ASCII

NB : Ces tables de caractères sont très utilisées dans les protocoles réseaux
(ex : pages web. : http, mail : smtp, configuration réseau : snmp..)(cf.cours 3)

Table ASCII

Contrôles :

CR : Carriage Return
LF : Line Feed
BEL : Bell
ESC : Escape
Del : Delete
...

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Les Fichiers : 'Binaires'

Fichiers 'Binaires' :

Les fichiers dits binaires nécessitent une application spécifique et adaptée pour être créés modifiés ou lus. Ils utilisent un système de codage approprié au domaine applicatif.

L'extension de fichier :

L'extension d'un fichier est constituée des dernières lettres de son nom qui sont précédées d'un point (par ex : .txt). Elle permet d'identifier le type d'un fichier et la nature des données qu'il contient.

Fichiers : Fonction open()

La fonction open() permet d'ouvrir un fichier, étape indispensable pour lire ou écrire dedans.
https://docs.python.org/3/library/functions.html#open

Prototype : F = open(file, mode='r', encoding=None)

file : le chemin du fichier par exemple
"L:\IN14\TP05\test.txt"
"test.txt"

mode : le mode d'ouverture (lecture /écriture / les 2)
'r' open for reading (default)
'w' open for writing, truncating the file first
'a' open for writing,
'r+' ou 'w+' open for read and write

encoding : l'encodage (ascii, iso-Latin-9, utf8 etc.)
Liste des encodages : (*https://docs.python.org/3/library/codecs.html#standard-encodings*)

Fichiers : classe TextIOWrapper

F = open(file, mode='r', encoding=None)

Type de donnée obtenu (type de la variable F) :

→ objet de type (classe) TextIOWrapper (et donc sous type de TextIOBase et IOBase)

(nb : to Wrap → emballer, empaqueter, entortiller, envelopper...)

Méthodes liées :

F.close() → ferme le fichier

F.readline() → lit une ligne (str) (et déplace la tête de lecture)
F.readlines() → liste des lignes (et déplace la tête de lecture)

F.write(str) → écrit str (et déplace la tête de lecture)
F.writelines(list) → écrit la liste de chaînes de caractères (et déplace la tête de lecture)

F.seek(int) → déplace la tête de lecture
F.tell() → (int) position de la tête de lecture

Fichiers : Exemple

Un exemple de read / write

```
# Lecture
fname = "demo.txt"
Fichier = open(fname,'r')
for l in Fichier.readlines():
    print(l,end='')
Fichier.close()


# Écriture (en mode ajout)
fout = "out.txt"
Fout = open(fout,'a')
Fout.write(" - pi : " + str(math.pi)+"\n")
Fout.close()
```

Chaine de Caractère

Objet de type str (classe str)
Exemple : S = "BONNEFOI"

Recherche :
- S.count(str) → compte le nombre d'occurrence de str dans S
- S.find(str) → donne l'index de la première occurence de str
- S.rfind(str) → donne l'index de la dernière occurrence de str

Découpage / Collage
- split(str) → découpe la chaine suivant le séparateur str
- join(list) → concatène les chaines contenues dans list

Rappels : Ajout / Accès (cf. conteneurs iterables)
- [0]-> première case [-1]denière [deb:fin]-> sous chaine
- len(), append(), insert(), extend(), pop(), remove()

Chaine de Caractère : Exemple

```
#Conversion d'une chaîne de caractères en liste

>>> a = " tralala, toto, blabliblablou, …, 125"
>>> l = a.split(',')
>>> print(l)
[' tralala', ' toto', ' blabliblablou', ' …', '
125']

>>> print(l[2])
 Blabliblablou

#Conversion d'une liste en chaîne de caractères

>>> b =  "," .join(l)
>>> print(b)
 tralala toto blabliblablou … 125
```

Sources & ressources

Documentation sur les fichiers

(type class "io.IOBase" et "io.TextIOBase") :

https://docs.python.org/3/library/io.html?highlight=readlines#io.IOBase.readlines

Supports de cours en ligne :

Docs I.P.S.A.

https://docs.ipsa.fr/

In 12 - Introduction à la programmation avec Python

IonisX

→ cours et Exo

https://ionisx.com/courses

→ Python pour les scientifiques