|  |
| --- |
| **الجمهورية العربية السورية** |
| **اللاذقية - جامعـــة تشريـــــن** |
| **كلية الهندسة الكهربائية والميكانيكية**  **قسم هندسة الاتصالات والالكترونيات**  **السنة الخامسة : مشروع برمجة شبكات** |

**حلقة بحث بإعداد طلاب السنة الخامسة في كلية الهمك اختصاص هندسة الاتصالات و الإلكترونيات :**

**رغد بسام حسن 2573 يارا رفيق صقور 2318 منار محسن منصور2278**

**الملخص :**

**TCPهو اختصار ل (Transmission Control Protocol )بروتوكول التحكم بالنقل أو بروتوكول التحكم بالإرسال, وهو أحد البروتوكولات الأساسيّة في حزمة بروتوكولات الانترنت ويُؤمّن نقلاً موثوقا خاليًا من الأخطاء لدفق من البايتات بين مضيفين يتصلان مع بعضهما البعض عبرشبكة تدعم بروتوكول الانترنت .**

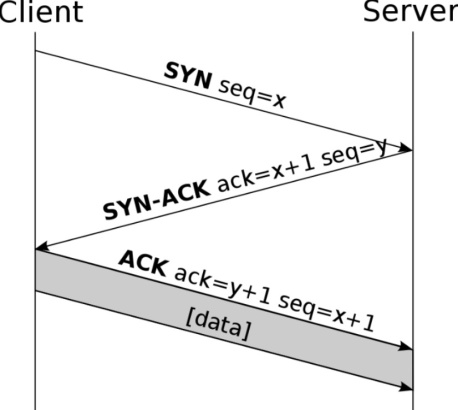
**في برنامجنا اعتمادا على بروتوكول TCP سنقوم بتنفيذ غرفة دردشة بلغة ا ل Python ب خادم واحد يستضيف الدردشة والعديد من العملاء الذين يتصلون به ويتواصلون مع بعضهم البعض في النهاية .**

**المقدمة :**

* **أولاً أهمية استخدام بروتوكولات الاتصال :**
* **يشير بروتوكول الاتصال المعروف أيضًا باسم بروتوكول الإرسال في الاتصالات السلكية واللاسلكية إلى معيار النظام الذي يسمح لطرفين أو أكثر في نظام الإرسال بنقل المعلومات في أي وسط مادي .**
* **بروتوكول الاتصال يشير إلى الاستخدام الشائع لاتصالات الكمبيوتر أومعدات الشبكة .**
* **يقوم باكتشاف الأخطاء المحتملة وتصحيحها و يمكن تنفيذ بروتوكول الاتصال في الأجهزة أو البرامج أو كليهما**

**نظرا لأهمية استخدام بروتوكولات الاتصال في حياتنا ولأن من أهم هذه البروتوكولات هو بروتوكول النقل TCP قررنا أن نناقش في حلقتنا أكثر ما يجعل هذا البروتوكول ذو أهمية بالغة .**

* **ثانياً**
* **ما هو ا ل TCP ( بروتوكول التحكم بالنقل أو التحكم بالإرسال ) :**

****

**هو بروتوكول شبكة هام  يستخدم في إرسال البيانات عبر الشبكات , وهو مجموعة من القواعد والإجراءات التي تحكم كيفية تنفيذ نقل البيانات بحيث يقوم كل شخص في العالم بغض النظر عن الموقع أو البرامج أو الأجهزة المستخدمة بالشيء نفسه.**

**الشكل (1)**

* **ما الذي يقوم به بروتوكول التحكم بالنقل TCP :**

**تتمثل وظيفة بروتوكول التحكم بالإرسال TCP في التحكم بنقل البيانات بحيث يمكن الاعتماد عليها فمثلاً في شبكات الإنترنت، يتم إرسال البيانات في حزم والتي هي وحدات من البيانات التي يتم إرسالها بشكل مستقل على الشبكة، ويتم إعادة تجميعها بمجرد وصولها إلى الوجهة لتشكيل البيانات الأصلية.**

## كيفية تحكم بروتوكول TCPبالنقل :

**يتعامل بروتوكول ال TCP على الجانب الأول (المصدر)مع العمليات الخاصة بالمستخدمين في طبقة التطبيقات , وعلى الجانب الآخر (الوجهة) بحيث يتعامل مع بروتوكول من المستوى السفلي مثل بروتوكول الانترنت (IP)**

**يمكن تقسيم خطوات عمل بروتوكول التحكم بالنقل لثلاث مراحل أساسية تتوضح بالشكل (1):**

* **يقوم الجهاز المصدر بإرسال رسالة إلى الجهاز الوجهة ليعلمه بأنه يريد أن يرسل له بعض البيانات .**
* **ينتظر المصدر حتى يتم الرد عليه بالموافقة من الوجهة (لا يتم البدء في إرسال البيانات حتى يتم الرد بالموافقة ) وعند الرد بالموافقة تبدأ عملية إرسال البيانات.**
* **عند انتهاء إرسال كافة البيانات يتم إنهاء الاتصال والتخلي عن الموارد التي سبق حجزها لإتمام عملية نقل البيانات .**

## ثالثاً أنماط تصميم تطبيقات الويب :

## لدينا نمطان كما هو موضح في الجدول الأتي :

|  |  |  |
| --- | --- | --- |
|  | C/S (Client-Server Model) | B/S( Broser-Server Model) |
| الوصف | يجب نشر العملاء و الخوادم على طرفي الاتصال لإكمال اتصال البيانات . | تحتاج فقط لنشر خادم من جهة , واستخدام المتصفح الذي تم تكوينه افتراضيا على كل جهاز كمبيوتر على الطرف الآخر لإتمام نقل البيانات. |
| المميزات | اختيار بروتوكول مرن وقابل للتخزين المؤقت . | دورة التطوير قصيرة ,تكلفة منخفضة , أمان عالي , قابلية عالية للحمل . |
| العيوب | دورة التطوير طويلة , تكلفة عالية , أمان منخفض , إمكانية نقل ضعيفة . | اختيار بروتوكول محدود غير قابل للتخزين المؤقت . |

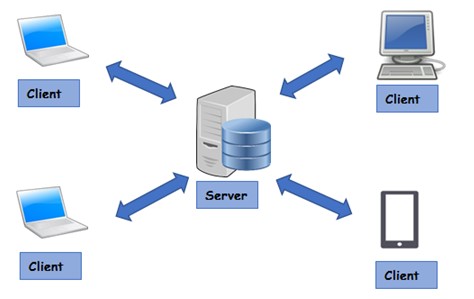
## C/S (Client-Server Model)



**الشكل (2)**

* **ما هو نموذج Client/Server ؟**

**نموذج خادم العميل هو اسم العملية المستخدمة لتوصيل المعلومات من الخادم إلى جهاز رقمي . يصف نموذج خادم العميل طريقة معينة للوصول إلى المعلومات المخزنة في الخوادم. يسمح للعديد من العملاء بفتح التطبيقات أو استرداد الملفات من خادم فردي ، مما يساعد في الحفاظ على التناسق عبر جميع الأجهزة. تُستخدم نماذج خادم العميل على نطاق واسع في جميع الصناعات التي تستخدم الخوادم لتخزين المعلومات والوصول إليها.**

****

**الشكل (3)**

**كما موضح في الشكل (3) نجد أن هناك أنواع متعددة لكل من العميل و الخادم :**

* **أنواع الخوادم :**

**خوادم الملفات**

**خوادم البريد**

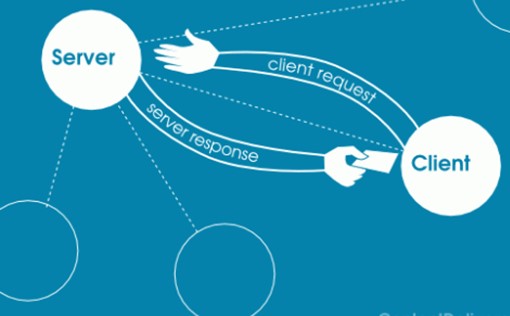
**خوادم الويب**

* **أنواع العملاء :**

**أجهزة الكمبيوتر المحمولة**

**الهواتف الذكية**

**أجهزة الكمبيوتر المكتبية**

****

**الشكل (4)**

* **كيف يعمل نموذج Client / Server ؟**

**يعمل نموذج خادم العميل كنظام تخزين للبيانات. حيث يتم تخزين الكثير من المعلومات والتطبيقات الموجودة على الجهاز أو تثبيتها فعليًا على خادم بعيد. عندما يحتاج العميل إلى الوصول إلى ملف أو تطبيق معين ، فإنه يطلب البيانات من الخادم. سيتحقق الخادم بعد ذلك من الطلب للتأكد من أنه جاء من جهاز معروف. بمجرد أن يتحقق الخادم من بيانات اعتماد العملاء ، سيقوم بتنزيل المعلومات المطلوبة على الجهاز.**

**يجوز لأجهزة العميل مشاركة الشبكة مع الخادم المضيف أو الاتصال عبر الإنترنت. نموذج خادم العميل يتبع نموذج الطلب / الاستجابة ، باستخدام نظام الرسائل لتقديم الطلبات إلى الخادم. تعمل الاتصالات بين العميل والخادم من خلال بروتوكولات محددة مثل TCP / IP.**

* **ما هي فوائد استخدام تطبيق Client / Server :**
* **زيادة أمان المعلومات الهامة حيث يتم تخزين البيانات في خادم واحد بدلاً من أجهزة متعددة ، بالتالي حماية البيانات بشكل أفضل من التهديدات الخارجية.**
* **يساعد هذا النموذج الخوادم على التعامل مع العديد من الطلبات في وقت واحد باستخدام نظام جدولة لتحديد أولويات الرسائل الواردة من العملاء.**
* **يمكن استخدام البيانات المشتركة من خلال شبكة (Client / Server )على أجهزة مختلفة لأنها ليست خاصة بالنظام الأساسي.**
* **في هذا النموذج ، يمكن استرداد البيانات بسهولة أكبر من نماذج الخادم اللامركزية.**
* **قد تساعد هذه النماذج المؤسسة في توفير المال لأنها تتطلب عادةً صيانة أقل.**
* **رابعاً TCP( Client / Server ) ؟**
* **فكرة :**

**ال server لديه اتصال مع عدة client , يميز بين client عن طريق تابع accept , عندما يحدث connect فإن التابع accept يعيد قيمتين :**

* **الأولى من النوع socket وهي socket جزئية ليميز client عن أخر .**
* **الثانية عنوان ال client .**

**• خطوات إنشاء TCP – SERVER # :**

**1- إنشاء ال SOCKET (SS)**

**SS=SOCKET.SOCKET(SOCKET, AF-INET , SOCKET-STREAM)**

**2- bind : ربط ال ip , port number**

**3- تابع listen: كم مستخدم يمكن تخديمه في نفس الوقت , العدد الأعظمي ل client ( أي كم شخص يمكنه الاتصال بهذا ال server ) .**

**4- استقبال الاتصال من خلال التابع () Accept ويدخل ال server في وضع الانتظار ( انتظار اتصال من client معين ) وتنفيذه يعيد قيمتين هما :**

* **Socket الاتصال**
* **عنوان client الذي اتصل مع ال server.**

**5- تابع send من socket الاتصال .**

**6- close .**

* **خطوات إنشاء TCP- CLIENT# :**

**1- إنشاء ال SOCKET (SS)**

**2- bind لا نقوم بها لأنه سوف يأخذ defult ip وهو ( loop block )**

**3- connect وهو تابع يأخذ بارمترين**

**Connect (ip , port num)**

**- عندما يأتي اتصال ل server يقوم بعمل socket افتراضية لكل client**

**أي عندما يأتي اتصال من client إلى server سوف يقوم بعمل socket جزئية يستخدمها للاتصال مع هذا ال client .**

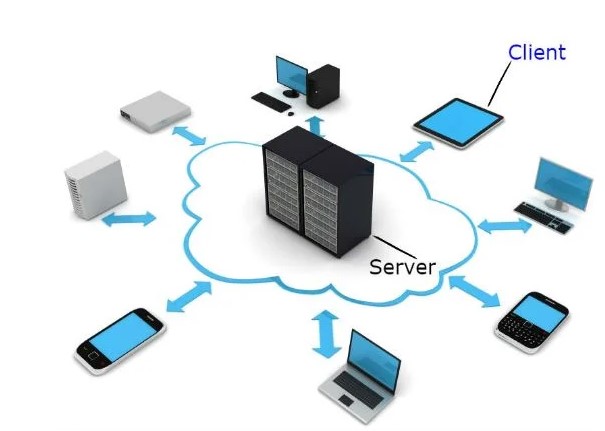
* **خامساً القسم العملي الخاص بحلقتنا :**
* **مقدمة :**

**Python هي لغة برمجة رائعة لشبكات الكمبيوتر . تسمح لنا بإنشاء تطبيقات صلبة بسرعة وسهولة . في هذا البرنامج سنقوم بتنفيذ دردشة TCP تعمل بكامل طاقتها . سيكون لدينا خادم واحد يستضيف الدردشة والعديد من العملاء الذين يتصلون به ويتواصلون مع بعضهم البعض في النهاية .**

**يمكننا أيضًا إضافة ميزات مخصصة مثل غرف الدردشة والأوامر والأدوار وما إلى ذلك .**

* **هيكل خادم العميل :**

**بالنسبة لتطبيقنا ، سوف نستخدم بنية خادم العميل . هذا يعني أنه سيكون لدينا عدة عملاء (المستخدمين) وخادم مركزي واحد يستضيف كل شيء ويوفر البيانات لهؤلاء العملاء**

****

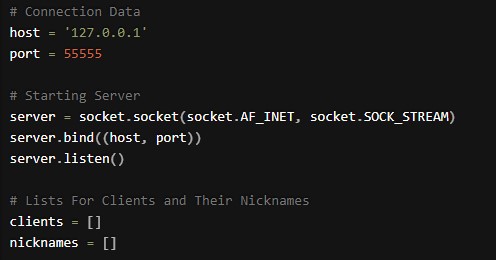
**سنحتاج إلى كتابة نصي Python . سيكون أحدهما لبدء تشغيل الخادم والآخر سيكون للعميل . سيتعين علينا تشغيل الخادم أولاً ، حتى تكون هناك محادثة يمكن للعملاء الاتصال بها , حيث أن العملاء أنفسهم لن يتواصلوا مباشرة مع بعضهم البعض ولكن عبر الخادم المركزي فقط .**

**سنبدأ بتنفيذ الخادم أولاً :**

**لهذا سنحتاج إلى استيراد مكتبتين ، وهما socket و threading . سيتم استخدام الأول للاتصال بالشبكة**

**1.jpg**

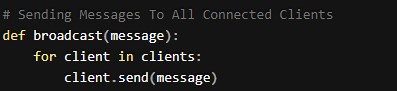
**مهمتنا التالية هي تحديد بيانات الاتصال الخاصة بنا وتهيئة المقبس الخاص بنا . سنحتاج إلى عنوان IP للمضيف ورقم منفذ مجاني لخادمنا . في مثالنا سنستخدم عنوان المضيف المحلي (127.0.0.1) والمنفذ 55555 هذا المنفذ في الواقع غير ذي صلة ولكن من الهم التأكد من أن المنفذ الذي نستخدمه مجاني وغير محجوز .**

****

**عندما نحدد المقبس الخاص بنا ، نحتاج إلى تمرير معاملين . الأول يحدد نوع المقبس الذي نريد استخدامه حيث يشير ( AF\_INET ) إلى أننا نستخدم مقبس إنترنت بدلاً من مقبس يونكس . بينما المعامل الثاني يرمز إلى البروتوكول الذي نريد استخدامه يشير SOCK\_STREAM ) ) إلى أننا نستخدم TCP وليس UDP .**

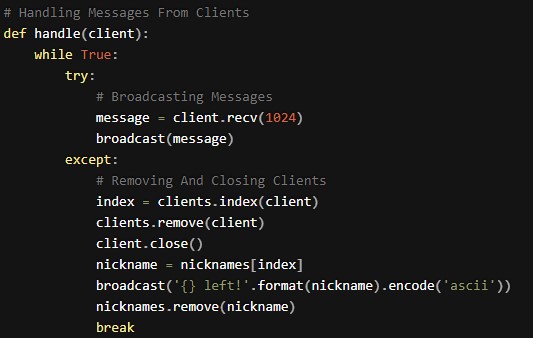
**بعد تحديد المقبس ، نربطه بمضيفنا والمنفذ المحدد عن طريق تمرير مجموعة تحتوي على كلتا القيمتين . ثم نضع خادمنا في وضع الاستماع ، بحيث ينتظر اتصال العملاء .**

**في النهاية ، سنقوم بإنشاء قائمتين فارغتين ، لنستخدمهما لتخزين العملاء المتصلين وألقابهم لاحقاً .**

****

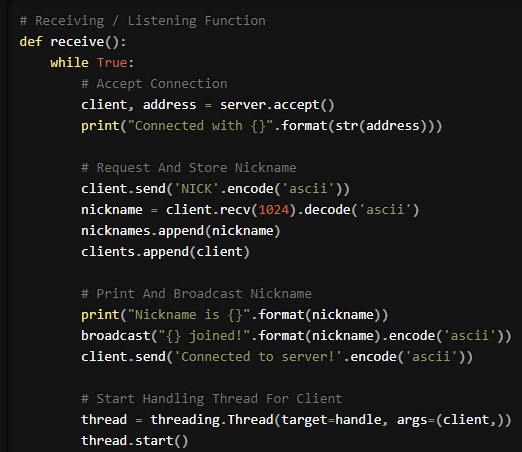
**حددنا هنا وظيفة صغيرة ستساعدنا في بث الرسائل وتجعل الكود أكثر قابلية للقراءة . ما يفعله الخادم هو مجرد إرسال رسالة إلى كل عميل متصل في قائمة العملاء .**

**سنبدأ الآن بتنفيذ الوظيفة الرئيسية الأولى . والتي ستكون مسؤولة عن التعامل مع الرسائل الواردة من العملاء .**

****

**تعمل هذه الوظيفة في حلقة while. لن يتوقف الأمر ما لم يكن هناك استثناء بسبب خطأ ما . الوظيفة تقبل العميل كمعامل . في كل مرة يتصل فيها العميل بخادمنا نقوم بتشغيل هذه الوظيفة له ويبدأ حلقة لا نهاية لها .**

**ما يفعله بعد ذلك هو تلقي الرسالة من العميل (إذا أرسل أيًا منها) وبثها إلى جميع العملاء المتصلين . لذلك عندما يرسل أحد العملاء رسالة ، يمكن لأي شخص آخر رؤية هذه الرسالة. الآن إذا كان هناك خطأ ما في الاتصال بهذا العميل لسبب ما ، نقوم بإزالته وإزالة اللقب الخاص به ، ونغلق الاتصال ونبث أن هذا العميل قد ترك الدردشة . بعد ذلك نكسر الحلقة وينتهي هذا الخيط .**

****

**عندما نكون مستعدين لتشغيل خادمنا ، سنقوم بتنفيذ وظيفة الاستلام هذه . كما أنها تبدأ حلقة لا نهاية والتي تقبل باستمرار اتصالات جديدة من العملاء . بمجرد اتصال العميل ، يرسل السلسلة "NICK" إليه ، والتي ستخبر العميل أنه تم طلب لقبه . بعد ذلك ينتظر الرد (الذي نأمل أن يحتوي على اللقب) ويلحق العميل باللقب المعني بالقوائم . بعد ذلك نقوم بطباعة هذه المعلومات وبثها .**

**أخيرًا ، سنبدأ خيطًا جديدًا يقوم بتشغيل وظيفة المعالجة التي تم تنفيذها مسبقًا لهذا العميل المعين . و الآن يمكننا تشغيل هذه الوظيفة والانتهاء من خادمنا .**

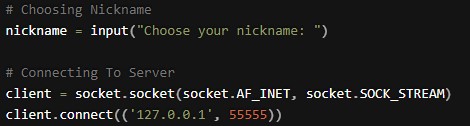
**كما نلاحظ فإننا نقوم دائمًا بترميز وفك تشفير الرسائل في كودنا هذا , والسبب في ذلك هو أنه يمكننا فقط إرسال البايت وليس السلاسل . لذلك نحتاج دائمًا إلى تشفير الرسائل (على سبيل المثال باستخدام ASCII) عندما نرسلها ونفك تشفيرها وعندما نستقبلها أيضاً .**

**6.jpg**

**الخادم عديم الفائدة إلى حد كبير بدون عملاء يتصلون به. لذلك نحن الآن بصدد تنفيذ عميلنا. لهذا سنحتاج مرة أخرى إلى استيراد نفس المكتبات .**

**7.jpg**

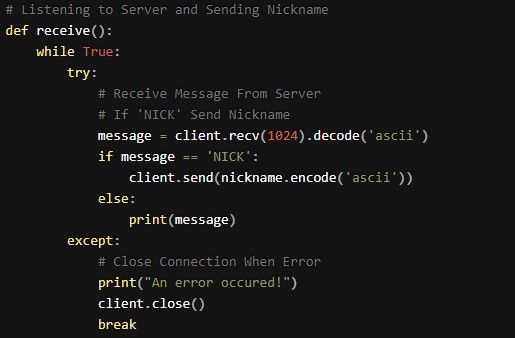
**أولى خطوات العميل هي اختيار اسم مستعار والاتصال بخادمنا . سنحتاج إلى معرفة العنوان الدقيق والمنفذ الذي يعمل فيه خادمنا .**

****

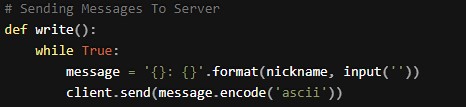
**نحن نستخدم وظيفة مختلفة هنا . بدلاً من ربط البيانات والاستماع ، نقوم بالاتصال بخادم موجود .**

**الآن ، يحتاج العميل إلى خيطين يتم تشغيلهما في نفس الوقت . الأول سيتلقى البيانات باستمرار من الخادم والثاني سيرسل رسائلنا الخاصة إلى الخادم .**

**إذن ، سنحتاج إلى وظيفتين هنا كالآتي :**

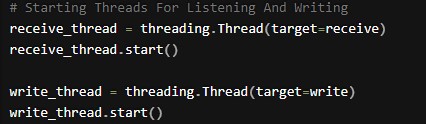
****

**مرة أخرى لدينا حلقة لا نهاية هنا. تحاول باستمرار تلقي الرسائل وطباعتها على الشاشة. ومع ذلك ، إذا كانت الرسالة "NICK" ، فإنها لا تطبعها ولكنها ترسل لقبها إلى الخادم . في حالة وجود خطأ ما ، نغلق الاتصال ونقطع الحلقة . الآن نحن فقط بحاجة إلى وظيفة لإرسال الرسائل .**



**وظيفة الكتابة قصيرة جدًا. يتم تشغيلها أيضًا في حلقة لا نهاية والتي تنتظر دائمًا إدخالًا من المستخدم . حيث بمجرد حصوله عليه ، فإنه يجمعه مع اللقب ويرسله إلى الخادم .**

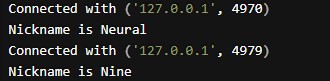
**آخر شيء يتعين علينا القيام به هو بدء سلسلتين تقومان بتشغيل هاتين الوظيفتين .**

****

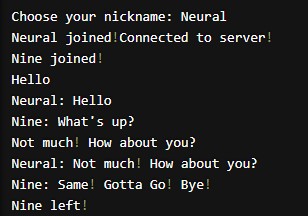
**والآن لدينا خادم يعمل بكامل طاقته , وعملاء عاملين يمكنهم الاتصال به والتواصل مع بعضهم البعض**

**سنذهب الآن لإجراء اختبار . سنحتاج دائمًا إلى بدء تشغيل الخادم أولاً لأنه بخلاف ذلك لا يمكن للعملاء الاتصال بمضيف غير موجود .**

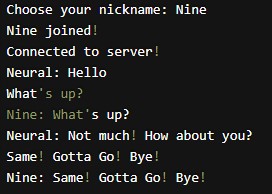
**سجل الخادم**

****

**سجل العميل**

****

**سجل بين العملاء**

****

* [**https://www.neuralnine.com/tcp-chat-in-python/**](https://www.neuralnine.com/tcp-chat-in-python/)
* [**https://ar.wikipedia.org/wiki/%D9%86%D9%85%D9%88%D8%B0%D8%AC\_%D8%A7%D9%84%D8%AE%D8%A7%D8%AF%D9%85\_%D9%88%D8%A7%D9%84%D8%B9%D9%85%D9%8A%D9%84**](https://ar.wikipedia.org/wiki/%D9%86%D9%85%D9%88%D8%B0%D8%AC_%D8%A7%D9%84%D8%AE%D8%A7%D8%AF%D9%85_%D9%88%D8%A7%D9%84%D8%B9%D9%85%D9%8A%D9%84)
* [**https://arabicprogrammer.com/article/41091436438/**](https://arabicprogrammer.com/article/41091436438/)