|  |  |  |
| --- | --- | --- |
| Résultat de recherche d'images pour "université abdelmalek essaadi" | Université Abdelmalek Essaadi  Ecole nationale des sciences appliquées Tanger |  |

Master Cybersécurité et Cybercriminalité

2021 – 2022



Rapport du devoir libre

|  |  |  |
| --- | --- | --- |
|  | Appels système en Python |  |

Matière : Programmation système

Rendu le 20/02/2022

|  |  |
| --- | --- |
| Réalisé par :  KHALDOUN Mohamed Amin | Encadré Par :  Mr. AMECHNOUE Khalid |

Table des matières

[Introduction 3](#_Toc91715324)

[1- Le code convolutif systématique récursive : 4](#_Toc91715325)

[Les Codes convolutifs : 4](#_Toc91715326)

[Les Codes Systématiques Récursifs (CSR) : 5](#_Toc91715327)

[2- Turbo codes : 6](#_Toc91715328)

[Turbo-code série : 6](#_Toc91715329)

[Turbo-code parallèle 6](#_Toc91715330)

[L'encodeur Turbo 6](#_Toc91715331)

[Décodage itératif 7](#_Toc91715332)

[Le décodage d’un turbo-code 7](#_Toc91715333)

[3- Implémentation : 9](#_Toc91715334)

[Conclusion 10](#_Toc91715335)

[Références 10](#_Toc91715336)

# Introduction

# Environnements de travail :

## PyCharm (2021.3 Comunity Edition) :

|  |  |
| --- | --- |
| upload.wikimedia.org/wikipedia/commons/thumb/1/... | PyCharm est un environnement de développement intégré utilisé pour programmer en Python.  Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions.  Dans le travail présent, je m’en suis servi pour éditer et exécuter les différents composants du projet. |

## Qt Designer (5.11.1):

|  |  |
| --- | --- |
| Qt — Wikipédia | Qt Designer est un logiciel qui permet de créer des interfaces graphiques Qt dans un environnement convivial. L'utilisateur, par glisser-déposer, place les composants d'interface graphique et y règle leurs propriétés facilement. Les fichiers d'interface graphique sont formatés en XML et portent l'extension .ui |

Lors de la compilation, un fichier d'interface graphique est converti en classe Python par l'utilitaire pyuic5.

# Appels système :

## Module OS :

Le module OS en Python fournit des fonctions d’interaction avec le système d’exploitation. Le système d’exploitation relève des modules utilitaires standard de Python. Ce module fournit un moyen portable d’utiliser les fonctionnalités dépendant du système d’exploitation.

## os.fork :

La méthode os.fork() en Python est utilisée pour créer un processus enfant. Cette méthode fonctionne en appelant la fonction sous-jacente du système d’exploitation fork(). Cette méthode renvoie 0 dans le processus enfant et l’ID de processus enfant dans le processus parent.

|  |  |
| --- | --- |
| Code Pthyon | Sortie en console |
| import os  pid = os.fork()  if pid > 0:  print("I am parent process:")  print("Process ID:", os.getpid())  print("Child's process ID:", pid)  else:  print("\nI am child process:")  print("Process ID:", os.getpid())  print("Parent's process ID:", os.getppid()) | I am parent process:  Process ID: 20900  Child's process ID: 20901  I am child process:  Process ID: 20901  Parent's process ID: 20900 |

## os.system :

Exécute une commande (String) dans un sous-shell. Ceci est implémenté en appelant la fonction standard C system() et a les mêmes limitations. Les modifications apportées à sys.stdin, etc. ne sont pas reflétées dans l'environnement de la commande exécutée. Si la commande génère une sortie, elle sera envoyée au flux de sortie standard de l'interpréteur. La norme C ne spécifie pas la signification de la valeur de retour de la fonction C, donc la valeur de retour de la fonction Python dépend du système.

|  |  |
| --- | --- |
| Code Pthyon | Sortie en console |
| import os  os.system("date") | Sun 20 Feb 2022 09:44:56 AM EST |

## os.exec\* :

Les méthodes os.exec\* exécutent toutes un nouveau programme, remplaçant le processus actuel, elles ne renvoient pas. Sur Unix, le nouvel exécutable est chargé dans le processus actuel, et aura le même identifiant de processus (PID) que l'appelant.

Les variantes « l » et « v » des fonctions exec\* différent sur la manière de passer les arguments de ligne de commande. Les variantes « l » (execl(), execlp(), execle() et execlpe()) sont probablement les plus simples à utiliser si le nombre de paramètres est fixé lors de l'écriture du code. Les paramètres individuels deviennent alors des paramètres additionnels aux fonctions exec\*(). Les variantes « v » (execv(), execvp(), execve() et execvpe()) sont préférables quand le nombre de paramètres est variable et qu'ils sont passés dans une liste ou un tuple dans le paramètre args. Dans tous les cas, les arguments aux processus fils devraient commencer avec le nom de la commande à lancer, mais ce n'est pas obligatoire.

Les variantes qui incluent un « p » vers la fin (execlp(), execlpe(), execvp(), et execvpe()) utiliseront la variable d'environnement PATH pour localiser le programme file. Quand l'environnement est remplacé, le nouvel environnement est utilisé comme source de la variable d'environnement PATH. Les autres variantes execl(), execle(), execv(), et execve() n'utiliseront pas la variable d'environnement PATH pour localiser l'exécutable. path doit contenir un chemin absolue ou relatif approprié.

Pour les fonctions execle(), execlpe(), execve(), et execvpe(), le paramètre env doit être un mapping qui est utilisé pour définir les variables d'environnement du nouveau processus. Les fonctions execl(), execlp(), execv(), et execvp() causent toutes un héritage de l'environnement du processus actuel par le processus fils.

|  |  |
| --- | --- |
| Code Pthyon | Sortie en console |
| import os  os.execl("/bin/date", "date") | Sun 20 Feb 2022 10:24:23 AM EST |
| os.execlp("date", "date") | Sun 20 Feb 2022 10:26:52 AM EST |
| cmd = ["ls", "-l", "-n"]  os.execvp("ls", cmd) | -rw-r--r-- 1 1000 1000 99 Feb 20 10:29 Execvp.py  -rw-r--r-- 1 1000 1000 99 Feb 20 10:26 Execlp.py  -rw-r--r-- 1 1000 1000 44 Feb 20 10:24 Execl.py |

## os.pipe :

Un tube est une méthode pour transmettre des informations d’un processus à un autre processus. Il n’offre qu’une communication unidirectionnelle et les informations transmises sont conservées par le système jusqu’à ce qu’elles soient lues par le processus de réception. La méthode os.pipe() en Python est utilisée pour créer un tube.

|  |  |
| --- | --- |
| Code Pthyon | Sortie en console |
| import os  r, w = os.pipe()  pid = os.fork()  if pid > 0:  os.close(r)  print("Parent process is writing")  text = b"Hello child process"  os.write(w, text)  print("Written text:", text.decode())  else:  os.close(w)  print("Child Process is reading")  r = os.fdopen(r)  print("Read text:", r.read()) | Parent process is writing  Written text: Hello child process  Child Process is reading  Read text: Hello child process |

## os.dup / os.dup2 :

Un descripteur de fichier est une petite valeur entière qui correspond à un fichier ou à une autre ressource d’entrée / sortie, telle qu’un canal ou une socket réseau. Un descripteur de fichier est un indicateur abstrait d’une ressource et agit comme un descripteur pour effectuer diverses opérations d’E / S de niveau inférieur telles que la lecture, l’écriture, l’envoi, etc.

**Par exemple:** l’entrée standard est généralement un descripteur de fichier avec la valeur 0, la sortie standard est généralement un descripteur de fichier avec la valeur 1 et l’erreur standard est généralement un descripteur de fichier avec la valeur 2. Les autres fichiers ouverts par le processus actuel auront la valeur 3, 4, 5 an bientôt.

La méthode os.dup() en Python est utilisée pour dupliquer le descripteur de fichier donné. Le descripteur de fichier dupliqué n’est pas héritable, mais sur la plate-forme Windows, descripteur de fichier associé au flux standard (entrée standard: 0, sortie standard: 1, erreur standard: 2) qui peut être héritée par les processus enfants.

La méthode os.dup2() en Python est utilisée pour dupliquer un descripteur de fichier fd à une valeur donnée fd2. Le descripteur de fichier sera dupliqué dans fd2 uniquement si fd2 est disponible et si le descripteur de fichier dupliqué est héritable par défaut.

Un descripteur de fichier héritable signifie que si le processus parent a un descripteur de fichier 4 utilisé pour un fichier particulier et que le parent crée un processus enfant, le processus enfant aura également un descripteur de fichier 4 utilisé pour ce même fichier.

|  |  |
| --- | --- |
| Code Pthyon | Sortie en console |
| import os  fd = os.open("Files/test.txt", os.O\_WRONLY)  print("Original file descriptor:", fd)  dup\_fd = os.dup(fd)  print("Duplicated file descriptor:", dup\_fd)  pid = os.getpid()  os.system("ls -l /proc/%s/fd" %pid)  os.close(fd)  os.close(dup\_fd)  print("File descriptor duplicated successfully") | Original file descriptor: 3  Duplicated file descriptor: 4  total 0  lr-x------ 1 kali kali 64 Feb 22 11:49 0 -> pipe:[312648]  l-wx------ 1 kali kali 64 Feb 22 11:49 1 -> pipe:[312649]  l-wx------ 1 kali kali 64 Feb 22 11:49 2 -> pipe:[312650]  l-wx------ 1 kali kali 64 Feb 22 11:49 3 -> /home/kali/PycharmProjects/ProjetProgSys/Files/test.txt  l-wx------ 1 kali kali 64 Feb 22 11:49 4 -> /home/kali/PycharmProjects/ProjetProgSys/Files/test.txt  File descriptor duplicated successfully |

# Manipulation de fichiers :

# 

# Conclusion

**Exercice d’application :**

Représenter sur le cercle trigonométrique les points d’abscisses curvilignes suivants :