

# DERLEYİCİLER VE ÇALIŞTIRMA ORTAMI

Genel amaçlı bilgisayarların çok kullanılmasıyla birlikte metin editörleri de (Windows Not Defteri, Mac TextEdit, Notepad++, Epsilon, EMACS, nano, vim veya vi) işletim sistemlerinin bir parçası olmuştur. Metin editörlerinde yazılan programlar ise **derleyiciler** (**compiler**) tarafından makine koduna çevrilerek **icra edilebilir** (**executable**) dosya olarak kaydedilebilmektedir.

C dilinde programlama öğrenmeye başlamak için ilk adım, programı yazabileceğiniz bir metin editörü kullanmak ve yazdığınız programa ilişkin kaynak kodları c uzantısı ile dosya olarak saklamaktır. İkinci adım ise **işletim sisteminizde** (**operating system**) çalışabilen bir icra edilebilir dosya oluşturan bir derleyici kurmaktır. Yani bilgisayarınızda iki yazılım aracına ihtiyacınız vardır; birincisi C derleyicisi, ikincisi ise basit bir betin düzenleyicisidir.

Derleyici için girdi, kaynak kodu dosyasıdır. Kaynak dosyasında yazılan kaynak kodu, içerisinde C **talimatları** (**statement**) olan, insan tarafından okunabilen metin dosyasıdır. Kaynak kodda verilen talimatlara göre işlemcinizin programı çalıştırabilmesi için, bunun makine diline **derlenmesi** (**compile**) gerekir. **Derleme**, bir başka programlama dilinden **sembolik isim** (**mnemonic**) ya da **makine koduna** (**instruction code**) dönüştürme işlemidir. Bunu yapan programlara **derleyici** (**compiler**) adı verilir.

## C derleyicileri

Günümüzde birçok C derleyicisi mevcuttur. En çok kullanılanları aşağıda verilmiştir;

- GNU Derleyici Koleksiyonu (GCC) – GCC, popüler bir açık kaynaklı C derleyicisidir<sup>9</sup>. Windows, macOS ve Linux dahil olmak üzere çok çeşitli platformlarda kullanılabilir. GCC, çok çeşitli özellikleri ve çeşitli C standartlarına desteğiyle bilinir.
- Clang – Clang, LLVM projesinin bir parçası olan açık kaynaklı bir C derleyicisidir<sup>10</sup>. Windows, macOS ve Linux dahil olmak üzere çok çeşitli platformlarda kullanılabilir. Clang, hızı ve optimizasyon yetenekleriyle bilinir.
- Microsoft Visual C – Microsoft Visual C, Microsoft tarafından geliştirilen tescilli bir C derleyicisidir<sup>11</sup>. Yalnızca Windows için kullanılabilir. Visual C, Microsoft Visual Studio geliştirme ortamıyla entegrasyonuyla bilinir.

Bilgisayarınızın 64 bitlik Windows olduğu varsayılarak GCC derleyicisinin MINGW sürümü ilgili web sayfasından indirilerek kurulur<sup>12</sup>. Kurulumu tamamlandıktan sonra PATH değişkenine GCC derleyicisinin klasörü eklenir. Böylece kurulum tamamlanmış olur.

Komut satırı (**DOS Prompt**) açıldığında **gcc -v** komutu yazılarak derleyicinin düzgün kurulup kurulmadığı kontrol edilebilir.

```
C:\Users\ILHANOZKAN>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=D:/msys64/ucrt64/bin/./lib/gcc/x86_64-w64-mingw32/13.2.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ./gcc-13.2.0/configure --prefix=/ucrt64 --with-local-prefix=/ucrt64/local --
build=x86_64-w64-mingw32 --host=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --with-native-
system-header-dir=/ucrt64/include --libexecdir=/ucrt64/lib --enable-bootstrap --enable-
checking=release --with-arch=nocona --with-tune=generic --enable-
languages=c,lto,c++,fortran,ada,objc,obj-c++,jit --enable-shared --enable-static --enable-
libatomic --enable-threads=posix --enable-graphite --enable-fully-dynamic-string --enable-
libstdcxx-filesystem-ts --enable-libstdcxx-time --disable-libstdcxx-pch --enable-lto --enable-
libgomp --disable-libssp --disable-multilib --disable-rpath --disable-win32-registry --
disable-nls --disable-werror --disable-symvers --with-libiconv --with-system-zlib --with-
```

<sup>9</sup> <https://gcc.gnu.org/>

<sup>10</sup> <https://clang.llvm.org/>

<sup>11</sup> <https://visualstudio.microsoft.com/tr/vs/features/cplusplus/>

<sup>12</sup> <https://www.mingw-w64.org/downloads/>

```
gmp=/ucrt64 --with-mpfr=/ucrt64 --with-mpc=/ucrt64 --with-isl=/ucrt64 --with-pkgversion='Rev3,
Built by MSYS2 project' --with-bugurl=https://github.com/msys2/MINGW-packages/issues --with-
gnu-as --with-gnu-ld --disable-libstdc++-debug --with-boot-ldflags=-static-libstdc++ --with-
stage1-ldflags=-static-libstdc++
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 13.2.0 (Rev3, Built by MSYS2 project)
```

```
C:\Users\ILHANOZKAN>
```

Daha sonra metin editörü ile ilk kaynak kodumuzu oluşturmak için komut satırında **notepad hello.c** komutu yazılır.

```
C:\Users\ILHANOZKAN>notepad hello.c
```

Eğer böyle bir dosya yoksa editör bu dosyayı oluşturmak için onay ister. Açılan metin editörü penceresine ilk C programı yazılır ve **hello.c** olarak dosyaya kaydedilir. Eğer böyle bir dosya yoksa editör bu dosyayı oluşturmak için onay ister. Açılan metin editörü penceresine ilk C programı yazılır ve **hello.c** olarak dosyaya kaydedilir.

```
#include <stdio.h>

int main() {
    /* İlk C Programı */
    printf("Hello, World! \n");
    return 0;
}
```

Aşağıdaki komut ile **hello.c** olarak hazırlanan kaynak kodumuz derleyici tarafından icra edilebilir **hello.exe** dosyası olarak derlenir.

```
C:\Users\ILHANOZKAN>gcc hello.c -o hello.exe
```

Derlenen programı çalıştırmak için yine aynı komut satırında **hello.exe** yazarak icra edilebilir programımız işletim sistemi tarafında işlemciye hedef gösterilerek çalıştırılır.

```
C:\Users\ILHANOZKAN>hello.exe
Hello, World!
```

```
C:\Users\ILHANOZKAN>
```

Burada programın yazılması **notepad** programında, derlenmesi **gcc** programı ile ve derlenen **hello.exe** programının çalışması tarafımızdan yapılmıştır.

Mac OSX ve Linux işletim sistemlerinde de derleyici kurulumu tamamlandıktan sonra terminal yani konsol uygulaması açılarak aşağıdakine benzer şekilde derlemeler yapılabilir.

## Entegre Geliştirme Ortamı

Şimdiye kadar anlatılan **kod yazma** (implementation), **derleme** (compile), **icra** (execute) ya da **koşma** (run) ve **izleme** (trace) gibi süreçlerin tamamını tek bir çatı altında yürütmemizi sağlayan programlar, **entegre geliştirme ortamı** (Integrated Development Environment-IDE) olarak adlandırılır.

C programlama dili için en çok kullanılan entegre geliştirme ortamları aşağıda verilmiştir;

- Code::Blocks – Açık kaynaklı bir C/C++ geliştirme ortamı olup bir **grafik kullanıcı ara yüzüne** (Graphic User Interface-GUI) sahiptir. Windows, macOS ve Linux işletim sistemleri üzerine kurulabilmektedir.
- Visual Studio – Microsoft tarafından geliştirilen, C dilinde yazılmış, güçlü, yüksek performanslı uygulamalar oluşturmak için kullanılabilen bir geliştirme ortamıdır. Yalnızca Windows'ta çalışır. Visual Studio, **kod tamamlama** (intellisense), **kullanıcı ara yüzü** (user interface-UI) hazırlama

desteği ve **hata ayıklama** (**debugger**) ve birçok **eklentisi** (**plug-in**) olan muazzam özelliklere sahiptir. Bu geliştirme ortamının kurulumu dipnotta verilmiştir<sup>13</sup>.

- Visual Studio Code – Microsoft tarafından geliştirilen açık kaynaklı bir geliştirme ortamıdır. Windows, macOS ve Linux gibi işletim sistemlerinde çalışır. Git **sürüm kontrol** (**version control**) sistemleriyle çok iyi çalışır. Ayrıca, akıllı kod tamamlamanın dikkate değer özellikleriyle birlikte gelir.
- CLion – JetBrains tarafından geliştirilmiştir ve C++ programcıları için en çok önerilen çapraz platformlu (CMake derleme sistemiyle entegre macOS, Linux ve Windows'u destekler) geliştirme ortamıdır. Ayrıca, **yerel** (**local**) ve **uzaktan** (**remote**) desteğe sahip birkaç IDE'den biri olarak kabul edilir. Bu da yerel bir makinede kod yazmanıza ancak uzak sunucularda derlemenize olanak tanır. Kaynak kodlarımız yönetmemizi sağlayan **Concurrent Versions System-CVS** ve **Team Foundation Server-TFS** ile entegre edilebilir.
- Eclipse – Java'da yazılmış ve IBM tarafından geliştirilmiş ücretsiz ve açık kaynaklı bir geliştirme ortamıdır. Yaklaşık otuz programlama dilini desteklediği için geniş topluluk desteğiyle iyi bilinir. C++ için Eclipse IDE, kod tamamlama, otomatik kaydetme, derleme ve hata ayıklama desteği, uzak sistem gezgini, statik kod analizi, **profilleme** (**profiling**) ve **yeniden düzenleme** (**refactoring**) gibi beklenebilecek tüm özelliklere sahiptir. Ayrıca çeşitli harici eklentileri entegre ederek işlevselliğini genişletebilirsiniz. Çok platformludur ve Windows, Linux ve macOS'ta çalışabilir.
- NetBeans – Apache Yazılım Vakfı – Oracle Corporation tarafından geliştirilen ücretsiz ve açık kaynaklı bir geliştirme ortamıdır. Öğrenciler veya başlangıç seviyesindeki C/C++ geliştiricileri için şiddetle tavsiye edilmesinin sebebi, Eclipse'e benzer şekilde daha iyi sürükle ve bırak işlevlerine sahip olmasıdır. Windows, Linux, MacOSX ve Solaris gibi birden fazla platformda çalışır.
- Xcode – MacOSX işletim sisteminde yazılım geliştirmek için kullanılan bir entegre geliştirme ortamıdır.

Entegre geliştirme ortamları metin dosyası olarak tutulan kaynak kodları programcıya renklendirerek gösterir. Bu da programcının kodu anlamasını daha da kolaylaştırır. Ancak kodu dosyaya yine sade metin olarak kaydeder.

## Çevrimiçi Derleyiciler

Bütün bu entegre geliştirme ortamlarının yanında online olarak da derleme yapılan web uygulamaları da bulunmaktadır. Bunlardan birkaçı aşağıda verilmiştir;

- [https://www.tutorialspoint.com/compile\\_c\\_online.php](https://www.tutorialspoint.com/compile_c_online.php)
- [https://www.onlinegdb.com/online\\_c\\_compiler](https://www.onlinegdb.com/online_c_compiler)
- <https://www.programiz.com/c-programming/online-compiler/>
- <https://onecompiler.com/c>
- [https://www.online-ide.com/online\\_c\\_compiler](https://www.online-ide.com/online_c_compiler)
- <https://onlinecompilers.com/online-c-compiler>

## Derleme Zamanı

Derleyici programının derleme işlemine başlayıp bitirildiği sürece **derleme zamanı** (**compile time**) denir. Bu süreç başarısızlıkla da sonuçlanabilir ve eğer derleme işleminde hata meydana gelirse programcı hata mesajları ile uyarılır.

Bir derleyici program, kaynak dosyayı makine diline çevirme çabasında, kaynak dosyanın C dilinin sözdizimi kurallarına uygunluğunu da denetler. Eğer dilin kurallarına uyulmamışsa, derleyici bu durumu bildiren bir ileti de vermek zorundadır.

```
int main()
{
    return
}
```

<sup>13</sup> <https://www.programiz.com/c-programming/getting-started>

Yukarıda hatalı yazılmış bir c programı derlendiğinde aşağıdaki hata alınacaktır.

```
main.c: In function 'main':
main.c:5:1: error: expected expression before '}' token
    5 | }
      | ^
```

Aşağıda derleme uyarısı alınacak, ancak derlemenin tamamlandığı bir kod örneği verilmiştir;

```
int main()
{
    return;
}
```

Bu durumda verilen derleme uyarıları ve programın çalışması aşağıda verilmiştir.

```
main.c: In function 'main':
main.c:4:5: warning: 'return' with no value, in function returning non-void
    4 |     return;
      |     ^~~~~~
main.c:1:5: note: declared here
    1 | int main()
      |     ^~~~

...Program finished with exit code 41
Press ENTER to exit console.
```

Kaynak kod içerisindeki metinlerde Türkçe karakter kullanılması halinde Windows altında derlenen programı çalıştırdığınızda aynı metni göremeyebilirsiniz. Bunu düzeltmek için komut satırında

```
C:\Users\ILHANOZKAN>Chcp 65001
```

Komutu verilerek komut satırının UTF-16 karakter kümesi ile işlem yapmasını sağlayabiliriz.

## Hatalar

Programcı tarafından kodlama yapılırken genellikle aşağıdaki üç tip hata yapılır;

1. **Derleme zamanı hataları** (**compile time error**): Bu tip hatalar genelde kullanılan dilin **yazım kurallarına** (**syntax**) uyulmadığından, **talimatların** (**statement**) yanlış yazılmasından ya da programcının kod metninde uygun olmayan karakterlerin kullanılmasından kaynaklanır.
2. **Koşma zamanı hataları** (**run time error**): Kaynak kod, kurallara uygun olarak yazılmıştır ve herhangi bir yazım hatası bulunmaz. Bu tip hatalara en iyi örneklerden birisi sıfıra bölme hatasıdır. Taşma hataları da bu hatalardandır. Daha sonra değinilecektir.
3. **Mantıksal hatalar** (**logical error**): Programcının çözüm için gerekli adımların oluşturulmasında, çözüm yönteminin yanlış olmasından ya da yanlış **işleçler** (**operator**) kullanılmasından kaynaklanır. Örneğin bir büyüktür (>) işleci yerine küçüktür (<) işleci kullanıldığında ne bir yazım hatası ne de bir çalışma zamanı hatası ortaya çıkar. Fakat program kendisinden istenilen davranışları yerine getirmez ve uygun çıktıları üretmez. Faktöriyel hesaplanırken **0! = 1** yerine **0! = 0** kabul etmek buna örnek verilebilir.

## C Programlama Kullanım Alanları

C dili, başlangıçta sistem geliştirme çalışmaları için, özellikle işletim sistemini oluşturan programlar için kullanıldı. **Montaj** (**assembly**) dilinde yazılan kod kadar hızlı çalışan kod ürettiği için sistem geliştirme dili olarak benimsendi. Bilgisayar mühendislerinin bilmesi gereken bu dilin birçok kullanım alanı mevcuttur;

- İşletim Sistemleri
- Dil Derleyicileri
- Assembler Derleyiciler
- Metin Düzenleyicileri
- Yazdırma Biriktiricileri

- Ağ Sürücüler
- Gömülü Programlar
- Veri tabanları

## C Programlama Dilinin Üstünlükleri

Verimlilik (**efficiency**) ve hız (**speed**): C, yüksek performanslı ve verimli olmasıyla bilinir. Düşük seviyede bellekle çalışmanıza olanak tanır ve donanıma doğrudan erişim sağlar, bu da onu hız ve ekonomik kaynak kullanımı gerektiren uygulamalar için ideal hale getirir.

Taşınabilir (**portable**): C programları, minimum veya hiç değişiklik yapılmadan farklı platformlarda derlenebilir ve yürütülebilir. Bu taşınabilirlik, dilin standartlaştırılmış olması ve derleyicilerin küresel olarak çeşitli işletim sistemlerinde kullanılabilmesinden kaynaklanmaktadır.

Donanıma Yakınlık: C, göstericiler ve düşük seviyeli işlemler kullanılarak donanımın doğrudan manipüle edilmesine olanak tanır. Bu, onu sistem programlama ve donanım kaynakları üzerinde ayrıntılı denetim gerektiren uygulamalar geliştirmek için uygun hale getirir.

Standart Kütüphaneler: Giriş/çıkış işlemleri, metin işleme ve matematiksel hesaplamalar gibi yaygın görevler için C, geliştiricilerin önceden oluşturulmuş işlevlerden yararlanarak daha verimli kod yazmalarına yardımcı olan büyük bir standart kütüphaneyle birlikte gelir.

Yapısal Programlama: C, kodu modüler ve anlaşılması kolay veri ve kontrol yapılarına sahiptir. Fonksiyonlar, döngüler ve koşullarla geliştiriciler, bakımı kolay net kod üretebilirler.

Yordamlı ya da Emreden Paradigma: C dilinde programlama fonksiyonların birbirini çağırması ile yapılır. Bu programlama bakış açısı **yordamlı programlama** (**procedural programing**) olarak adlandırılır. **Prosedür** (**procedure**), bir değer üretmeyen fonksiyonlardır. Fonksiyonlar içinde ise bilgisayarın gerçekleştireceği süreç, basit ve anlaşılır **talimatlar** (**statement**) halinde art arda yazılır. Talimatların art arda verilmesi **emreden programlama** (**imperative programming**) olarak da bilinir.

## C Dilinin Zayıf Yönleri

Manuel Bellek Yönetimi: C dili, bir geliştiricinin belleği açıkça ayırma ve ayırmayı kaldırmasıyla ilgilenmesi gereken manuel bellek yönetimine ihtiyaç duyar. Programcı bunu kodu yazarken yapar.

Nesne Yönelimli Olmaması: Günümüzde, programlama dillerinin çoğu **nesne yönelimli programlamayı** (**object oriented programming**) özelliklerini destekler. Ancak C dili bunu desteklemez.

Çöp Toplama Olmaması: C dili, bellekte yer ayrılan ve kullanılmayan bileşenleri bellekten otomatik kaldırmaz. Yani **çöp toplama** (**garbage collection**) kavramını desteklemez. Programcının belleği manuel olarak ayırması ve kaldırması gerekir ve bu hataya açık olabilir ve bellek sızıntılarına veya verimsiz bellek kullanımına yol açabilir.

İstisna İşleme Olmaması: C dili, **istisnaları** (**exception**) ya da **hata ayıklama** (**error handling**) işlemek için herhangi bir kütüphane sağlamaz. Programcının her türlü beklentiği işlemek için kod yazması gerekir.

## C Sürümlerinin Tarihçesi

1971 yılında *Dennis Ritchie*, C üzerinde çalışmaya başladı ve Bell Labs'daki diğer geliştiricilerle birlikte C'yi geliştirmeye devam etti. Geleneksel C'den sonraki C sürümlerinin tarihi aşağıda verilmiştir<sup>14</sup>.

K&R C – *Dennis Ritchie, Brian Kernighan* ile birlikte "The C Programming Language" adlı kitaplarının ilk baskısını 1978 yılında yayınladı. Halk arasında K&R olarak bilinen kitap, uzun yıllar boyunca dilin gayri resmi bir kılavuzu olarak hizmet etti. Açıkladığı C sürümüne genellikle "K&R C" denir ve ayrıca C78 olarak da adlandırılır.

<sup>14</sup> [https://www.tutorialspoint.com/cprogramming/c\\_history.htm](https://www.tutorialspoint.com/cprogramming/c_history.htm)

K&R C'de tanıtılan C dilinin birçok özelliği, 2018'de onaylanan dilin bir parçasıdır. C'nin erken sürümlerinde, yalnızca **int** dışındaki türleri döndüren fonksiyonlar, fonksiyon tanımı öncesinde kullanılıyorsa bildirilmelidir; önceden bildirilmeden kullanılan fonksiyonların **int** türünü döndürdüğü varsayılırdı.

AT&T ve diğer satıcılar tarafından üretilen C derleyicileri, K&R C diline eklenen çeşitli özellikleri destekledi. C popülerlik kazanmaya başlasa da farklılıklar problem yaşatmaya başladı. Bu nedenle, dil özelliklerinin standartlaştırılması gerektiği düşünüldü.

ANSI C – 1980'lerde, Amerikan Ulusal Standartlar Enstitüsü (ANSI), C dili için resmi bir standart üzerinde çalışmaya başladı. Bu, 1989'da standartlaştırılan ANSI C'nin geliştirilmesine yol açtı. ANSI C, birkaç yeni özellik tanıttı ve dilin önceki sürümlerinde bulunan belirsizlikleri giderdi.

C89/C90 – ANSI C standardı uluslararası olarak kabul edildi ve C89 (veya onaylanma yılına bağlı olarak C90) olarak tanındı. Uzun yıllar boyunca derleyiciler ve geliştirme araçları için temel teşkil etti.

C99 – 1999'da ISO/IEC, C99 olarak bilinen C standardının güncellenmiş bir sürümünü onayladı. C standardı 1990'ların sonlarında daha da revize edildi.

C99, **satır içi fonksiyonlar** (**inline function**), karmaşık sayıları temsil eden karmaşık bir tür gibi çeşitli yeni veri türleri ve değişken uzunluklu diziler vb. gibi yeni özellikler sundu. Ayrıca // ile başlayan C++ tarzı tek satırlık yorumlar için destek ekledi.

C11 – 2011'de yayınlanan C11, C standardının bir başka önemli revizyonudur. C11 standardı, C'ye ve kütüphaneye yeni özellikler ekler ve **çoklu iş parçacığı** (**multi threading**) desteği, **anonim yapılar** (**anonym structure**) ve birleşimler ve geliştirilmiş Unicode desteği gibi özellikler sunar.

Tür genel makroları, anonim yapılar, geliştirilmiş Unicode desteği, atomik işlemler, çoklu iş parçacığı ve sınır denetimli işlevler içerir. C++ ile geliştirilmiş bir uyumluluğu vardır.

C17 – C17 standardı Haziran 2018'de yayınlanmıştır. C17, C programlama dili için geçerli standarttır. Bu standart revizyonu yeni özellikler getirilmemiştir. Sadece belirli teknik düzeltmeler ve C11'deki kusurlara yönelik açıklamalar yapar.

C18 – C standardının en son sürümü olan C18, 2018'de yayınlandı. C11'e kıyasla küçük revizyonlar ve hata düzeltmeleri içeriyor.

C23 – C23, 2024'te yayınlanması beklenen bir sonraki büyük C dili standardı revizyonunun gayri resmi adıdır. Bu revizyonda 14 yeni anahtar kelimenin tanıtılması bekleniyor.

C, basitliği, verimliliği ve çok yönlülüğü nedeniyle zamanla popülerliğini korudu. İşletim sistemleri, gömülü sistemler, uygulamalar ve oyunlar dahil olmak üzere çeşitli yazılımlar oluşturmak için kullanıldı. C'nin sözdizimi ve semantiği, C++, Java ve Python gibi farklı modern programlama dillerini de etkilemiştir.