

# AmigaOS3(m68k) için en güncel çapraz derleme ortamı kurulumu

Eğer AmigaOS için halen etkin olarak güncellenmekte olan açık kaynak programlardan (örn *YAM*, *AmiSSL*, *NList.mcc* vs.) bazılarını derleme ihtiyacı duyduysanız, bunların artık *SAS/C*, *StormC* vs. gibi *legacy* derleyiciler ve onlarla birlikte gelen güncelliğini kaybetmiş API'ler ile derlenemediğini fark etmişsinizdir. Ayrıca forumlar, web sayfaları ve hatta Aminet'te yayınlanan *m68k toolchain*'ler de bir işe yaramıyor. Şimdilik bu tip güncel programları *m68k* için derleyebileceğimiz tek ortam <https://github.com/cahirwpz/amigaos-cross-toolchain> adresinden edinebileceğiniz çapraz derleme ortamı. *cahirwpz* ortamı yalnızca kaynak kodu olarak yayınladığı için, kullanabilmek için ilk önce derlememiz gerekiyor ve işbu doküman bunu kolayca yapabilmeyi sağlayacak *terminal* yönergelerini içeriyor.

## Gereksinimler:

- Güncel bir Linux Sürümü. *Bu örnekte Ubuntu 16.04 LTS (32bit).*  
**64bitlik Linux dağıtımı kullanıyorsanız ŞİMDİ ilgili kısmı okuyunuz (Sayfa 4).**  
**(!) Bir nedenle 14.04 LTS sürümüne bağımlıysanız ŞİMDİ ilgili kısmı okuyunuz (Sayfa 3-4).**
- GNU gcc 5.x **32-bit sürümü!** or Clang
- Python 2.7.x
- libncurses-dev
- python-dev 2.7
- GNU make 4.x
- perl 5.22
- git
- GNU patch
- GNU gperf
- GNU bison

Tüm güncellemeleri yaptıysanız bu listeden sadece libncurses-dev, python-dev, git, patch, gperf ve bison'u kurmanız gerekecek. Hemen kuralım:

```
> sudo apt-get install libncurses-dev
> sudo apt-get install python-dev
> sudo apt-get install patch
> sudo apt-get install gperf
> sudo apt-get install bison
> sudo apt-get install git
```

Evet artık Ubuntu 16.04'ümüz **amigaos-cross-toolchain**'i derlemeye hazır.

İlk önce kaynak kodu sabit diskimize git ile indiriyoruz (derleme böyle yapmamızı gerektiriyor).

Bir indirme konumu belirleyelim, (ben ~/Genel/AmigaDev tercih ettim):

```
> cd ~/Genel
> mkdir AmigaDev
> cd AmigaDev
> git clone git://github.com/cahirwpz/amigaos-cross-toolchain.git
> cd amigaos-cross-toolchain
```

Evet... Sonunda derleme aşamasına geldik. Önce /opt dizinimize (dilersek geçici olarak) kullanıcı değişiklik izni verelim:

```
> sudo chmod a+w /opt
```

...ve derleyelim:

```
> ./toolchain-m68k --prefix=/opt/m68k-amigaos build
```

...sonra ekranınız bilim kurgu filmlerdeki konsollar gibi olacak... ..oldukça uzun süren bir dizi işlem... Tamamlanıncaya kadar bekliyoruz.

Bu tamamlandıktan sonra dilersek aşağıdaki eklentileri de kurabiliriz:

```
> ./toolchain-m68k --prefix=/opt/m68k-amigaos install-sdk ahi cgx mui
```

**(!) MUI geliştiricileri EK2'yi okusun!**

Son olarak yeni derlenen *gcc* ve diğer araçları yola ekliyoruz:

```
> export PATH=/opt/m68k-amigaos/bin:$PATH
```

Artık AmigaOS3 için derleme yapacağımızda **m68k-amigaos-gcc** komutunu kullanacağız. Deneyelim bakalım olmuş mu?

```
> m68k-amigaos-gcc -v
...
Gcc version 2.95.3
```

...yazarsa her şey tamam demektir.

Şimdi son olarak bu yeni yolu **.bashrc**'ye ekleyelim ki terminali her açtığımızda bu yol erişilebilir olsun:

```
> leafpad ~/.bashrc
```

Dosyanın en sonuna iniyoruz ve aşağıdaki girdiyi ekleyip, kaydediyoruz:

```
# m68k AmigaOS gcc binaries
export PATH=/opt/m68k-amigaos/bin:$PATH
```

Evet arkadaşlar. Geçmiş olsun. Çapraz derleme ortamımız artık hazır. Güle güle güzel kodlarda kullanın.

#### EK :

AmigaOs3.9 NDK başlık dosyalarından birinde iki kez tanımlanma yapıldığı için derlemelerde uyarı iletisi veren gereksiz bir makro tanımlaması var. Bunu devre dışı bırakırsak derleme iletilerimiz daha temiz olabilir:

```
> leafpad /opt/m68k-amigaos/m68k-amigaos/ndk/include/g_raphics/gfxmacros.h
```

...120'nci satıra atlayalım ve bu satırı /\* ve \*/ arasına alarak devre dışı bırakalım:

```
/* #define GetOutlinePen(rp) GetOPen(rp) */
```

...kaydedip çıkalım.

#### EK2:

**amigaos-cross-toolchain** ile gelen derleme betiği, *gcc*'nin ihtiyaç duyduğu *mui lib* dosyasını sadece *libnix* için kopyalıyor. Ancak derleyeceğimiz kaynak kodları *nostdlib* ile derliyor olabilir. Bu durumlarda:

```
ld: cannot open -lmui: No such file or directory
```

...hatası alabilirsiniz.

Aşağıdaki satırı terminalden çalıştırarak bu eksikliği giderebiliyoruz:

```
> cp /opt/m68k-amigaos/m68k-amigaos/libnix/lib/libmui.a /opt/m68k-amigaos/lib/gcc-lib/m68k-amigaos/2.95.3/
```

...benzer sorun *debug* kütüphanesi için de karşımıza çıkabilir. Öyleyse hadi onu da kopyalayalım:

```
> cp /opt/m68k-amigaos/m68k-amigaos/libnix/lib/libdebug.a /opt/m68k-amigaos/lib/gcc-lib/m68k-amigaos/2.95.3/
```

#### EK3:

Linux kurulumumuzda Amiga yazılımı derlemeyi hedeflediğimiz için Amiga dosya sıkıştırma standardı *LHA*'ya her an ihtiyaç duyabiliriz. Sistemimizde bulunsa iyi olur:

```
> sudo apt-get install jlha-utils
```

#### EK4:

Bazı kaynak kodları *build* aşamasında Amiga'nın yeni standart *catalog* derleyicisi *flexcat*'i kullanıyor olabiliyor. Bunun kurulumunu da şimdiden yapsak iyi olur:

```
> wget http://aminet.net/dev/misc/FlexCat-2.18.lha
> lha -e FlexCat-2.18.lha FlexCat/Linux-i386/flexcat
> chmod a+x FlexCat/Linux-i386/flexcat
> cp FlexCat/Linux-i386/flexcat /opt/m68k-amigaos/bin/
> rm -r FlexCat FlexCat-2.18.lha
```

Şimdi dilersek */opt* dizinimize kullanıcı için verdiğimiz değişiklik yapma iznini kaldırabiliriz:

```
> sudo chmod u-w /opt
```

## Ubuntu 14.04 LTS kullananlar için:

Linux dağıtımları birlikte geldikleri *C/C++*, *perl*, *python* vs. derleyicilerine bağımlı oldukları için bu yazılımları ancak uyumlu oldukları en yüksek sürüme kadar güncelleyebiliyorlar (örneğin *trusty* depolarında *gcc-5*'i bulamazsınız). Ancak bunlara ihtiyacımız varsa bunları yan (*alternative*) kurulumlar olarak kurup ihtiyacımız olduğunda kullanmamız mümkün. Aşağıdaki *terminal* yönergeleri bunu sağlamanıza yardımcı olacaktır.

### - İlk önce GCC sürümümüzü 5.x'e getiriyoruz :

```
> sudo add-apt-repository ppa:ubuntu-toolchain-r/test
> sudo apt-get update
> sudo apt-get install gcc-5 g++-5
> sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-5 1 --slave /usr/bin/g++ g++ /usr/bin/g++-5
```

### - Şimdi make sürümümüzü yükseltelim :

```
> mkdir ~/Genel/src ; cd ~/Genel/src
> wget ftp://ftp.gnu.org/gnu/make/make4.2.tar.gz
> tar -xzf make4.2.tar.gz ; cd make4.2
> sudo make install
```

...tamamlandığında:

```
> make -v
GNU make 4.2
...
```

...yazarsa yükseltme tamam demektir. Dilersek kaynak kodu silebiliriz.

```
> cd ~
> rm -rf ~/Genel/src
```

### - Şimdi git'i kuralım:

```
>sudo apt-get install git
```

### - Şimdi PerlBrew kuruyoruz:

```
>sudo apt-get install perlbrew
```

*perlbrew* için kök dizini belirliyoruz (ben **/opt/perlbrew** tercih ettim)

*Tabi önce bu konunun kullanıcı değişiklik yapma iznine sahip olması gerekiyor:*

```
> sudo chmod a+w /opt
```

...

```
> export PERLBREW_ROOT=/opt/perlbrew
> source /opt/perlbrew/etc/bashrc
> leafpad ~/.bashrc
```

...aşağıdakileri dosyanın en altına ekleyip, kaydediyoruz:

```
# PerlBrew Environment Variables
export PERLBREW_ROOT=/opt/perlbrew
source /opt/perlbrew/etc/bashrc
```

...şimdi perlbrew'i ilklendirmemiz gerekiyor :

```
> perlbrew init
```

- **Şimdi ihtiyacımız olan perl sürümünü kuruyoruz:**

```
> perlbrew install perl-5.22.2
```

Yeni bir terminal penceresi açıyor ve şunu yazıyoruz:

```
> tail -f /opt/perlbrew/build.perl-5.22.2.log
```

Eğer make aşamasında:

```
error: invalid preprocessing directive # HAS_SCM_RIGHTS
```

...hatası verirse:

```
> leafpad /opt/perlbrew/build/perl-5.22.2/config.h
```

**Ara -> "HAS\_SCM" yapalım ve bir sonrakini bulalım (CTRL+G)**

...bakın burada #'den sonra **define** anahtar sözcüğü eksik. Eksiği tamamlayalım ve kaydedip çıkalım.

...şimdi tekrar derlemeyi deneyelim:

```
> cd /opt/perlbrew/build/perl-5.22.2/  
> make install
```

Derleme tamamlandı:

```
> perlbrew switch perl-5.22.2
```

...kurulum başarılı olmuş mu sınavalım:

```
> perl -v  
...  
perl 5, version 22 subversion 2  
...
```

...yazarsa işlem tamam demektir.

- **Şimdi Python sürümümüzü kontrol edelim. (Ubuntu 14.04 2.7 ile geliyor ki bu bizim için yeterli)**

```
> python --version  
Python 2.7.6
```

---

*Şimdi amigaos-cross-toolchain kurulumuna kaldığınız yerden devam edebilirsiniz.*

---

## **64bitlik Linux dağıtımı kullananlar:**

Kurulumu başlamadan önce aşağıdaki komutları çalıştırsanız 32bitlik programları sistemlerine çalıştırabilirler.

```
> sudo apt-get install gcc-multilib  
> sudo dpkg --add-architecture i386  
> sudo apt-get update  
> sudo apt-get dist-upgrade
```