MASM32 in Ubuntu

Using MASM32 on Ubuntu with Wine

MASM32 is a compiler and assembler for the Windows operating system. It cannot be directly used on non-Windows platforms. Although alternatives such as **NASM** and **TASM** exist, their code structure and functionality differ significantly from MASM. To use MASM on Ubuntu, **Wine** must be used as an intermediary. Wine creates a virtual Windows environment that allows MASM executables to run on Linux.

- ## Direct Installation

Script to directly install wine, masm32 and shell to execute an assembly code

Copy the script and paste it on the Ubuntu terminal. A total of 1.9 GB files will be downloaded. Provide permission and/or password whenever asked. MASM will be installed on `~/.wine` folder. If the folder is not visible from file manager, press `CTRL+H`.

```
#wine64 installation
  sudo apt install wine64 #wine32 if 32bit cpu
  wineboot
  wine --version
  sudo apt upgrade

#masm32 installation
  curl -o masm32v11r.zip https://masm32.com/download/masm32v11r.zip
  unzip masm32v11r.zip
  cd ~/Downloads/masm32v11r
  wine start install.exe

#creation of script to run .asm file
  mkdir ~/assembly/
  cd ~/assembly/
  touch ./build.sh
  chmod +x build.sh
  cat << 'EOF' > build.sh
#!/bin/bash
FILENAME=$1
WIN_FILENAME=$(winepath -w "$FILENAME.asm")
if ! wine "C:\\masm32\\bin\\ml.exe" /c /coff /Cp "$WIN_FILENAME"; then
    echo "Error: Assembly failed!"
    exit 1
fi
WIN_OBJ_FILE=$(winepath -w "$FILENAME.obj")
if ! wine "C:\\masm32\\bin\\link.exe" -entry:main /subsystem:console "$WIN_OBJ_FILE"; then
    echo "Error: Linking failed!"
    exit 1
fi
WIN_EXE_FILE=$(winepath -w "$FILENAME.exe")
if ! wine "$WIN_EXE_FILE"; then
    echo "Error: Executable failed to run!"
    exit 1
fi
EOF
```

Write the assembly programs in `~/assembly` directory of your computer. To assemble, compile and execute an assembly code named `program.asm`, go to terminal and write

```
./build.sh program      #not program.asm
```

The file should run now.

- # Step by step guide

Steps to Set Up and Run MASM32 on Ubuntu

1. Check CPU Architecture

The CPU architecture can be checked by typing the following in the terminal:

```
lscpu
```

Check the value listed under `CPU op-mode(s)`. This indicates the instruction set and application architecture (e.g., 32-bit, 64-bit) that the CPU can support.

2. Install Wine

Installation of Wine requires 1.8 GB of disk space. Make sure that enough free storage is available before proceeding to installation.

- If the CPU architecture is **32-bit**, the following command should be used to install 32-bit Wine:

  ```
  sudo apt install wine32
  ```

- If the CPU architecture is **64-bit**, 64-bit Wine should be installed using this command:

  ```
  sudo apt install wine64
  ```

- Some CPUs support both 32-bit and 64-bit modes, so either version can be installed.

Once installed, Wine must be configured with:

```
wineboot
wine --version
```

This ensures that Wine is installed correctly. To update all packages, the following command should be run:

```
sudo apt upgrade
```

3. Download and Install MASM32

1. MASM32 can be downloaded from the official website or via this direct link.
2. Once downloaded, the zip file should be extracted.
3. The unzipped folder will contain an `install.exe` file. To navigate to this folder, the following terminal command can be used:

   ```
   cd ~/Downloads/masm32v11r
   #cd /path/to/unzipped/folder
   ```

4. The MASM32 installer should be run using Wine:

```
wine start install.exe
```

   The installation prompts should be followed, and all terms agreed to. After successful installation, MASM32 will be stored in the `~/.wine` directory.

For more information or clarity, this page can be consulted.

Running Assembly Code with MASM32

1. A script file named `build.sh` (or any other preferred name) should be created with the following content:

```
#!/bin/bash

FILENAME=$1

# Convert the filename to a Windows-compatible path
WIN_FILENAME=$(winepath -w "$FILENAME.asm")

# Assemble the ASM file to produce the object file
if ! wine "C:\\masm32\\bin\\ml.exe" /c /coff /Cp "$WIN_FILENAME"; then
    echo "Error: Assembly failed!"
    exit 1
fi

# Convert the object file to a Windows-compatible path
WIN_OBJ_FILE=$(winepath -w "$FILENAME.obj")

# Link the object file to produce the executable
if ! wine "C:\\masm32\\bin\\link.exe" -entry:main /subsystem:console "$WIN_OBJ_FILE"; then
    echo "Error: Linking failed!"
    exit 1
fi

# Convert the executable file to a Windows-compatible path
WIN_EXE_FILE=$(winepath -w "$FILENAME.exe")

# Run the executable
if ! wine "$WIN_EXE_FILE"; then
    echo "Error: Executable failed to run!"
    exit 1
fi
```

Or

```
filename=$1
wine "C:\\masm32\\bin\\ml.exe" /c /coff /Cp "$filename.asm"

wine "C:\\masm32\\bin\\link.exe" -entry:main /subsystem:console "$filename.obj"

wine "$filename.exe"
```

2. The file should be saved, closed, and made executable by running:

```
chmod +x ./build.sh
```

3. To assemble and run an `.asm` file, the following command should be used:

   ```
   ./build.sh filename
   ```

   Note: Only the filename should be entered without the extension (`prog1` instead of `prog1.asm`). The script will automatically append the `.asm` extension. It is also important that both the `build.sh` script and the `.asm` file are in the same directory, and the terminal should be pointed to that directory.

Otherwise, this command can can be directly executed to run an assembly file(for example, `output.asm`) that is located in the active directory(active path in terminal):

```
wine "C:\\masm32\\bin\\ml.exe" /c /coff /Cp output.asm

wine "C:\\masm32\\bin\\link.exe" -entry:main /subsystem:console output.obj

wine "output.exe"
```

- # Uninstalling Wine and Wine Applications

1. Remove Applications Installed via Wine

To uninstall an application installed through Wine, the following command should be executed:

```
wine uninstaller
```

A GUI will appear, allowing for the uninstallation of specific applications. If no GUI appears, the entire `~/.wine` folder can be deleted, which will remove all applications installed via Wine:

```
rm -r ~/.wine
```

The Wine environment can later be recreated by running:

```
wineboot
```

2. Uninstall Wine

Wine can be removed from Ubuntu with the following command:

```
sudo apt-get --purge remove wine wine64 wine32 libwine fonts-wine
```

If Wine is not fully uninstalled, these commands should be run to manually delete any leftover files:

```
cd $HOME
rm -r .wine
rm .config/menus/applications-merged/wine*
rm -r .local/share/applications/wine
rm .local/share/desktop-directories/wine*
rm .local/share/icons/????_*.xpm
```

To complete the removal, the following command should be run:

```
sudo apt-get remove --purge wine
```

Finally, to correct any installation errors, these commands should be executed:

```
sudo apt-get update
sudo apt-get autoclean
sudo apt-get clean
sudo apt-get autoremove
```

For further guidance, the instructions found in this Ask Ubuntu guide can be consulted.