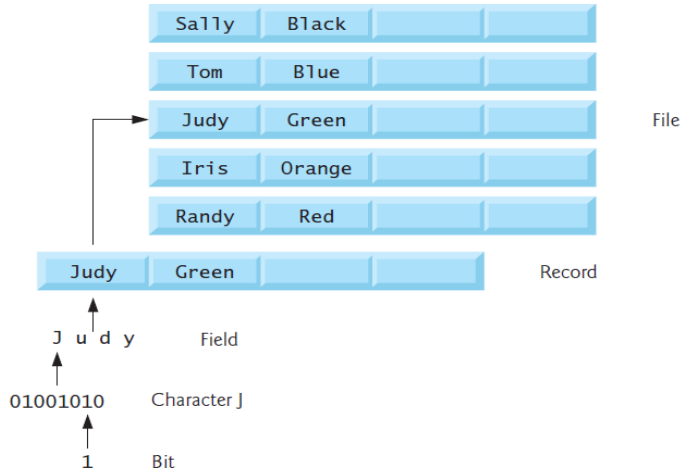


Veri Hiyerarşisi

Bilgisayarlar tarafından işlenen veri öğeleri, en basit veri öğelerinden ("bitler" olarak adlandırılır) karakterler ve alanlar gibi daha zengin olanlara doğru ilerledikçe yapı olarak daha büyük ve daha karmaşık hale gelen bir veri hiyerarşisi oluşturur. Aşağıdaki diyagram, veri hiyerarşisinin bir bölümünü göstermektedir:



bit

Bit, "ikili rakam"ın (iki değerden birini alabilen bir rakam) kısaltmasıdır ve bilgisayarın en küçük veri öğesidir. 0 veya 1 değerine sahip olabilir. Bilgisayarların etkileyici fonksiyonları, yalnızca 0'lar ve 1'lerin en basit manipülasyonlarını içerir; bir bitin değerini incelemek, bir bitin değerini ayarlamak ve bir bitin değerini tersine çevirmek (1'den 0'a veya 0'dan 1'e). Bitler, daha önce tartıştığımız ikili sayı sisteminin temelini oluşturur.

Karakterler

Düşük seviyeli bit biçimindeki verilerle çalışmak sıkıcıdır. Bunun yerine insanlar ondalık basamaklarla (0–9), harflerle (A–Z ve a–z) ve aşağıdaki gibi özel simgelerle çalışmayı tercih eder:

\$ @ % & * () - + " : ; , ? /

Rakamlar, harfler ve özel semboller karakter olarak bilinir. Bilgisayarın karakter seti, programları yazmak ve veri öğelerini temsil etmek için kullanılan karakterleri içerir. Bilgisayarlar yalnızca 1'leri ve 0'ları işler, bu nedenle bir bilgisayarın karakter seti her karakteri 1'ler ve 0'lardan oluşan bir model olarak temsil eder. C, varsayılan olarak ayarlanan ASCII (Bilgi Değişimi için Amerikan Standart Kodu) karakter kümesini kullanır. C ayrıca bir, iki, üç veya dört bayttan (sırasıyla 8, 16, 24 veya 32 bit) oluşan Unicode® karakterlerini destekler. Unicode dünya dillerinin çoğu için karakterler içerir. ASCII, Unicode'un harfleri (a–z ve A–Z), rakamları ve bazı genel özel karakterleri temsil eden (küçük) bir alt kümesidir. Unicode'un ASCII alt kümesini şu adreste görüntüleyebilirsiniz:

<https://www.unicode.org/charts/PDF/U0000.pdf>

Tüm diller, semboller, emojiler ve daha fazla uzun Unicode çizelgeleri için şu adresi ziyaret edin:

<http://www.unicode.org/charts/>

Ayrıca standart ASCII kod tablosuna şu linkten ulaşabilirsiniz.

<https://www.comfront.com/pages/ascii-chart>

Fields (Alanlar),

Karakterlerin bitlerden oluşması gibi, alanlar da karakterlerden veya baytlardan oluşur. Alan, anlamı ileten bir karakter veya bayt grubudur. Örneğin, büyük ve küçük harflerden oluşan bir alan bir kişinin adını, ondalık basamaklardan oluşan bir alan ise bir kişinin yıl cinsinden yaşını temsil edebilir.

Records (kayıtlar)

Bir kayıt oluşturmak için birkaç ilgili alan kullanılabilir. Örneğin bir bordro sisteminde, bir çalışanın kaydı aşağıdaki alanlardan oluşabilir (bu alanların olası türleri parantez içinde gösterilmiştir):

- Çalışan kimlik numarası (tam sayı).
- İsim (bir karakter grubu).
- Adres (bir karakter grubu).
- Saatlik ödeme oranı (ondalık noktalı bir sayı).
- Yılbaşından bugüne kazançlar (ondalık noktalı bir sayı).
- Kesilen vergi miktarı (ondalık noktalı bir sayı).

Bu nedenle, bir kayıt ilgili alanların bir grubudur. Yukarıda listelenen tüm alanlar aynı çalışana aittir. Bir şirketin birçok çalışanı ve her biri için bir bordro kaydı olabilir.

Files (Dosyalar)

Bir dosya, ilgili kayıtlardan oluşan bir gruptur. Daha genel olarak, bir dosya rasgele biçimlerde rasgele veriler içerir. Bazı işletim sistemleri bir dosyayı yalnızca bir bayt dizisi olarak görüntüler; bir dosyadaki baytların herhangi bir organizasyonu, örneğin verilerin kayıtlar halinde düzenlenmesi, uygulama programcısı tarafından oluşturulan bir görünümdür. Bunu nasıl yapacağınızı Bölüm 11, Dosya İşleme'de göreceksiniz. Bir kuruluşun, bazıları milyarlarca, hatta trilyonlarca bilgi karakteri içeren çok sayıda dosyası olması olağan bir durumdur. Aşağıda göreceğimiz gibi, büyük verilerle çok daha büyük dosya boyutları giderek daha yaygın hale geliyor.

Veri tabanları (Database)

Veritabanı, kolay erişim ve manipülasyon için düzenlenmiş bir veri koleksiyonudur. En popüler model, verilerin basit tablolarda saklandığı ilişkisel veri tabanıdır. Bir tablo kayıtları ve alanları içerir. Örneğin, bir öğrenci tablosunda ad, soyad, branş, yıl, öğrenci kimlik numarası ve not ortalaması alanları bulunabilir. Her öğrencinin verileri bir kayıttır ve her kayıttaki ayrı bilgi parçaları alanlardır. Verileri, birden çok tablo veya veri tabanı ile olan ilişkisine göre arayabilir, sıralayabilir ve başka şekillerde değiştirebilirsiniz. Örneğin, bir üniversite, öğrenci veri tabanındaki verileri dersler, kampüs içi barınma, yemek planları vb. veri tabanlarından gelen verilerle birlikte kullanabilir.

Büyük veri (Big Data)

Aşağıdaki tabloda bazı genel bayt ölçüleri gösterilmektedir:

Birim	Bayt	Yaklaşık değer
1 kilo byte (KB)	1024 byte	2^{10} byte (tam olarak 1024 bayt)
1 mega byte (MB)	1024 kilo byte	2^{20} byte
1 giga byte (GB)	1024 mega byte	2^{30} byte
1 tera byte (TB)	1024 giga byte	2^{40} byte
1 peta byte (PB)	1024 tera byte	2^{50} byte
1 eksa byte (EB)	1024 peta byte	2^{60} byte
1 zetta byte (ZB)	1024 eksa byte	2^{70} byte

Dünya çapında üretilen veri miktarı çok büyük ve büyümesi hızlanıyor. Büyük veri uygulamaları, büyük miktarda veriyle ilgilenir. Bu alan hızla büyüyor ve yazılım geliştiriciler için pek çok fırsat yaratıyor. Küresel olarak milyonlarca bilgi teknolojisi (BT) işi, halihazırda büyük veri uygulamalarını desteklemektedir.

Twitter®—Favori Büyük Veri Kaynağı

Geliştiriciler tarafından tercih edilen bir büyük veri kaynağı Twitter'dır. Günde (yıl 2020 de) yaklaşık 800.000.000 tweet atılmakta idi. Tweet'ler 280 karakterle sınırlı gibi görünse de, Twitter aslında tweet'leri analiz etmek isteyen programcılara tweet başına yaklaşık 10.000 bayt veri sağlamaktadır. Yani 800.000.000 çarpı 10.000, günde yaklaşık 8.000.000.000.000 bayt veya 8 terabayt (TB) veridir. Bu büyük veridir.

Tahmin, zorlu ve genellikle maliyetli bir süreçtir, ancak doğru tahminler için potansiyel ödüller harikadır. Veri madenciliği, bireyler ve kuruluşlar için değerli olabilecek iç görüleri bulmak için, genellikle büyük veriler olmak üzere kapsamlı veri koleksiyonları arasında arama yapma sürecidir. Tweet'lerden veri madenciliği yaptığınız duygu, seçim sonuçlarını, yeni bir filmin elde etmesi muhtemel gelirleri ve bir şirketin pazarlama kampanyasının başarısını tahmin etmenize yardımcı olabilir. Ayrıca şirketlerin rakiplerinin ürün tekliflerindeki zayıflıkları tespit etmesine yardımcı olabilir.

Makine Dilleri, Assembly Dilleri ve Üst Düzey Diller

Programcılar, bazıları bilgisayarlar tarafından doğrudan anlaşılabilen ve diğerleri ara çeviri adımları gerektiren çeşitli programlama dillerinde talimatlar yazarlar. Bugün bu tür yüzlerce dil kullanılıyor. Bunlar üç genel türe ayrılabilir:

- Makine dilleri (Machine languages)
- Derleme dilleri (Assembly languages)
- Üst düzey diller (High-level languages).

Makine dilleri (Machine languages)

Herhangi bir bilgisayar, yalnızca donanım tasarımı tarafından tanımlanan kendi makine dilini doğrudan anlayabilir. Makine dilleri genellikle, bilgisayarlara en temel işlemlerini birer birer gerçekleştirmeleri talimatını veren sayı dizilerinden (nihayetinde 1'lere ve 0'lara düşürülür) oluşur. Makine dilleri makineye bağlıdır—belirli bir makine dili yalnızca bir tür bilgisayarda kullanılabilir. Bu tür diller insanlar için çok karmaşıktır.

Assembly Dilleri ve Assembler

Makine dilinde programlama, çoğu programcı için çok yavaş ve sıkıcıydı. Programcılar, bilgisayarların doğrudan anlayabileceği sayı dizilerini kullanmak yerine, temel işlemleri temsil etmek için İngilizce benzeri kısaltmalar kullanmaya başladılar. Bu kısaltmalar, assembly dillerinin temelini oluşturdu. Assembly dili programlarını bilgisayar hızlarında makine diline çevirmek için derleyici (compiler) adı verilen çevirmen programları geliştirildi. Bu tür kodlar insanlar için daha anlaşılır olsa da, makine diline çevrilene kadar bilgisayarlar tarafından anlaşılabilir.

Üst Düzey Diller ve Derleyiciler

Assembly dillerinin ortaya çıkmasıyla birlikte bilgisayar kullanımı hızla arttı. Ancak programcılar, basit görevleri gerçekleştirmek için bile çok sayıda talimat kullanmak zorundaydı. Programlama sürecini hızlandırmak için, tek ifadelerin önemli görevleri yerine getirebileceği yüksek seviyeli diller geliştirildi. Tipik bir üst düzey dil programı, programın kaynak kodu olarak bilinen birçok ifade içerir.

Derleyici adı verilen çevirmen programları, üst düzey dil kaynak kodunu makine diline dönüştürür. Üst düzey diller, neredeyse günlük İngilizceye benzeyen ve ortak matematiksel gösterimler içeren yönergeler yazmanıza olanak tanır. Üst düzey bir dilde yazılmış bir bordro programı, aşağıdaki gibi tek bir ifade içerebilir:

`grossPay = basePay + overTimePay`

Programcının bakış açısından, yüksek seviyeli diller makine ve Assembly dillerine tercih edilir. C, dünyanın en yaygın kullanılan üst düzey programlama dilleri arasındadır.

Yorumlayıcılar (Interpreters)

Büyük bir üst düzey dil programını makine diline derlemek bilgisayarda önemli ölçüde zaman alabilir. Yorumlayıcılar, üst düzey dil programlarını doğrudan yürütür. Yorumlayıcılar derleme gecikmelerinden kaçınır, ancak kodunuz derlenmiş programlardan daha yavaş çalışır. Java ve Python gibi bazı programlama dilleri, programları çalıştırmak için akıllı bir derleme ve yorumlama karışımı kullanır.

İşletim sistemleri (Operating System)

İşletim sistemleri, kullanıcılar, yazılım geliştiriciler ve sistem yöneticileri için bilgisayar kullanımını kolaylaştıran yazılımlardır. Uygulamaların birbirleriyle güvenli, verimli ve eşzamanlı olarak çalışmasına izin veren hizmetler sağlarlar. Çekirdek işletim sistemi bileşenlerini içeren yazılıma çekirdek (kernel) denir. Linux, Windows ve macOS popüler masaüstü bilgisayar işletim sistemleridir; C programlamada bunlardan herhangi birini kullanabilirsiniz. Her biri kısmen C ile yazılmıştır. Akıllı telefonlarda ve tabletlerde kullanılan en popüler mobil işletim sistemleri Google'ın Android'i ve Apple'ın iOS'udur.

Windows—Tescilli Bir İşletim Sistemi

1980'lerin ortalarında Microsoft, kullanıcıların komut yazarak etkileşime girdiği son derece popüler bir kişisel bilgisayar işletim sistemi olan DOS'un (Disk İşletim Sistemi) üzerine inşa edilmiş bir grafik kullanıcı arabiriminden oluşan Windows işletim sistemini geliştirdi. Windows tescilli bir işletim sistemidir; yalnızca Microsoft tarafından denetlenir. Açık ara dünyanın en yaygın kullanılan masaüstü işletim sistemidir.

Linux—Açık Kaynaklı Bir İşletim Sistemi

Linux işletim sistemi, açık kaynak hareketinin en büyük başarılarından biridir. Satış veya kiralama için tescilli yazılımlar, yazılımların ilk yıllarında baskındı. Açık kaynak ile bireyler ve şirketler, yazılımın geliştirilmesine, sürdürülmesine ve geliştirilmesine katkıda bulunur. Daha sonra herkes bu yazılımı kendi amaçları için kullanabilir - normalde ücretsiz, ancak çeşitli lisanslama gereksinimlerine tabidir. Açık kaynak kodu, genellikle özel mülk yazılımdan çok daha geniş bir kitle tarafından incelenir, bu nedenle hatalar daha hızlı kaldırılabilir ve bu da yazılımı daha sağlam hale getirir. Açık kaynak üretkenliği artırır ve inovasyon patlamasına katkıda bulunur. Açık kaynak topluluğunda birçok kuruluş vardır. Bazı önemli olanlar:

. GitHub (açık kaynak projelerini yönetmek için araçlar sağlar; geliştirme aşamasında olan milyonlarcası vardır).

- Apache Software Foundation (başlangıçta Apache web sunucusunun yaratıcıları) şimdi birkaç büyük veri altyapı teknolojisi de dahil olmak üzere 350'den fazla açık kaynaklı projeyi yönetiyor.
- Eclipse Foundation (Eclipse Entegre Geliştirme Ortamı, programcıların uygun bir şekilde yazılım geliştirmesine yardımcı olur).
- Mozilla Foundation (Firefox web tarayıcısının yaratıcıları).
- OpenML (makine öğrenimi için açık kaynaklı araçlara ve verilere odaklanır).
- OpenAI (yapay zeka üzerine araştırma yapan ve yapay zeka takviyeli öğrenme araştırmalarında kullanılan açık kaynak araçlarını yayınlayan).
- OpenCV (çeşitli işletim sistemlerinde ve programlama dillerinde kullanılabilen açık kaynaklı bilgisayarlı görü araçlarına odaklanır).
- Python Software Foundation (Python programlama dilinden sorumlu).

Bilgi işlem ve iletişimdeki hızlı gelişmeler, azalan maliyetler ve açık kaynaklı yazılımlar, yazılım tabanlı işletmeler kurmayı on yıl öncesine göre çok daha kolay ve ekonomik hale getirdi. Harika bir örnek, bir üniversite yurt odasından başlatılan ve açık kaynaklı yazılımla oluşturulan Facebook'tur. Linux çekirdeği, en popüler açık kaynaklı, serbestçe dağıtılan, tam özellikli işletim sisteminin çekirdeğidir.

Gönüllülerden oluşan bir ekip tarafından geliştirildi ve sunucularda, kişisel bilgisayarlarda ve gömülü sistemlerde popüler oldu (akıllı telefonların, akıllı TV'lerin ve otomobil sistemlerinin kalbindeki bilgisayar sistemleri gibi).

Microsoft'un Windows ve Apple'ın macOS kaynak kodunun aksine, Linux kaynak kodu inceleme ve değişiklik için halka açıktır ve ücretsiz olarak indirilip kurulabilir. Sonuç olarak, Linux kullanıcıları, çekirdeği (kernel) aktif olarak hata ayıklayan ve iyileştiren büyük bir geliştiriciler topluluğundan ve işletim sistemini belirli ihtiyaçları karşılayacak şekilde özelleştirme yeteneğinden yararlanır.

C Programlama Dili

C, önceki iki dilden, BCPL ve B'den evrimleşmiştir. BCPL, 1967'de Martin Richards tarafından işletim sistemleri ve derleyiciler yazmak için bir dil olarak geliştirildi. Ken Thompson, B dilindeki pek çok özelliği BCPL'deki benzerlerinden sonra modelledi ve 1970'te Bell Laboratuvarlarında UNIX işletim sisteminin ilk sürümlerini oluşturmak için B'yi kullandı. C dili, Bell laboratuvarlarında Dennis Ritchie tarafından B'den geliştirildi ve ilk olarak 1972'de uygulandı. C başlangıçta UNIX işletim sisteminin geliştirme dili olarak geniş çapta tanındı. Günümüzün önde gelen işletim sistemlerinin çoğu C ve/veya C++ ile yazılmıştır. C çoğunlukla donanımdan bağımsızdır; dikkatli bir tasarımla çoğu bilgisayara taşınabilir C programları yazmak mümkündür. C, işletim sistemleri, gömülü sistemler, gerçek zamanlı sistemler ve iletişim sistemleri gibi performans gerektiren sistemleri geliştirmek için yaygın olarak kullanılır:

Uygulama	Açıklaması
İşletim sistemleri	C'nin taşınabilirliği ve performansı, Linux gibi işletim sistemlerini ve Microsoft'un Windows ve Google'ın Android'inin bazı bölümlerini uygulamak için cazip hale getirir. Apple'ın macOS'u, C'den türetilen Objective-C'de yerleşiktir.
Gömülü sistemler	Her yıl üretilen mikroişlemcilerin büyük çoğunluğu genel amaçlı bilgisayarlar dışındaki cihazlarda gömülü olarak kullanılmaktadır. Bu gömülü sistemler, navigasyon sistemlerini, akıllı ev aletlerini, ev güvenlik sistemlerini, akıllı telefonları, tabletleri, robotları, akıllı trafik kavşaklarını ve daha fazlasını içerir. C, tipik olarak olabildiğince hızlı çalışması ve bellek tasarrufu yapması gereken gömülü sistemler geliştirmek için en popüler programlama dillerinden biridir.
Gerçek zamanlı sistemler	Gerçek zamanlı sistemler genellikle neredeyse anlık ve öngörülebilir yanıt süreleri gerektiren "görev açısından kritik" uygulamalar için kullanılır. Gerçek zamanlı sistemlerin sürekli çalışması gerekir. Örneğin, bir hava trafik kontrol sistemi, uçakların konumlarını ve hızlarını sürekli olarak izlemeli ve bu bilgiyi hava trafik kontrolörlerine gecikmeden rapor etmelidir, böylece bir çarpışma olasılığı varsa uçakları rotasını değiştirmeleri için uyarabilirler.
Haberleşme sistemleri	İletişim sistemlerinin, ses ve video gibi şeylerin sorunsuz ve gecikmeden iletilmesini sağlamak için büyük miktarda veriyi hedeflerine hızlı bir şekilde yönlendirmesi gerekir.

1970'lerin sonlarında C, şimdi "geleneksel C" olarak adlandırılan şeye dönüştü. Kernighan ve Ritchie'nin The C Programming Language adlı kitabının 1978'de yayınlanması, dile büyük dikkat çekti. Bu, tüm zamanların en başarılı bilgisayar bilimi kitaplarından biri haline geldi.

Standardizasyon

C'nin çeşitli donanım platformlarına (yani bilgisayar donanımı türlerine) hızlı bir şekilde yayılması, birçok benzer ancak genellikle uyumsuz C sürümlerine yol açtı. Bu, birkaç platform için kod geliştirmesi gereken programcılar için ciddi bir sorundu. Standart bir C versiyonuna ihtiyaç olduğu anlaşıldı. 1983'te, Bilgisayar ve Bilgi İşleme İlişkin Amerikan Ulusal Standartlar Komitesi (X3), "dilini açık ve makineden bağımsız bir tanımını sağlamak" için X3J11 teknik komitesini oluşturdu. 1989'da standart, Amerika Birleşik Devletleri'nde Amerikan Ulusal Standartlar Enstitüsü (American National Standards Institute-ANSI) aracılığıyla, ardından dünya çapında International Standartlar Organizasyonu (ISO) aracılığıyla onaylandı. Bu sürüme basitçe Standart C adı verildi.

C11 ve C18 Standartları

2011'de onaylanan (C11 olarak anılan) ve 2018'de hata düzeltmeleriyle güncellenen (C18 olarak anılan) en son C standardıdır. C, donanımdan bağımsız, yaygın olarak kullanılabilen bir dil olduğundan, C uygulamaları genellikle çok çeşitli bilgisayar sistemlerinde çok az değişiklik veya hiç değişiklik yapmadan çalışabilir.

Standart C Kütüphanesi ve Açık Kaynak Kütüphaneleri

C programları, fonksiyon adı verilen parçalardan oluşur. Bir C programı oluşturmak için ihtiyacınız olan tüm fonksiyonları programlayabilirsiniz. Bununla birlikte, çoğu C programcısı, C standart kütüphanesinde var olan zengin fonksiyon koleksiyonundan yararlanır. Bu nedenle, C programlamayı öğrenmenin gerçekten iki bölümü vardır:

- C dilinin kendisini öğrenmek ve
- C standart kütüphanesindeki fonksiyonların nasıl kullanılacağını öğrenmek.

C'de programlama yaparken, genellikle aşağıdaki yapı taşlarını kullanırsınız:

- C standart kütüphane fonksiyonları,
- açık kaynak C kütüphane fonksiyonları,
- kendi oluşturduğunuz fonksiyonlar ve
- (güvendiğiniz) diğer kişilerin oluşturduğu ve kullanımınıza sunduğu fonksiyonlar.

Kendi fonksiyonlarınızı oluşturmanın avantajı, tam olarak nasıl çalıştıklarını bilmenizdir. Dezavantajı, yeni fonksiyonların tasarlanması, geliştirilmesi, hatalarının ayıklanması ve performans ayarının yapılması için harcanan zaman alıcı çabadır. Ders boyunca, program geliştirme çabalarınızdan yararlanmak ve "tekerleği yeniden icat etmekten" kaçınmak için mevcut C standart kitaplığını kullanmaya odaklanıyoruz. Buna yazılımın yeniden kullanımı denir. Kendi sürümlerinizi yazmak yerine C standart kitaplık fonksiyonlarını kullanmak program performansını artırabilir, çünkü bu fonksiyonlar verimli bir şekilde çalışmak için dikkatlice yazılmıştır. Kendi karşılaştırılabilir sürümlerinizi yazmak yerine C standart kitaplık fonksiyonlarını kullanmak da program taşınabilirliğini iyileştirebilir.

Açık Kaynak Kütüphaneleri

Makul miktarda kodla önemli görevleri gerçekleştirmenize yardımcı olabilecek çok sayıda üçüncü taraf ve açık kaynaklı C kütüphanesi vardır. GitHub, C kategorisinde 32.000'den fazla depoyu listeler:

<https://github.com/topics/c>

Ayrıca, Müthiş- C gibi sayfalar çok çeşitli uygulama alanları için popüler C kütüphanelerinin derlenmiş listelerini sağlar.

<https://github.com/kozross/awesome-c>

Diğer Popüler Programlama Dilleri

Aşağıdakiler, diğer bazı popüler programlama dillerinin kısa bir tanımı verilmektedir:

- BASIC, 1960'larda Dartmouth College'da acemileri programlama tekniklerine alıştırmak için geliştirildi. En son sürümlerinin çoğu nesne yönelimlidir.
- C'yi temel alan C++, Bjarne Stroustrup tarafından 1980'lerin başında Bell Laboratuvarlarında geliştirildi. C++, C dilini geliştiren özellikler sağlar ve nesne yönelimli programlama yetenekleri ekler.
- Python, 1991 yılında halka açık olarak piyasaya sürülen nesne yönelimli bir dildir. Amsterdam'daki Ulusal Matematik ve Bilgisayar Bilimleri Araştırma Enstitüsü'nden Guido van Rossum tarafından geliştirilmiştir. Python, özellikle eğitim ve bilimsel bilgi işlem için hızla dünyanın en popüler programlama dillerinden biri haline geldi ve 2017'de en popüler veri bilimi programlama dili olarak R programlama dilini geride bıraktı. Python'un popüler olmasının bazı nedenleri: Açık kaynaklı, ücretsiz ve geniş çapta erişilebilir. Devasa bir açık kaynak topluluğu tarafından destekleniyor. Öğrenmesi nispeten kolaydır. Kodunun okunması, diğer birçok popüler programlama dilinden daha kolaydır. Kapsamlı standart kitaplıklar ve binlerce üçüncü taraf açık kaynaklı kitaplık ile geliştirici üretkenliğini artırır. Kısmen veri bilimi ile olan özel ilişkisi nedeniyle, web geliştirme ve yapay zeka alanında popülerdir. Finans camiasında yaygın olarak kullanılmaktadır.
- Java—Sun Microsystems 1991'de James Gosling liderliğindeki dahili bir kurumsal araştırma projesini finanse etti ve bunun sonucunda Java adı verilen C++ tabanlı nesne yönelimli programlama dili ortaya çıktı. Önemli bir Java hedefi, geliştiricilerin çok çeşitli bilgisayar sistemlerinde çalışan programlar yazmasını sağlayan "bir kez yaz, her yerde çalıştır"dır. Java, kurumsal uygulamalarda, web sunucularında (web tarayıcılarımıza içerik sağlayan bilgisayarlar), tüketici cihazları (ör. akıllı telefonlar, tabletler, televizyon alıcı kutuları, cihazlar, otomobiller ve daha fazlası) için uygulamalarda ve birçok diğer amaçlar. Java başlangıçta tercih edilen Android uygulama geliştirme diliydi, ancak artık birkaç başka dil de destekleniyor.
- C# (C++ ve Java tabanlı), Microsoft'un üç ana nesne yönelimli programlama dilinden biridir; diğer ikisi Visual C++ ve Visual Basic'tir. C#, web'i bilgisayar uygulamalarına entegre etmek için geliştirildi ve şimdi birçok türde uygulama geliştirmek için yaygın olarak kullanılıyor. Microsoft'un birçok açık kaynak girişiminin bir parçası olarak, artık C# ve Visual Basic'in açık kaynak sürümlerini sunuyorlar.
- JavaScript, öncelikle web sayfalarına programlanabilirlik (ör. animasyonlar, kullanıcı etkileşimi ve daha fazlası) eklemek için kullanılan, yaygın olarak kullanılan bir betik dilidir. Tüm büyük web tarayıcıları bunu destekler. Birçok Python görselleştirme kitaplığı, web tarayıcınızda görüntüleyebileceğiniz etkileşimli görselleştirmeler oluşturmak için JavaScript çıktısı alır. NodeJS gibi araçlar, JavaScript'in web tarayıcılarının dışında çalışmasını da sağlar.
- 2014 yılında tanıtılan Swift, Apple'ın iOS ve macOS uygulamaları geliştirmek için kullandığı programlama dilidir. Swift, Objective-C, Java, C#, Ruby, Python ve diğer dillerin popüler özelliklerini içeren çağdaş bir dildir. Swift açık kaynaklıdır, dolayısıyla Apple dışı platformlarda da kullanılabilir.

R, istatistiksel uygulamalar ve görselleştirme için popüler bir açık kaynaklı programlama dilidir. Python ve R, en yaygın kullanılan iki veri bilimi dilidir.

Tipik C Programı Geliştirme Ortamı

C sistemleri genellikle birkaç bölümden oluşur: program geliştirme ortamı, dil ve C standart kütüphanesi. Aşağıda tipik C geliştirme ortamını açıklamaktadır. C programları genellikle yürütülmek üzere altı aşamadan geçer: düzenleme, ön işleme, derleme, bağlantı kurma, yükleme ve çalıştırma.

1. Aşama bir dosyanın bir editör programında düzenlenmesinden oluşur: Microsoft Visual Studio ve Apple Xcode gibi C ve C++ Tümüleşik Geliştirme Ortamları (Integrated Development Environment-IDE'ler) tümleşik düzenleyicilere sahiptir. Editöre bir C programı yazarsınız, gerekirse düzeltmeler yaparsınız, ardından programı sabit disk gibi ikincil bir depolama aygıtında depolarsınız. C programı dosya adları .c uzantısıyla bitmelidir.

2. Aşamada programı derlemek için komut verirsiniz: Derleyici, C programını makine dili koduna çevirir (nesne kodu olarak da anılır). Bir C sisteminde, derleme komutu, derleyicinin çeviri aşaması başlamadan önce bir ön işlemci programını çağırır.

3. Aşamada, derleyici C programını makine dili koduna çevirir: Derleyici, dil kurallarını ihlal ettiği için bir ifadeyi tanıyamadığı zaman bir söz dizimi hatası oluşur. Derleyici, yanlış ifadeyi bulmanıza ve düzeltmenize yardımcı olmak için bir hata mesajı verir. C standardı, derleyici tarafından yayınlanan hata mesajlarının ifadesini belirtmez, bu nedenle sisteminizde gördüğünüz mesajlar diğer sistemlerde gördüğünüz mesajlardan farklı olabilir. Söz dizimi hataları ayrıca derleme hataları veya derleme zamanı hataları olarak da adlandırılır.

4. Sonraki aşama bağlama olarak adlandırılır: C programları tipik olarak standart kütüphaneler, açık kaynak kütüphaneler veya belirli bir projenin özel kütüphaneleri gibi başka yerlerde tanımlanan fonksiyonları kullanır. C derleyicisi tarafından üretilen nesne kodu, bu eksik kısımlar nedeniyle tipik olarak "delikler" içerir. Bir bağlayıcı, yürütülebilir bir görüntü (eksik parça olmadan) oluşturmak için bir programın nesne kodunu eksik fonksiyonların koduyla ilişkilendirir.

5. Bir sonraki aşama yükleme olarak adlandırılır: Bir programın yürütülebilmesi için işletim sisteminin onu belleğe yüklemesi gerekir. Yükleme yürütülebilir görüntüyü diskten alır ve belleğe aktarır. Programı destekleyen kütüphanelerden ek bileşenler de yüklenir.

6. Son aşamada, CPU'nun kontrolü altındaki bilgisayar, her seferinde bir komut olarak programı yürütür.

Yürütme Sırasında Oluşabilecek Sorunlar

Programlar her zaman ilk denemede çalışmayabilir. Önceki aşamaların her biri, tartışacağımız çeşitli hatalar nedeniyle başarısız olabilir. Örneğin, çalışan bir program sıfıra bölme girişiminde bulunabilir (tıpkı aritmetikte olduğu gibi bilgisayarlarda geçersiz bir işlem). Bu, bilgisayarın bir hata mesajı göstermesine neden olur. Daha sonra düzenleme aşamasına geri döner, gerekli düzeltmeleri yapar ve düzeltmelerin düzgün çalıştığını belirlemek için kalan aşamalardan tekrar geçersiniz.

Programlar çalışırken ortaya çıkan sıfıra bölme gibi hatalara çalışma zamanı hataları veya yürütme zamanı hataları denir. Sıfıra bölme genellikle, programın işini başarıyla gerçekleştirmeden hemen sonlandırılmasına neden olan ölümcül bir hatadır. Ölümcül olmayan hatalar, programların genellikle yanlış sonuçlara yol açarak tamamlanmasına sebep olur.

Visual Studio ile Bir C Uygulamasını Derleme ve Çalıştırma

Bu bölümde, Microsoft Visual Studio 2019 Community sürümünü kullanarak Windows üzerinde bir C programı çalıştıracaksınız. Visual Studio'nun çeşitli sürümleri mevcuttur. Bazı sürümlerde, sunduğumuz seçenekler, menüler ve talimatlar biraz farklılık gösterebilir. Bu noktadan itibaren, basitçe "Visual Studio" veya "IDE" diyeceğiz.

Adım 1: Kurulumunuzu Kontrol Etme: Henüz yapmadıysanız, IDE'yi yükleme ve kitabın kod örneklerini indirme talimatları için bu kitabın Başlamadan Önce bölümünü okuyun.

Adım 2: Visual Studio'yu Başlatma Başlat menüsünden Visual Studio'yu başlatın. Esc tuşuna basarak ilk Visual Studio penceresini kapatın. Visual Studio'yu sonlandıracağı için sağ üst köşedeki X'e tıklamayın. Dosya > Başlat Penceresi'ni seçerek bu pencereye istediğiniz zaman erişebilirsiniz. Bir menüden bir menü öğesini seçmeyi belirtmek için > kullanırız, bu nedenle Dosya > Aç, "Dosya menüsünden Aç menü öğesini seçin" anlamına gelir.

3. Adım: Bir Proje Oluşturma: Bir proje, bir uygulama oluşturan C kaynak kodu dosyaları gibi ilgili dosyalardan oluşan bir gruptur. Visual Studio, uygulamaları bir veya daha fazla proje içeren projeler ve çözümler halinde düzenler. Programcılar, büyük ölçekli uygulamalar oluşturmak için çoklu proje çözümleri kullanır. Örneklerimiz yalnızca tek proje çözümleri gerektirir. Kod örneklerimiz için, Boş bir Proje ile başlayacak ve ona dosyalar ekleyeceksiniz. Bir proje oluşturmak için:

3.1. Yeni bir proje oluştur iletişim kutusunu görüntülemek için Dosya > Yeni > Proje...'yi seçin.

3.2. C++, Windows ve Konsol etiketleriyle Boş Proje şablonunu seçin.

Visual Studio'nun bir C derleyicisi yoktur, ancak Visual C++ derleyicisi çoğu C programını derleyebilir. Burada kullandığımız şablon, bir Komut İstemi penceresindeki komut satırında çalışan programlar içindir. Visual Studio sürümünüze ve yüklü seçeneklere bağlı olarak, başka birçok proje şablonu olabilir. Şablon ara metin kutusunu ve altındaki açılır listeleri kullanarak seçimlerinizi filtreleyebilirsiniz. Yeni projenizi yapılandırın iletişim kutusunu görüntülemek için İleri'ye tıklayın.

3.3. Bir Proje adı ve Konum girin. Proje adı için (c_test) belirttik. Konum için, kullanıcı hesabınızın Belgeler klasöründe olduğunu varsaydığımız bu kitabın örnekler klasörünü seçtik. Yeni projenizi Visual Studio'da açmak için Oluştur'a tıklayın. Bu noktada, Visual Studio projenizi oluşturur, klasörünü C:\Users\YourUserAccount\Documents\examples konumuna yerleştirir. (veya belirttiğiniz klasör) ve Visual Studio'nun ana penceresini açar. C kodunu düzenlediğinizde, Visual Studio her dosyayı pencere içinde ayrı bir sekme olarak görüntüler. Visual Studio'nun sol veya sağ tarafına yerleştirilen Çözüm Gezgini, uygulamanızın dosyalarını görüntülemek ve yönetmek içindir.

Bulut

Bugün giderek daha fazla bilgi işlem "bulutta", yani yerel olarak masa üstünüzde, dizüstü bilgisayarınızda veya mobil cihazınızda değil, dünya çapında İnternet üzerinden dağıtılan yazılım ve veriler kullanılarak yapılıyor. Bulut bilgi işlem, herhangi bir zamanda ihtiyaçlarınızı karşılamak için bilgi işlem kaynaklarını artırmanıza veya azaltmanıza olanak tanır; bu, ara sıra meydana gelen yoğun talepleri karşılamak için yeterli depolama ve işlem gücü sağlamak üzere donanım satın almaktan daha uygun maliyetlidir. Bulut bilgi işlem ayrıca bu uygulamaları yönetme yükünü (yazılımı yükleme ve yükseltme, güvenlik, yedeklemeler ve olağanüstü durum kurtarma gibi) servis sağlayıcıya kaydırarak para tasarrufu sağlar. Günlük olarak kullandığınız uygulamalar, büyük ölçüde çeşitli bulut tabanlı hizmetlere bağımlıdır. Bu hizmetler, İnternet üzerinden birbirleriyle ve kullandığınız uygulamalarla iletişim kuran çok büyük bilgi işlem kaynakları (bilgisayarlar, işlemciler, bellek, disk sürücüler vb.) ve

veri tabanları kullanır. İnternet üzerinden kendisine erişim sağlayan bir hizmet, web hizmeti olarak bilinir.

Nesnelerin İnterneti (Internet of Things- IoT)

İnternet artık sadece bir bilgisayar ağı değil, bir Nesnelerin İnterneti (IoT). Bir cihaz, bir IP adresine ve İnternet üzerinden otomatik olarak veri gönderme ve bazı durumlarda alma yeteneğine sahip herhangi bir nesnedir. Bu tür cihazlar şunları içerir:

- geçiş ücretlerinin ödenmesi için alıcı vericili bir araba,
- bir garajda park yeri olup olmadığını izler,
- bir insana implante edilmiş bir kalp monitörü,
- su kalitesi monitörleri,
- enerji kullanımını bildiren bir akıllı sayaç,
- radyasyon dedektörleri,
- bir depodaki ürün izleyicileri,
- hareketinizi ve konumunuzu takip edebilen mobil uygulamalar,
- oda sıcaklıklarını hava tahminlerine ve evdeki aktiviteye göre ayarlayan akıllı termostatlar ve
- akıllı ev aletleri.

statista.com'a göre, 2025'te 75 milyardan fazla IoT cihazı olabilir.

Büyük Veri Ne Kadar Büyük?

Bilgisayar bilimcileri ve veri bilimcileri için veriler artık program yazmak kadar önemlidir. IBM'e göre günde yaklaşık 2,5 kilyon bayt (2,5 eksabayt) veri oluşturuluyor ve dünyadaki verilerin %90'ı son iki yılda oluşturuldu. IDC'ye göre, küresel veri kaynağı 2025 yılına kadar yılda 175 zettabayta (175 trilyon gigabayta veya 175 milyar terabayta eşit) ulaşacak.

Büyük Veri Analitiği

Veri analitiği, iyi gelişmiş bir disiplindir. "Veri analizi" terimi 1962'de ortaya çıktı, ancak insanlar binlerce yıldır, eski Mısırlılara kadar uzanan, istatistikleri kullanarak verileri analiz ediyor. Büyük veri analitiği daha yeni bir olgudur; "büyük veri" terimi 1987 civarında kullanılmaya başlandı.

Büyük verinin dört durumunu düşünün:

1. Hacim—dünyanın ürettiği veriler katlanarak artıyor.
2. Hız—verinin üretilme hızı, kuruluşlar arasında hareket etme hızı ve veri değişimlerinin hızı hızla artıyor.
3. Çeşitlilik—veriler eskiden alfa-nümerikti (yani alfabetik karakterler, rakamlar, noktalama işaretleri ve bazı özel karakterlerden oluşuyordu)—bugün aynı zamanda görüntüleri, sesleri, videoları ve sayıları hızla artan Nesnelerin İnterneti sensörlerinden gelen verileri de içeriyor (Evler, işyerleri, araçlar, şehirler ve daha fazlasından gelen veriler).

4. Doğruluk—verinin geçerliliği—tam ve doğru mu? Önemli kararlar alırken bu verilere güvenebilir miyiz? Bu gerçek mi? Çoğu veri artık çeşitli türlerde, olağanüstü hacimlerde ve şaşırtıcı hızlarda hareket ederek dijital olarak oluşturuluyor. Moore Yasası ve ilgili gözlemler, verileri ekonomik bir şekilde depolamamızı ve daha hızlı işlememizi ve taşımamızı sağladı - hem de zaman içinde katlanarak artan oranlarda. Dijital veri depolama kapasitesi o kadar geniş ve o kadar ucuz ve küçük hale geldi ki, artık oluşturduğumuz tüm dijital verileri rahat ve ekonomik bir şekilde tutabiliyoruz.

Veri Bilimi ve Büyük Veri Fark Yaratıyor

Veri bilimi alanı, fark yaratan önemli sonuçlar ürettiği için hızla büyüyor. Aşağıda veri bilimi ve büyük veri kullanım durumlarından bazıları şöyle sıralanabilir;

- anomali tespiti
- engelli insanlara yardım etmek
- otomatik altyazı
- beyin haritalama
- kanser teşhisi/tedavisi
- el yazısını sınıflandırmak
- Bilgisayar görüşü
- kredi notu
- suç Önleme
- CRISPR gen düzenleme
- mahsul verimi iyileştirme
- müşteri kaybı ve elde tutma
- müşteri hizmetleri temsilcileri
- siber güvenlik
- veri madenciliği
- veri görüntüleme
- teşhis ilacı
- dinamik sürüş rotaları
- dinamik ücretlendirme
- elektronik sağlık kayıtları
- duygu tespiti
- yüz tanıma
- dolandırıcılık tespiti
- oyun oynama
- sağlık sonucu iyileştirme
- insan genom dizilimi
- kimlik hırsızlığı önleme
- immünoterapi
- akıllı asistanlar
- Nesnelerin İnterneti (IoT) ve
- tıbbi cihaz izleme
- stok kontrolü
- dil çevirisi
- konum tabanlı hizmetler
- kötü amaçlı yazılım tespiti
- pazarlama analitiği

- doğal dil çevirisi
- yeni ilaçlar
- kişisel asistanlar
- kişiselleştirilmiş tıp
- kimlik avı önleme
- kirlilik azaltma
- hassas ilaç
- hastalık salgınlarını tahmin etmek
- sağlık sonuçlarını tahmin etmek
- hava durumuna duyarlı tahmin
- ürün satışları
- önleyici tıp
- hastalık salgınlarını önleme
- gayrimenkul değerlemesi
- öneri sistemleri
- araç paylaşımı
- risk minimizasyonu
- robo finansal danışmanlar
- enerji tasarrufu
- sürücüsüz arabalar
- duygu analizi
- Paylaşım ekonomisi
- benzerlik tespiti
- akıllı Şehirler
- akıllı evler
- Akıllı sayaçlar
- akıllı termostatlar
- akıllı trafik kontrolü
- sosyal grafik analizi
- spam algılama
- borsa tahmini
- metni özetlemek
- teletıp
- terör saldırısı önleme
- hırsızlık önleme
- trend tespit
- görsel ürün arama
- ses tanıma
- hava Durumu tahmini
- vb...