

Bilgisayar mimarisinden hastane mimarisine

Yrd. Doç. Dr. İlker Köse



İstanbul Üniversitesi Elektronik Mühendisliği Bölümü'nden 1999 yılında mezun oldu. Yüksek lisans ve doktorasını Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü'nde tamamladı. 2003-2009 arasında Sağlık Bakanlığı'nın Aile Hekimliği Bilgi Sistemi, Merkezi Hastane Randevu Sistemi ve Ulusal Sağlık Bilgi Sistemi (Sağlık-NET) gibi bilişim projelerinde danışman ve proje yöneticisi olarak çalıştı. Ardından sağlık sigorta sektöründe uluslararası bir şirkette (CGM) 5 yıl boyunca Ar-Ge Direktörü olarak görev yaptı. Çalışma alanları veri madenciliği ve yazılım mühendisliğidir. Halen Medipol Üniversitesi Teknoloji Transfer Ofisi Direktörü olarak görev yapan ve aynı üniversitede öğretim üyesi olan Köse, evlidir ve bir çocuk babasıdır.

Muayene veya tedavi olmak amacıyla hastaneye giden çoğu kişi gibi ben de kendimi zayıf ve tedirgin hissedirim. Tedaviye ihtiyaç duymam nedeniyle hissettiğim teslimiyetle sağlık profesyonelleri ile aramdaki bilgi asimetrisinin hissettirdiği tedirginlik birleşince tuhaf bir stresle geçer ziyaretlerim. Buna bir de medyada şahit olduğumuz ve neredeyse birbirine zıt iddialar öne sürerek tartışan hekimler aklıma geldiğinde endişem daha da artar. Bunların hiçbiri olmasaydı dahi, sağlıkla ilgili popüler tartışma konuları da sıradan bir hasta olarak kafamı karıştırmaya yeterdi doğrusu. Sağlık verilerinin mahremiyeti, hasta güvenliği, malpraktis, uzman hekim-genel pratisyen ikilemi, modern ve alternatif tıp yaklaşımlarının çekişmesi, kişiselleştirilmiş tıp ekolünün iddialı yükselişi vb. konular, benim gibi sıradan bir vatandaşın aydınlatmak yerine biraz daha sorgular hale getiren tartışmalar. Bütün bunları söylerken, saygıdeğer sağlık çalışanlarının özveri ve fedakârlığına hâle getirmek istemem; ancak bu koca sistemin sıradan bir yararlanıcısı olarak sağlık sistemimizin bana hissettirdikleri maalesef bunlar.

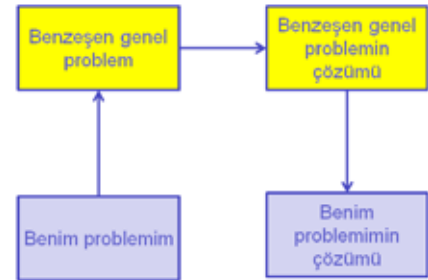
Son yıllarda beni biraz daha huzursuz eden diğer bir konu da hastanelerin giderek daha büyük kampüslere sahip olmaya başlaması. Ülkemiz için gurur verici modern ve kapsamlı tesislere kavuştuğumuz farkındayım; ancak binaların büyümesi, bir yandan da tedavi olmak isteyen bir hasta için sistemin biraz daha karmaşık hale gelmesi anlamına gelmiyor mu? En azından benim için öyle. Zira büyük hastane denince, 2003-

2004 yıllarında rahmetli babamın tedavisi sırasında Ankara Etlik SSK Hastanesinde yaşadıklarım gelir aklıma. SSK defterinin kapandığını ve pek çok reform sayesinde geçerliliği kalmayan bir örnek olduğunu biliyorum. Ama o koca hastanenin binaları ve katları arasında, kimi zaman imza attırmak, kimi zaman provizyon almak için dolaşırken her defasında kaybolurdum. Karıştırmayayım diye işlem sırasını ve yerini not almama rağmen, aylık ziyaretlerimde ya bir şeyler değişir ya da ben yönümü şaşırıp kaybolurdum. Refakatçi olarak zorlandığım bu işlemleri, hastaların veya yaşlı refakatçilerin ne kadar zorlukla yaptığını düşünerek daha kolay bir yolu olmalı diye kafa yordum. Sanıyorum biraz da o günlerin hatırası, bana şimdilerde inşa edilen büyük kampüslerin de benzer bir duyguyu yaşatmaya aday olduğunu düşündürüyor. Umarım yanılırim.

Diğer taraftan son yıllarda herkesin dilinde olan "hasta merkezli sağlık sistemi" yaklaşımının bu endişelerimi azaltacağına ümit ediyorum. "Sağlık sistemi" denince, bunun bir mühendislik alanı olduğunu düşündüğümü belirtmeden geçemeyeceğim. Zira dünyada son 20-30 yılda "sağlık sistem mühendisliği" adı altında çok sayıda doktora programının açıldığını dergimizin 25. sayısındaki "Sağlık Sistem Mühendisliği" başlıkla yazımda aktarmıştım (1). Dolayısıyla hasta merkezli bir sağlık sisteminin de analiz, tasarım/modelleme, geliştirme, test, işletme ve iyileştirme süreçleri itibarıyla bir mühendislik yaklaşımına muhtaç olduğunu söylemeliyim. Sağlık yöneticilerinin ve politika belirleyicilerin, merkez ve taşra teşkilat yapısından hastanelerin işleyişine kadar alınan

pek çok kararda dikkate aldıkları bu yaklaşımın, yıllardır mühendislik alanında geliştirmiş ve kullanılmış yöntemlerden yararlanması son derece önemlidir.

Aslına bakarsanız bahsettiğim yöntemler, yöntem haline gelmeden önce insanlığın ortak bilgi dağarcığında (literatür) ortaya çıkmış, etki ve faydası ölçülüp yayınlanmış, zaman içinde yaygınlık kazandıkça birer teknoloji (yöntem bilim, bilgi bilimi) halini almışlardır. Tüm yöntemler gibi, bir süre sonra da artık meslek branşlarının rutin yöntemi haline gelmişlerdir. Bilginin hayata geçişi sırasında bu değişim yaşanırken, bu durumun sadece belirli disiplinlere özgü gibi algılanması bir yanılgıdır aslında. Zira olan şudur: Benzer problemler için geliştirilmiş benzer çözümleri zaman içerisinde farklı disiplinlere uygulamaktayız. Bu nedenle disiplinler arası bu yöntemlerin transferi oldukça önemli ve bir o kadar da ilgi çekicidir. Öyle ki, yaratıcı düşünme teknikleri arasında oldukça popüler olan TRIZ yöntemi de bu yaklaşımı benimser (TRIZ Rusça adının kısaltmasıdır, İngilizcesi Theory of Inventive Problem Solving –TIPS, Türkçesi ise



Şekil: TRIZ ile çözüm bulma yöntemi



Yenilikçi Problem Çözme Teorisi olarak kullanılır) (2). TRIZ, 1946'da, patent ofisi çalışanları olan Genrich Saulovich Altshuller ve meslektaşları tarafından 3 milyondan fazla patentin incelenmesi sonucu geliştirilmiş mantığa ve verilere dayalı sorun kalıplarına odaklanan sistematik bir çözüm yöntemidir. Bu kadar çok patenti inceleyen bu araştırmacılar, farklı alanlarda yapılan binlerce yeniliğin aslında benzer/ortak çözüm yöntemlerini kullandığını fark etmişler ve jenerik çözüm yöntemlerini kendi projemize indirgemeyi yeni buluşlar için bir düşünme yöntemi olarak öne sürmüşler. Bu düşünme tekniği, şekilde de görüleceği üzere önce yaşanan lokal problemi daha genel bir probleme dönüştürür; sonra bu problem türü için başka disiplinler tarafından bulunmuş çözümlere odaklanır. Son olarak da bulunan bu çözümü kendi problemi için kullanılabilir hale getirir.

Acaba, hastanelerin çok büyük kampüsler haline gelerek daha karmaşık olmaları ile ilgili endişeme bu yaklaşımdan esinlenerek bazı çözümler bulabilir miyiz? Aslına bakarsanız uzun zamandır bu yöntemi dikkate alarak kendi uzmanlık alanındaki bilgileri esas alıp başka alanlardaki problemler için çözümler bulmaya çalışıyorum. Buna bir "takıntı" demek de mümkün. İtiraf ediyorum, bir bilgisayar mühendisi olarak pek çok meslektaşım gibi ben de günlük hayatımda karşılaştığım hemen her şeyle ilgili "modelleme/tasarım" ve "optimizasyon" yapmaya dair bir takıntı hastalığına sahibim. Doğrusu bundan çok da şikâyetçi değilim. Bütün takıntılar gibi bunun da tuhaf ve belki sağlıksız bir yönü mutlaka vardır. Ama bu riskin farkında olarak sa-

dece faydalı yönlerinden yararlanmaya gayret ediyorum ve olası zararlarını asgariye indirmeye çalışıyorum. Bu sayede günlük yaşamımda bu durumdan oldukça yararlandığımı söyleyebilirim. Bu takıntıyı edinmemde en büyük rol, mühendislikte okutulan bilgisayar mimarisi, işletim sistemleri, algoritma tasarımı, yazılım mühendisliği ve hatta insan-bilgisayar etkileşimi (kullanılabilirlik) dersleridir. Genel hatlarıyla söyleyecek olursam, bu derslerde bilgisayar donanımı, bu donanımla kullanıcı (kullanıcının kullandığı programlar) arasındaki etkileşimi sağlayan işletim sistemi, bunların çalışması sırasında kullanılan algoritmalar, uygulamaların geliştirilme aşamaları ve nihayet uygulamaların son kullanıcı için etkin ve verimli şekilde "kullanılabilir" olması için insan ve bilgisayar arasındaki etkileşim konuları ele alınır. Böylelikle donanımdan son kullanıcıya kadar tüm süreçlerde neler yapılıyor, kaynaklar nasıl optimize kullanılıyor, ihtiyaç nasıl analiz ediliyor ve çözüm nasıl üretiliyor gibi baştan sona tüm süreç ele alınır. Tüm bu derslerde probleme dair üstünde iyi çalışılmış optimal yaklaşımlar ele alınır.

Gerçek hayatta karşılaştığımız hemen her türlü yönetim problemi, şaşırtacak derecede bilgisayardaki unsurlarla benzerlik gösterir. Tüm kurumlarda bilgisayardaki donanıma karşılık gelecek ve belirli bir mimaride konuşlanmış bir takım kaynaklar vardır. Bu kaynakların belirli bir amaca uygun kullanılması sırasında en iyi sonucu verecek şekilde rol alması için bir işletim sistemine (prosedür vb.) ihtiyaç duyulur. Bu işletim sistemi, aynı kaynaklardan yararlanmak isteyen farklı birimlerin (uygulamaların) en etkin şekilde nasıl sıraya koyulacağını, işlemler arası

bir bağımlılık varsa bunun nasıl planlanacağını vb. ihtiyaçları yönetir. Kurumda oluşturulacak birimler de bilgisayardaki uygulamalara karşılık gelebilir. Hangi birimlerin oluşturulacağı, bu birimlerin ne üreteceği, diğer birimlerle nasıl iletişim ve etkileşim içinde olacağı, kurumun çalışma alanındaki ihtiyaçlarına göre belirlenir. Bu birimlerin hizmet verdiği iç veya dış müşterileri de bilgisayardan yararlanan kullanıcılara benzer. Bilgisayarda olduğu gibi bu birimlerin müşterilere dönük olan yüzlerinin amaca uygun, etkin ve verimli olması beklenir.

Kurumlarla bilgisayarları benzetmek, çok büyük kurumlar, devasa holdingler, hastane genel sekreterlikleri, vb. yapıların karmaşık yapısı dikkate alındığında basit bir benzetme gibi görünebilir. Öyleyse meseleyi biraz karmaşıklaştıralım... Kurumlar tek merkezden yönetilmeyebilir. Bugünkü hastanelerde kamu hastane birliklerinde olduğu gibi merkezde bir genel sekreter olsa da her hastanenin kendi başhekimisi ve yarı özerk yönetimi olabilir. Ancak birbirleri ile "aynı hedefler" için eşgüdüm içerisinde ve verimli şekilde çalışmalarını beklenir. Ayrıca müşteri türleri farklı olabilir. Acil hizmetler, ayaktan ve yatarak tedavi vb. hizmet türleri sunulabilir. Ancak bilgisayarlarda da bu duruma karşılık gelecek yaklaşımlar vardır. Paralel bilgisayar mimarisi ve bilgisayar ağları disiplinleri, birbiri ile aynı iş üzerinde çalışan çok işlemcili bilgisayarlarla farklı işler yapıp birbiriyle haberleşen bilgisayarları ele alırlar ve bu alanda pek çok mimari, haberleşme protokolü ve iş yönetim algoritmaları geliştirirler.

Yönetimin ve işletmenin birer bilim dalı

olduğuna inancım tamdır. Diğer taraftan insan faktörünün işletme biliminde önemli bir yeri olduğunu da biliyorum. Ancak bilgisayarlarda da en azından kullanıcı tarafında insan faktörü dikkate alınır. İşletme aşamasında ise tam anlamıyla rasyonel ve deterministik bir dünya vardır ki, bence bu bir eksiklik değil ideal bir kurumsal ortam tasarlamamıza yardımcı olacak bir ilham kaynağı ve modeldir. Ayrıca bu dünyadaki her çözüm, hesaplamalı bir modele dayalı olduğundan ölçülebilir ve ispatlanabilir. Uzun lafın kısısı, kurumlar bilgisayarlar gibidir ve bence bizim hastanelerimiz de bilgisayar gibi çalışmalıdır.

Hastanelerle bilgisayarın oldukça benzer olduğunu söyledim ama hala somut bir çözüm önerisinde bulunmadım. Bazı örnekler verdiğimizde belki ne demek istediğim daha iyi anlaşılabilir. Ancak baştan belirtiyim, söyleyeceklerimin pek çoğu için, “hastanelerde de zaten böyle veya böyle olmasına çalışılıyor” diyebilirsiniz. Ya da işletme ve yönetim alanında çalışanlar, bana bunların kendi disiplinlerindeki karşılıklarını örnek vererek bunların kendi disiplinlerinde de karşılığı olduğunu ve yeni şeyler olmadığını söyleyebilirler. Bunları saygıyla kabul ediyorum. Zira bu durum başlangıçtaki tezimi destekliyor zaten. TRIZ yönteminin önerdiği üzere problemlerimiz de çözümlerimiz de benzerdir. Vereceğim örnekler de doğrudan bu örnekleri model almakla ilgili olmayıp hastanelerimizin mimarisinden işletmesine kadar her alanda bütün olarak bilgisayarı model almamızın faydalı olacağını göstermek içindir.

Hastaneler İçin Mimari ve İşleyiş Önerileri

1) En sık kullandığımızı en yakımızda tutmalıyız:

Biz buna Yerellik Prensipli (principle of locality) diyoruz. Sanıyorum işletme bilimindeki “stok yönetimi” konusu da benzer paradigmayı esas alır. Bilgisayar bilimlerinde “yakında olmanın” iki türü vardır. Birincisi fiziksel olarak yakınlıktır ki bilgisayar mimarisi ile ilgilidir. Örneğin işlemcinin hesap yaparken kullanacağı sayısal değerleri saklayacak yere ihtiyacı vardır. Bunun için en sık kullanacağı değerleri işlemci entegre devresinin içindeki kayıtçılara (registers) ve ön belleğe (cache) kaydeder. Sonra biraz daha uzakta olan ana kart üzerindeki bellek (RAM) vardır. Daha az kullanacağı veriler gerektiğinde çekilmek üzere sabit disk veya CD/DVD disklerde bekletilir. Bu hiyerarşideki bellekleri, birinci, ikinci ve üçüncü bellek olarak adlandırıyoruz. Her bir seviyedeki belleğe fiziksel olarak erişim süresi, bir sonraki hiyerarşiye erişim süresinden yüzlerce veya binlerce

kez daha hızlıdır. Belleklerin teknolojileri, erişim süresi arttıkça pahalılaşır. Dolayısıyla alan olarak da işlemciye en yakın olanlar en pahalı ve en küçükleri iken, uzaklaştıkça alan maliyeti düşer ve alan miktarı gerektiği kadar büyütülebilir.

İkinci yakınlık konusu ise bu fiziksel alanlarda hangi verilerin saklanacağına karar verilmesidir. Bu da işletim sisteminin karar verdiği bir konudur. Eğer en yakınınızdaki fiziksel alanda hiç kullanmayacağınız ya da seyrek ihtiyaç duyduğunuz verileri saklıyorsanız bu fiziksel yakınlığın size faydası kısıtlı olacaktır. Dolayısıyla işletim sistemi hangi verilerin ne sıklıkla istendiğine bakar ve bilginin hangi bellek hiyerarşisinde duracağını belirler. Ayrıca istenen bir veriyi uzaktaki bir fiziksel yerden getirirken, hazır işlemciyi bekletmişken bu verinin yakınında bulunan diğer veriler de lazım olabilir diye fazladan getirir. Buna sayfalama (paging), bu yaklaşıma da yüzeysel yakınlık (spatial locality) diyoruz. Yani sık kullanılan bir veriye yakın yerdeyseniz sizin de çağırılma vaktiniz yakındır diye varsayılıyor. Sayfalama, her işletim sisteminin kullandığı ve verimliliği büyük miktarda artırdığı ispatlanmış bir yöntemdir.

Bu özet teknik bilgiden sonra, bilgisayarlardan ilham alarak hastanelerdeki görece küçük medikal cihazların optimal yerini hesaplamalı olarak tespit edebiliriz. Eşimin hamileliği sırasında farklı hastanelerde izlem yaptırmıştık. Kimi hastanelerde klinik odalarının arasında ultrason cihazı odası vardı. Hekim ön muayeneden sonra hastayı içerideki kapıdan yan odaya alır, muayenesini yapardı. Ultrason odası, diğer yan odadaki hekim tarafından da aynı şekilde kullanılırdı. Böylelikle işlem hızlı olur, hasta da az yorulurdu. Bazı hastanelerde ise, ultrason cihazı için başka bir kata yönlendirilirdik ve orada ayrı bir sıra beklerdik. Dahası, işlemin sonucunu tekrar aynı hekime gidip sonucu değerlendirmesini beklemek durumunda kalırdık.

Hasta gözünden baktığımızda hekimlerin sık kullandığı tüm cihazlar için benzer bir durum söz konusudur. Sağlık yönetimi mezunları, işletme veya endüstri mühendisleri, hastane içindeki her klinik için simülasyona dayalı optimizasyon hesabı yapabilirler. Böylelikle ilave cihazların maliyetini de dikkate alarak hem hasta memnuniyetini, hem de birim zamandaki işlem sayısını artıracak maliyet etkin bir yerleşim önerisinde bulunabilir. Her odaya cihaz koymak ile tüm hastane için bir cihaz kullanmak arasında bir yerde optimal çözümü bulabiliriz.

2) Hastanın hastanede geçirdiği süre minimum olmalıdır:

Bir hastanın hekimle birlikte geçirdiği zamanın uzun olması istenen bir durumken, hastane içinde harcadığı toplam sürenin uzun olması memnuniyeti azaltan bir faktördür. Bu süreyi azaltma adına bilgisayar bilimlerinden bazı çıkarımlar yapabilmek için önce bazı teknik bilgiler vermeliyim: Hastalar, bilgisayarda işlem görmek için bekleyen işlem parçalarına (transaction) veya işlemlere (process) benzerler. Bilgisayarlarda da bir işlemin tamamlanmak için uzun süre beklemesi istenmeyen bir durumdur. Bu nedenle çeşitli planlama (scheduling) algoritmaları kullanılır. Ancak hangi işlemde kaç tane geleceğini bilgisayar önceden bilemez. Kurumlarda da durum genelde aynıdır. Gelen işlem türü ve frekansına göre her planlama algoritmasının güçlü ve zayıf yönleri vardır. Bu nedenle performans ölçümü için birden fazla gösterge kullanılır. Açıkçası bilgisayar işlemcisinin performansını ölçmek için kullanılan göstergeler de hastanede kullandığımız kaynakların performansı için ilham vericidir. “İşlemcinin birim zamandaki adım sayısı”, “bir adımda kaç işlem yapabildiği” veya “belirli bir iş yükü altında ne kadar enerji harcadığı” gibi daha çok donanımsal kapasite ile ilgili olanları hariç tutarsak, hastaneler için önemli olduğunu düşündüğüm üç tane göstergesi vardır:

- İşlemcinin çalışma süresi (CPU time): İşlemci önemli ve merkezi bir kaynaktır. Boş kalmamalıdır. Bu gösterge işlemcinin boş kalmadan meşgul olduğu süreyi gösterir. Hastanedeki kaynakların bir işle meşgul olduğu süre buna benzetilebilir.

- İşlem hacmi (Throughput): Kimi durumda işlemci meşguldür, buna rağmen üretilen çıktı azdır. Bu gösterge, birim zamanda tamamlanan iş sayısını, yani çıktıyı ifade eder. Tedavi veya öneriyle gönderilen hasta sayısı da hastanenin işlem hacmidir.

- İşlem zamanı (process time): Bir işlemin işlemcide işlenmeye başlaması ile tamamlanması arasında geçen süredir. Hastanede hastanın geçirdiği süre, işlemin işlemcide geçirdiği süre gibidir (process time).

Eğer planlama algoritmamız bu üç göstergeyi de optimal hale getirebiliyorsa başarılıdır. Hastanelerde de tüm kaynakların azami miktarda meşgul olduğu, hastaların en az süre bekletildiği ve birim zamanda en çok miktarda iş yapıldığı durumda hastane “sayısal anlamda” başarılıdır.

Hastaneler için öneride bulunmadan önce, işlemcilerdeki planlama algoritmalarına bakalım. Planlama deyince aklımıza hemen klasik ilk gelen-ilk hizmet alır (first come-first served) algoritması



gelir. Kaynağın talebi karşılamaktan çok uzak olduğu durumlarda kaçınılmazdır. Bu yüzden minibüs/otobüs kuyrukları hep böyle çalışır. Ancak öncelikleri dikkate almadığı ve sırası gelene kadar kuyrukta bekleyen diğer işlemlere ilgi göstermediği için bilgisayar dünyasında da çok popüler değildir.

Kuyrukta bekleyen her işlemin gönlünü eşit miktarda alan algoritma “round robin” algoritmasıdır. İşlemcinin kuyrukta bekleyen işlemlerden sırayla eşit miktarda işlem yapmasını sağlar, sonra başa döner ve bu süreci tekrarlar. Bu sayede kuyruktaki tüm işlemler diğerleriyle eşit miktarda ilerleme kaydeder ve hiçbiri “ilgisizlikten” şikâyet etmez. Ancak bu algoritmada da öncelikli işler dikkate alınmaz. Yani kuyrukta bekleyenler eşit zamanda eşit miktar hizmet alır.

Acil ve öncelikli işleri dikkate almak için kullanılan algoritma ise “priority” algoritmasıdır. Bu algoritma kuyrukta bekleyen işlemleri önceliklerine göre ele alır ve tamamlar. Ancak kuyrukta bekleyen bazı işler, kuyruğa yeni giren ve daha öncelikli işler nedeniyle bir türlü işlemci tarafından muhatap kabul edilmez. Bu duruma açlıktan ölüme (starvation) adını veririz. “Priority” algoritmasının bir diğer kötü yönü de kimi zaman işlemcinin çok meşgul olmasına rağmen hiçbir çıktı üretememesine neden olmasıdır. Bunun temel sebebi işlemcinin üstünde çalıştığı bir işi bırakıp başka bir işe başlaması (anahtarlar- switching) aşamasında bir miktar zaman kaybedilir. Eğer bir işlemciye sürekli olarak elindeki işi bitirmeden daha öncelikli bir iş verilirse, işlemci sürekli meşgul görünür (CPU time yüksek çıkar); ama çıktı (throughput) oldukça düşer. Bu duruma da işi telef etmek (trashing) adını veririz ve hiç istenmeyen bir durumdur. Bu durum sistemsiz ve plansız çalışan kurumlarda pek çok beyaz yakalının başına gelen ve

iş tatminsizliğinin temel nedenlerinden biridir. Ancak dışarıdan ne tür işlerin geleceğini öngöremediğiniz operasyon odaklı yerlerde çalıştığınızda “priority” algoritması tek seçeneğiniz olabilir.

Diğer ilginç bir algoritma ise “bitimine en kısa süre kalanı öncele” (shortest remaining time first) algoritmasıdır. Kimi zaman kısa bir iş için uzun bir kuyruğa girmemiz gerektiğinde, kuyrukta bekleyenlerden “benim işim kısa öne geçebilir miyim?” diye izin alırsız ya, bu algoritma tam olarak bu mantığı kullanır. Bitmeye yakın işleri öne alır. Gerçek hayatta eğer bu şekilde kuyruğa kaynak yapanların sayısı artarsa, kuyrukta bekleyenler itiraz eder ve izin vermemeye başlarlar, aksi durumda onlara hiç sıra gelmeyecektir. Bu algoritmada da tahmin edebileceğiniz üzere, eğer kuyruğa sürekli tamamlanması kısa sürecek işler gelir ve öne geçerse, bitmesine çok uzun süre kalan bazı işler “priority” algoritmasında olduğu gibi yine açlıktan ölebilirler (starvation). Bu algoritmaların mutlaka birinin seçilmesi şart değildir. Çoğu işletim sistemi bunları birlikte/hibrit olarak kullanır. Daha yeteneklileri, gelen işlerin tür ve frekansına bakarak adaptif şekilde algoritmaları harmanlar.

Bu kadar teknik bilgiden sonra, hastanelerde hastaların geçirdiği süreyi azaltmak ve verimliliği artırmak için önerilerde bulunabiliriz. Bilgisayarlarda ideal bir performans, işlemcinin sürekli meşgul olduğu (CPU time), üretilen iş miktarının (throughput) yüksek olduğu ve işlemlerin tamamlanma sürelerinin (process time) az olduğu durumda ulaşılabilir. Dolayısıyla eğer hastanelerde kaynakların meşgulliyet süresi, hastaların toplam muayene süresi ve hastanede geçirdikleri zamanlar ölçülebilirse, planlama adına işletim sistemlerinin kullandığı algoritmalar adaptif şekilde uygulamaya alınabilir. Daha önemlisi,

daha işin başında hastane binaları tasarlanırken, çeşitli simülasyonlarla bu üç göstergelyi optimal hale getirecek bina tasarımı veya yerleşim düzeni yapılabilir. Mimarlar arasında “kullanılabilirlik” konusunun merkezi öneme sahip olduğunu biliyorum. Ancak bilgisayar bilimlerinde kullanılan gelişmiş ölçme yöntemlerinin bu yaklaşıma ek bir katkısı olacağını düşünüyorum.

Bu önerilere, insan-bilgisayar etkileşimi disiplininin “kullanılabilirlik” ile ilgili; paralel işleme (parallel computing) disiplininin iş paylaşımı ve senkronizasyon konusunda pek çok öneriler eklenebilir. Ancak sanırım fikir vermek adına bu kadarı yeterlidir. Bilgisayar bilimleri işletme problemleriyle gerçek hayattan bir adım önce karşılaşmış teknik çözümler üreten bir alandır. Üstelik bunu çok daha ölçülebilir ve ispatlanabilir yöntemlerle başarır. Maliyet faktörüne gelince... Bilgisayar mimarisi ve işletim sistemleri maliyet konusunda en titiz ve ince hesapların yapıldığı yerlerdir. Öyle ki ufak bir optimizasyon avantajı ile aynı performansı daha ucuza verebilen bir bilgisayar, rakiplerine göre önemli avantaj sağlayabilir. Sonuç olarak bilgisayar dünyasında optimizasyon her şeydir ve israfı yer yoktur. Dolayısıyla hastanelerimiz için bilgisayardan sadece mimari ve işletme ipuçlarını değil, bu kültürü de transfer edebiliriz.

Kaynaklar

1) İlker Köse, Sağlık Sistem Mühendisliği, Sağlık Düşüncesi ve Tıp Kültürü Dergisi, 25. sayı, s: 52-55, <http://sdplatform.com/Yazilar/Konuk-Yazilar/325/Saglik-sistem-muhendisligi.aspx> (Erişim Tarihi: 01.04.2016)

2) Hua, Z.; Yang, J.; Coulibaly, S.; Zhang, B. (2006). “Integration TRIZ with problem-solving tools: a literature review from 1995 to 2006”. *International Journal of Business Innovation and Research* 1 (1-2): 111-128.