TUGAS INDIVIDU BAHASA ASSEMBLY

Disusun untuk memenuhi salah satu tugas mata kuliah

Teori Bahasa Formal & Auto Mata

Dosen Pengampu : Muh. Hajar Akbar, ST., M.Kom

Disusun Oleh :

Nama : Miftahul Jannah

Nim : A120043

Prodi : Teknik Informatika

UNIVERSITAS NAHDLATUL ULAMA

SULAWESI TENGGARA

Bahasa pemprograman ASSEMBLY

Bahasa assembly pertama dikembangkan pada tahun 1940-an. Meskipun pemrogram modern menghabiskan sedikit waktu untuk berurusan dengan bahasa rakitan, bahasa tersebut tetap penting untuk keseluruhan fungsi komputer.

Bahasa pemrograman tingkat rendah seperti bahasa assembly adalah jembatan yang diperlukan antara perangkat keras yang mendasari komputer dan bahasa pemrograman tingkat tinggi, seperti Python atau JavaScript. Di mana program perangkat lunak modern ditulis.

Bahasa assembly atau biasa disebut bahasa rakitan atau bahasa mesin. Alasannya mungkin karena bahasa assembly memang sangat dekat dengan prosesor dibandingkan bahasa tingkat tinggi seperti C dan lain-lain. Perlu diketahui juga bahwa bahasa assembly ini ada bermacam-macam tergantung jenis prosesor yang menjadi targetnya. Jadi bahasa assembly untuk PC yang menggunakan arsitektur prosesor x86 atau x64 akan berbeda dengan bahasa assembly untuk arsitektur ARM, ARM64, MIPS atau yang lainnya. Selain itu, pada setiap assembler yaitu aplikasi untuk mengubah *source code* menjadi file biner memiliki sintaks tersendiri. Misalnya assembler GNU atau biasa disebut gas dapat menggunakan sintaks AT&T dan Intel tergantung direktif yang diberikan

Biar tidak terlalu panjang lebar, *download* salah satu assembler itu untuk dijadikan bahan eksperimen yaitu Macro Assembler (MASM).

Perlu diingat bahwa masm32 ini dirancang untuk menghasilkan *executable* 32-bit. Pilih salah satu *mirror* terdekat untuk *download* file zipnya. Untuk Indonesia, *mirror* terdekat adalah Australia. Setelah *download* selesai, lanjutkan dengan mengekstrak dan menginstall masm32. Secara default proses instalasi akan menyimpan file masm32 ke folder c:\masm32\. Lalu jangan lupa masukkan folder instalasi masm32 ke dalam *PATH Environment Variables*.

Untuk contoh kali ini akan digunakan dari tutorial LSASS Injection. Pada artikel tersebut terdapat *source code* menggunakan bahasa C. *Source code* tersebut akan saya tulis ulang menggunakan bahasa assembly dengan sintaks masm32. Berikut ini adalah *source code* agent yang dikonversi dari bahasa C++ menjadi bahasa assembly dengan sintaks masm32:

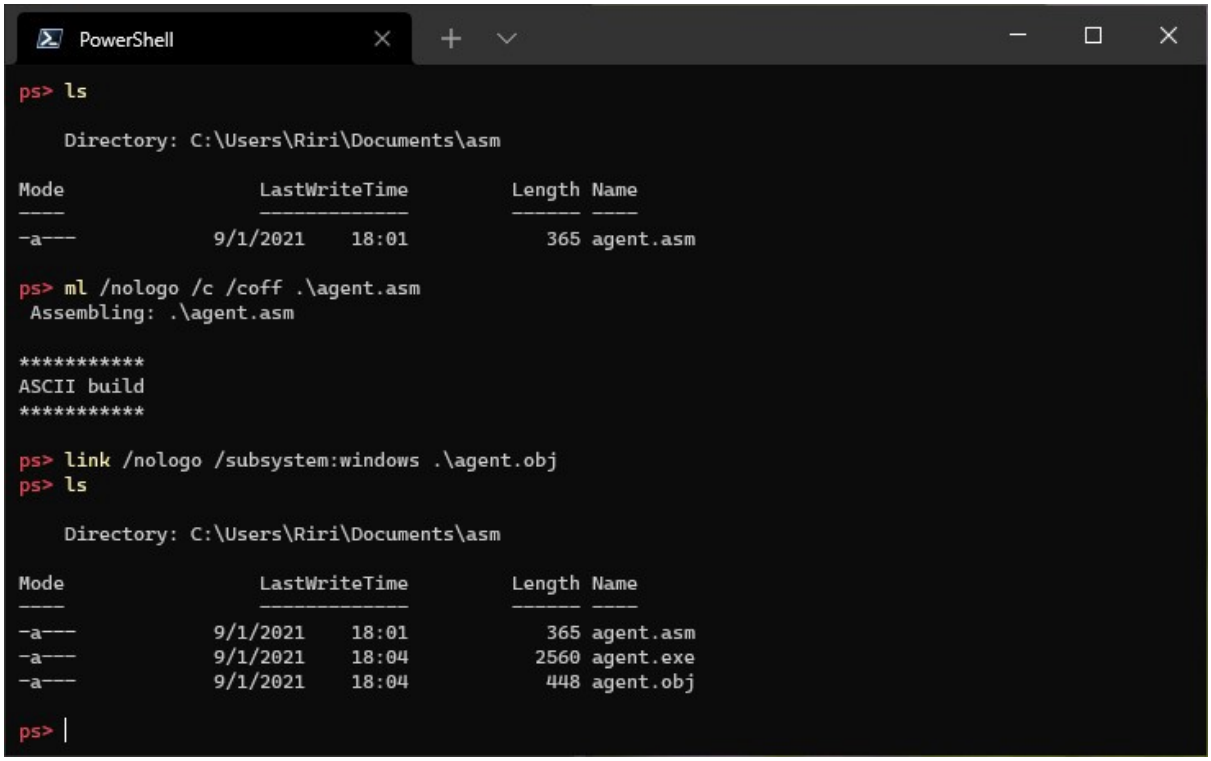
```
include \masm32\include\masm32rt.inc

.code
main:
    invoke  Sleep, 3600 ; istirahat selama 3600 detik (1 jam)
    jmp     main        ; infinite loop
end main
```

Sedangkan *source code* untuk honeycred yang sudah dikonversi ke assembly adalah sebagai berikut:

```
include \masm32\include\masm32rt.inc
includelib \masm32\lib\advapi32.lib

CreateProcessWithLogonW proto :ptr, :ptr, :ptr, :dword, :ptr, :ptr, :dword, :ptr, :ptr, :ptr, :pt

.data
 str_usr    db "felix",0
 str_dom    db "contoso.com",0
 str_pwd    db "xQc2021!!",0
 str_dir    db "C:\",0
 str_app    db "C:\Users\administrator\Desktop\agent.exe",0
 s_sinfo    STARTUPINFO <>
 s_pinfo    PROCESS_INFORMATION <>

.code
main:
   mov      s_sinfo.dwFlags, 00000001h
   mov      s_sinfo.wShowWindow, 0
   invoke   CreateProcessWithLogonW, addr str_usr, addr str_dom, addr str_pwd, 00000002h, addr str
end main
```

Untuk melakukan *assemble* dan *link* menggunakan masm32, gunakan perintah seperti ini:

```
ml /nologo /c /coff agent.asm
link /nologo /subsystem:windows agent.obj
```



Jadi proses untuk mengubah dari *source code* menjadi *executable* terdiri dari dua proses. Proses pertama mengubah dulu *source code* assembly menjadi file objek, baru kemudian diubah menjadi *executable* menggunakan proses *linking*. Tapi ada juga *assembler* yang bisa langsung mengubah *source code* assembly menjadi *executable* yaitu Flatassembler (FASM). Jika menggunakan bahasa assembly, ukuran *executable* yang dihasilkan cukup kecil. Misalnya *executable* agent.exe di atas itu ukurannya hanya 2560 byte atau 2.5 KB. Jika dijalankan, maka agent.exe akan berjalan secara terus-menerus dan cara menghentikannya adalah dengan menggunakan *Task Manager*. Proses *assemble* & *link* dapat dibuat menjadi mudah dengan menggunakan *script PowerShell* seperti ini:

```
1   (ls *.asm) | % {ml /nologo /c /coff $_}
2   (ls *.obj) | % {link /nologo /subsystem:windows $_}
3   (ls *.obj) | % {rm $_}
```