

Search

فهرست نشریه پردازش

All Images Books Videos More

about 12 results (0.1 seconds)

سخن آغازین

۱

رایانش ابری دیگه چیه؟؟!!

۲

اپلیکیشن های کلود

۸

میکروسرویس ها

۱۰

UI/UX

۱۲

Javascript

۱۴

مصاحبه با دانشجوی موفق

۱۶

آشنایی با sdn [بخش دوم]

۲۲

RFID

۲۸

3D Sensors

۳۲



جهان سیم کارت

۳۴

تعمیرات مهندس : load نشدن سرچ در ویندوز 10

۳۷

فصل نامه علمی - دانشجویی پردازش  
پاییز 97  
صاحب امتیاز: انجمن علمی کامپیوتر  
کارشناس نشریه: زهرا وزیری  
استاد مشاور: دکتر محمدرضا کیوان پور  
مدیر مسئول: فاطمه زهرا ادیب  
سردبیر: مبینا پاک  
طراح جلد: ملیکا صباغیان

هیئت تحریریه: فاطمه زهرا ادیب - نسیم توحیدی - شبنم غلامشاهی - مبینا پاک - الهام بیدقی - زهرا قدسی - هما فراست - محمدحسن عباسی - میلاد الله یاری - نگار محبی - شیدا تارانی  
ویراستاری: پریناز میرباقری - فاطمه اسدیار - فاطمه دومهری - پرستو جعفری  
صفحه آرایی: سما کلانتری - ملیکا صباغیان - پریناز میرباقری - نازنین عباسی  
مقدم - شیوا کرمی - پرستو جعفری  
لیتوگرافی، چاپ و صحافی: چاپخانه دانشگاه الزهرا (س)

به نام خدا

همیشه زمانی که صحبت از سال تحصیلی جدید باشد، کلمات امید و اشتیاق در ذهن‌ام نقش می‌بندد...

زیبایی آغاز سال تحصیلی از نظر من، دیدن دانشجویان ورودی جدید است. نه اینکه زانم لال! چون سال آخری هستم آن‌ها را دست بیاندازم و به آن‌ها بخندم...! مسالهی قابل توجه در مورد این عزیزان "اندک ترم" برای من، این است که ورودی‌های جدید، خوشحال یا ناراحت از نتایج‌شان در آزمون زندگی (=کنکور!)، حریصانه در جستجوی شاهراه پیشرفت هستند... آن‌ها می‌دانند که بیل گیتس صرفاً با قبول شدن در دانشگاه، "بیل گیتس" نشده... (به علت بد آموزی اصل داستان را سانسور می‌کنیم...). آن‌ها می‌خواهند "خفن" بشوند... اما متاسفانه این سرمایه‌های فوق العاده، نمی‌دانند که چه راهی را باید طی کنند! به همین علت معمولاً زمانی که ترم اولی‌ها در دانشکده رویت می‌شوند، سال بالایی بی‌نوایی را پیدا کرده‌اند و وی سوال پیچ می‌نمایند... اما متاسفانه اکثراً به جای جواب، نصیحت‌هایی درباره‌ی اهمیت پاس کردن ریاضی یک و برنامه نویسی پیشرفته، دریافت می‌کنند (نصیحت ذکر شده را جدی بگیرید) ...

در این شماره از نشریه‌ی پردازش، من و دوستانم سعی کردیم با در نظر گرفتن دوستان جدید و با عرض خیر مقدم به آن‌ها! مطالبی با موضوعات مناسب برای آن‌ها در نظر بگیریم تا شاید بتونیم در رسیدن به اهداف متعالی‌شان، کمک کوچکی باشیم ...

دوستان عزیز پردازشی جدید می‌توانید در صورتی که به مشاوره احتیاج داشتید و یا سوالی در ذهن‌تان بود، به ما ایمیل بزنید... تا جایی که دانش‌اش را داشته باشیم به شما کمک می‌کنیم. اگر هم جواب را نداشتیم (ندانستن عیب نیست!) برایتان جواب یا راه رسیدن به آن را پیدا می‌کنیم. (جوینده یابنده است!)

ان شا... که به تمام اهداف زیبای‌تان برسید.. و "چه چیزی لذت بخش‌تر از انجام دادن کاری که دیگران می‌گویند نمی‌توانی..."

سال تحصیلی سرشار از شادی و موفقیت را برای شما دانشجویان عزیز، اساتید بزرگوار و کارمندان گرامی آرزومندیم

هم شاگردی سلام ...

مهر ماه ۹۷

مبینا پاک

سخن آغازین





مثال اگر از سرویس Dropbox استفاده می کنید کمتر پیش می آید که از مرورگرتان برای انتقال فایل ها استفاده کنید. اما در برای دسترسی جیمیل اغلب از مرورگر استفاده می کنید.

### چرا Cloud Computing؟

همانگونه که گفتیم این اصطلاح نامفهوم که در واقع نرم افزارهای تحت وب را توصیف می کند؛ مدت هاست که در اطراف ما وجود داشته اما بیشتر افراد آن را به این اسم نمی شناختند. پس به منظور رسیدن این بازار به شروع جدید و جایگزینی اپلیکیشن های تحت وب به جای نرم افزارهای وابسته به سرورهای میزبان، این اصطلاح به وجود آمده است.

توصیف اینکه چرا از کلمه «Cloud» یا «ابر» در این اصطلاح استفاده شده نیز بسیار ساده است. اگر توجه کرده باشید در نمودارهای شبکه های رایانه ای، اینترنت معمولاً به شکل ابر در تصویر نمایش داده می شود. شاید دلیل این تشبیه این است که اینترنت همانند یک ابر، جزئیات فنی اش را از دید کاربران مخفی نگه می دارد. بنابراین در واقع این عبارت راهی برای مشاوران و شرکت ها فراهم آورده تا خدمات خود را در بسته بندی های جدید به فروش بگذارند.

### رایانش ابری چگونه به ما کمک می کند؟

از آنجا که شرکت ها و برندهای تجاری در حال انتقال نرم افزارهای خود بر روی وب هستند و این اپلیکیشن ها روز به روز با ویژگی های جدید و جالبتری از طریق مرورگرها به معرض نمایش در می آیند؛ می توان گفت: به زودی قادر خواهیم بود که از هر مرورگر و با هر کامپیوتر و بدون وجود هیچگونه مرزی بین کامپیوتر شخصی مان و اینترنت به همه چیز دست پیدا کنیم.

سرویس Dropbox مثال عالی از رایانش ابری است که هم اکنون میلیون ها نفر از آن استفاده می کنند تا همیشه به فایل های مهم شان دسترسی

## رایانش ابری چیست؟!!!

امروزه اکثریت مردم در سراسر جهان بارها و بارها کلمه «Cloud computing» را می شنوند، اما بسیار اندک اند افرادی که معنای واقعی این اصطلاح را بدانند. در این مطلب سعی می کنیم معنی آن را به نحوی ساده تر و قابل فهم تر بیان کنیم.

### Cloud computing چیست؟

این اصطلاح که در فارسی بدان رایانش ابری گویند بر اساس تعاریف موسسه ملی فناوری و استانداردها، اینگونه تعریف شده: رایانش ابری مدلی است که بر حسب تقاضای شبکه، دسترسی آسان و فراگیر به مجموعه عظیمی از منابع محاسباتی قابل تنظیم (همانند شبکه ها، سرورها، فضای ذخیره سازی، برنامه های کاربردی و سرویس ها) را به سرعت و بدون دخالت سرویس دهنده به راحتی ممکن می سازد.

البته این تعریف بسیار کلی و تا حدودی گیج کننده است. پس تعریف ساده تر آن برای افراد عادی چیست؟

رایانش ابری = نرم افزارهای تحت وب به همین راحتی!

اگر شما از نرم افزارهای تحت وب که توسط سرویس دهندگانی همچون گوگل و مایکروسافت ارائه شده استفاده کنید، در واقع از رایانش ابری بهره مند می شوید. هر کدام از اپلیکیشن های مبتنی بر وب از جمله Gmail ، Google Calendar ، Hotmail ، Dropbox و Google Doc بر مبنای رایانش ابری تعریف شده اند. زمانی که شما به هر کدام از این سرویس ها متصل می شوید در واقع به منبع گسترده ای از سرورهای موجود در اینترنت متصل شده اید. نیاز نیست که کاربر حتماً از مرورگر وب برای رایانش ابری استفاده کند؛ اما ظاهراً همه چیز در حال حرکت به این سمت و سو است. برای



## • پزشکی:

در سال‌های اخیر، رایانش ابری با سرعت قابل توجهی به حوزه پزشکی نیز وارد شده است. منابع نامحدود ارائه شده توسط رایانش ابری و انعطاف‌پذیری ذاتی این فناوری، موجب شده تا از این فناوری برای توسعه و ارائه خدمات درمانی استفاده شود. علاوه بر این، از قدرت پردازش ابری در مطالعات پزشکی در زمینه‌هایی نظیر ژنتیک و پزشکی مولکولی نیز استفاده می‌شود.

ذخیره‌سازی اطلاعات بیماران به‌طور محلی، چه به‌صورت آرشیوهای فیزیکی قدیمی و چه به‌صورت الکترونیکی و در سرورهای داخلی، نه تنها هزینه‌بر است، بلکه پیچیدگی‌های خاصی را نیز برای مراکز پزشکی به‌همراه می‌آورد. به‌همین دلیل بیمارستان‌ها و مراکز پزشکی نیز در حال روی آوردن به ذخیره‌سازی ابری اطلاعات بیماران هستند، این مسئله باعث می‌شود پزشکان و بیماران بتوانند در هر لحظه و بدون مراجعه به مراکز پزشکی، به اطلاعات مورد نیاز خود دسترسی داشته باشند.



یکی از مثال‌های موفق در این زمینه، شبکه‌ای موسوم به **Health Hi Way** در هندوستان است. این شبکه با متصل کردن بیش از ۱۱۰۰ بیمارستان و ۱۰ هزار پزشک به یکدیگر، نه تنها اشتراک اطلاعات و انجام هماهنگی‌های مختلف میان مدیران، پزشکان و شرکت‌های فعال در حوزه پزشکی را ساده کرده، بلکه هزینه‌های پزشکی را نیز به‌مراتب کاهش داده است.

## • آموزش:

حوزه آموزش نیز همچون دیگر حوزه‌ها از مزایای رایانش ابری بی‌بهره نمانده است. رایانش ابری انتخابی مناسب برای آن‌دسته از موسسات آموزشی است که فاقد توان مالی کافی برای تهیه و نگهداری زیرساخت‌های اطلاعاتی هستند. علاوه بر این، به‌کمک رایانش ابری، دسترسی به خدمات آموزشی به‌صورت راه‌دور نیز آسان‌تر از گذشته شده است.

می‌توان از افزایش بهره‌وری زمانی، کاهش هزینه‌ها و ساده‌شدن رویه‌های آموزشی به‌عنوان مزایای اصلی رایانش ابری در حوزه آموزش یاد کرد. این مزایا، به‌ویژه برای کشورهای با چالش‌های بودجه‌ای روبه‌رو هستند، بسیار قابل توجه خواهد بود.

## معماری رایانش ابری

در حالت کلی و جامع‌ترین نوع معماری، رایانش ابری شامل ۵ لایه اصلی است.

## (۱) کاربر

## (۲) نرم افزار به عنوان سرویس

## (۳) بستر به عنوان سرویس

## (۴) زیرساخت به عنوان سرویس

## (۵) سرور

این ۵ لایه، معماری رایانش ابری را تشکیل می‌دهند که در ادامه به توضیح و مثال هر کدام خواهیم پرداخت.



## • کاربر:

کاربر یا **Client** در معماری پردازش ابری به معنای یوزر و یا فردی که از **Cloud** استفاده می‌کند، نیست.

کاربر در رایانش ابری متشکل از سخت افزار یا نرم افزاری است که برای تحویل برنامه‌های ابر فعالیت می‌کند و به طور ویژه وظیفه رساندن سرویس به دست یوزر را دارد.

## ویژگی‌های کاربر و نحوه شناسایی آن

به طور کلی به نرم افزار و سخت افزاری که به تنهایی کارایی نداشته و صرفاً یک سخت افزار یا نرم افزار خاموش است، کاربر گفته می‌شود.

از جمله مثال‌هایی که می‌توان برای کاربر نام برد:

(۱) **گوگل کروم** (مرورگر کروم بدون اتصال به ابر گوگل کاملاً بی استفاده خواهد بود)

(۲) **نرم افزار گوگل درایو** (بدون اتصال به ابر گوگل درایو بدون استفاده است)

(۳) **نرم افزار One Drive** (نرم افزار متصل شدن به فضای ابری مایکروسافت که بدون اتصال بی‌مصرف است)

در این بخش مثال‌های زیادی را می‌توان ذکر کرد که با شناختن نرم افزارهای کاربری بالا، شما می‌توانید سایر نرم افزارهای این لایه را شناسایی کنید.

## • نرم افزار به عنوان سرویس:

لایه بعدی لایه ایست با نام نرم افزار به عنوان سرویس که در انگلیسی آن را **Software as a Service** هم می خوانند و مخفف آن نیز **SaaS** است.

**SaaS**، نرم افزارها را به صورت یک سرویس بر روی اینترنت تحویل داده و بدین وسیله کاربر می تواند به آن متصل شده و از آن بهره ببرد. بدین شکل، سرویس دهی نرم افزار بر روی ابر بسیار آسان بوده و آپدیت، نگهداری و پشتیبانی به صورت متمرکز تنها بر روی ابر صورت می گیرد.

از ویژگی های اصلی نرم افزار به عنوان سرویس می توان به



موارد زیر اشاره کرد:

- تحویل نرم افزار به صورت مدل یک به چند (در این حالت یک نرم افزار در حال اجراست و چندین کاربر از آن استفاده می کنند).
- بروزرسانی و ارتقای نرم افزاری به صورت متمرکز بوده و نیازی نیست برای تمامی کاربران آپدیت و یا ارتقا صورت پذیرد.
- دسترسی و مدیریت نرم افزارهای تجاری

مثال SaaS:

**Google Labs**: یک آزمایشگاه برای تست های نرم افزاری متفاوت است.

**Salesforce.com**: یکی از گسترش دهندگان سیستم رایانش ابری است که نرم افزارهای سازمانی و دولتی تحت سرور را می تواند در بستر اینترنت در دسترس کاربران قرار دهد.

**Facebook**: فیسبوک برای سرویس دهی از یک اینترفیس اینترنتی استفاده می کند اما در اصل یک نرم افزار است که در حال اجرا شدن می باشد.

**You tube**: سایت یوتیوب در اصل یک نرم افزار قدرتمند است که در پس پرده فعالیت می کند و شما تنها لایه کاربر آن را می بینید که به صورت صفحه وب به نمایش در می آید.

## • بستر به عنوان سرویس:

بستر به عنوان سرویس لایه ایست که در انگلیسی آن را **PaaS** یا **Platform as a Service** می خوانند.

لایه **PaaS**، قسمتی از معماری رایانش ابری است که نرم افزار های لایه **SaaS** روی آن نصب می گردد و اجرای آن به عهده این لایه می باشد.

در تعریفی ساده می توان گفت این لایه، نرم افزارهای لایه **SaaS** را به عنوان یک سرویس بر روی بستر شبکه (اینترنت) ارائه می دهد.

بستر به عنوان سرویس (**PaaS**) در اصل یک بسته نرم افزاری جامع است که امکاناتی مانند تجمیع، تبادل پیغام، اطلاعات و میان افزار است که ارتباط بین لایه **SaaS** و **IaaS** را برقرار می کند.

مثال PaaS:

موتور نرم افزار **Google Apps**: این ابزار امکان اجرای برنامه های کاربردی توسط زیر ساخت های گوگل را می دهد.

پلتفرم هایی مانند **Google Apps** می توانند امکانات پایه قدرتمندی را در اختیار توسعه دهندگان نرم افزارها قرار دهند اما افزایش امکانات زیرساختی و ... در دست گوگل است که آن ها را افزایش دهد یا خیر.

سیستم عامل: **Azure**: سیستم عامل **Azure** میکروسافت یک **PaaS** است که می توان از آن برای توسعه نرم افزارها در سطح اینترنت استفاده نمود.

## • زیرساخت به عنوان سرویس:

لایه چهارم از معماری رایانش ابری است که به آن **IaaS** یا **Infrastructure as a Service** نیز می گویند.

سرویس زیرساخت رایانش ابری در اصل یک بستر مجازی است که به صورت سرویس ارائه می شود. کاربران به جای خرید سخت افزار، نرم افزار، فضای دیتاسنتر و یا تجهیزات شبکه می توانند تمام آنها را به صورت یک سرویس مجازی خریداری کرده و از آن استفاده کنند.

هزینه های پرداختی کاربران برای این سرویس عموماً به شکل رایانش همگانی و میزان منابع مصرف شده دریافت می گردد و هزینه پرداختی معادل هزینه ایست که سرویس شما فعالیت داشته است.

ارائه این لایه به مشتریان در اصل مدل پیشرفته و تکامل یافته عرضه سرورهای مجازی خصوصی است.

در مجموع می توان با **IaaS**، زیر ساخت کامپیوتر و محیط پلتفرم مجازی سازی را مانند یک سرویس در اختیار کاربران قرار داد.

رایانش ابری یا Cloud در واقع فراهم کردن محیط محاسباتی برای کاربران پایانی از راه دور است، به صورتی که نرم افزارها به عنوان یک سرویس بجای اینکه بر روی کامپیوترهای خود کاربران اجرا شوند، بر روی سرورهای قابل اعتماد و توسعه پذیر اجرا می شوند. محاسبات ابری می تواند چیزهای دیگری را هم شامل شود ولی معمولاً منظور اصلی، اجرای آیتهم های مختلف مانند نرم افزارها، پلتفرم ها و زیرساخت، به عنوان یک سرویس می باشد. این استک در واقع جزء دسته بندی سوم می شود و یک زیرساخت به عنوان سرویس یا [IaaS] محسوب می شود. فراهم کردن زیرساخت به این معنی است

که کاربران با استفاده از این استک، به آسانی می توانند نمونه های جدید اضافه کرده و مولفه های دیگر cloud بر روی آنها اجرا کنند. معمولاً، این زیرساخت یک پلتفرم را اجرا می کند که توسعه دهنده بر روی آن می تواند اپلیکیشن های نرم افزاری را ایجاد کرده و به کاربران پایانی خود خدمات ارائه دهد



### سرویس های open stack چیست؟

این استک از قسمت های اجرایی مختلفی ساخته شده است. بخاطر ذات آزاد بودن آن، هر شخصی می تواند مولفه های دیگری نیز به آن بیفزاید تا نیازهای خود را برآورده سازد. انجمن این استک، نه مولفه کلیدی را مشخص کرده است که در واقع بخشی از هسته این استک محسوب می شوند و به عنوان بخشی از هر سیستم این استکی توزیع می شوند و به صورت رسمی توسط انجمن این استک، پشتیبانی می شوند.

#### :Nova

Nova، موتور محاسباتی اصلی، برای این استک است. این موتور برای توسعه و مدیریت تعداد زیادی از ماشین های مجازی و نمونه های دیگر برای کنترل وظایف محاسباتی استفاده می شود.

#### :Swift

سوئیفت، یک سیستم ذخیره سازی اشیا و فایل هاست. بجای ایده قدیمی که برای بدست آوردن فایل ها به آدرس آنها بر روی دیسک ارجاع می داد، توسعه دهندگان می توانند به جای ارجاع به یک آدرس منحصر بفرد، به یک فایل یا بخشی از اطلاعات ارجاع دهند و به این استک این فرصت را بدهند که آنها را در هر جایی ذخیره کند. این کار، توسعه بخشی را آسان می کند و در این صورت توسعه دهندگان دغدغه ظرفیت یک سیستم منفرد و رای نرم افزار را نخواهند داشت.

#### :Cinder

Cinder یک مولفه ذخیره سازی بلاک است که شباهت بیشتری به مفهوم قدیمی دسترسی به مکان های خاص یک دیسک درایو دارد. این

از نمونه هایی که امروزه می توان به آن اشاره کرد سرویس **Sun Parascle, Azure Platform, Amazon EC2** می باشد.

### • سرور:

لایه سرور به سخت افزار و نرم افزارهایی اطلاق می شود که به صورت کامل وظیفه پردازش را در ابر به عهده دارند.

به عنوان نمونه می توان به پردازنده های چند هسته ای موازی و سوپر سرورهای پایه ابر اشاره کرد.

### openstack چیست؟

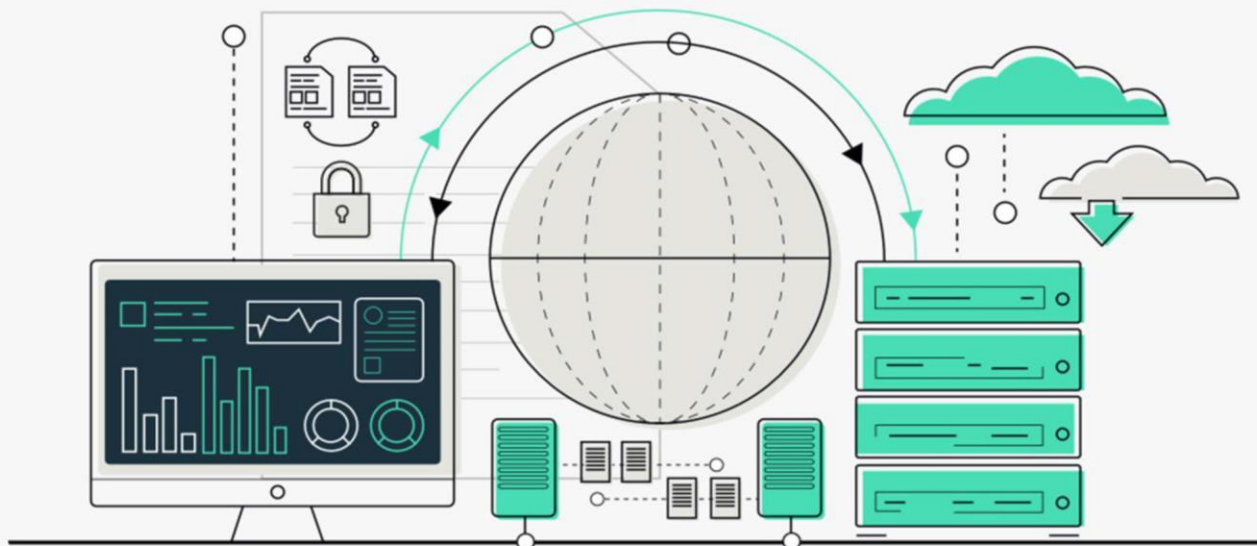
اوپن استک (به انگلیسی: OpenStack) یک پلتفرم نرم افزاری متن باز برای رایانش ابری است که توسط Rackspace و ناسا در ژوئیه ۲۰۱۰ طرح ریزی شد. هم اکنون بیش از ۵۰۰ شرکت از جمله IBM، فوجیتسو، اوراکل، یاهو، سیتریکس، دل، AMD، اینتل، کنونیکال، سوزه، اچ پی و سیسکو سیستمز و شرکت های بسیار دیگری به این پروژه پیوسته اند.

### OpenStack یک پروژه رایانش ابری است که به تجهیز بستری

متن باز و قابل دسترس در همه جا برای ابرهای شخصی و عمومی کمک می کند. این پروژه توسط OpenStack Foundation مدیریت می شود که یک بنیاد غیرانتفاعی است که در سپتامبر ۲۰۱۲ تأسیس شده است.

این استک به کاربران اجازه می دهد تا بتوانند ماشین های مجازی و دیگر نمونه ها برای کنترل وظایف گوناگون در مدیریت یک محیط ابری در حال اجرا را توسعه دهند. این مجموعه ابزار گسترش عمودی را آسان می کند به این معنی که وظایفی که از اجرای همزمان سود می برند، می توانند در زمان اجرا، به آسانی با تنظیم نمونه های متفاوت، به کاربران بیشتر یا کمتری خدمات ارائه دهند. برای مثال، یک اپلیکیشن موبایل، که باید با یک سرور از راه دور ارتباط برقرار کند، می تواند کار ارتباطی خود را با کاربران دیگر در طول نمونه های مختلف، تقسیم کند که همه اینها با سرعت و به آسانی با یکدیگر در ارتباط خواهند بود؛ و مهمتر از آن، این استک یک ابزار متن باز است، به این معنی که هر فردی که آن را انتخاب کند، می تواند به کد منبع آن نیز دسترسی داشته، هر تغییر یا اصلاحی را که مورد نیاز است، اعمال کند و به صورت رایگان، تغییرات اعمال شده را در سطح وسیعتری به اشتراک بگذارد. متن باز بودن آن نیز این امکان را هم می دهد که هزاران توسعه دهنده در سرتاسر جهان می توانند بر روی آن کار کنند و مجموعه نرم افزاری با ویژگی های قوی تر و امن تر تولید کنند.

### نحوه کار با این استک چیست ؟



### :Ceilometer

Ceilometer، سرویس تله متری را برای اپن استک فراهم می‌کند که در واقع فراهم کننده سرویس‌های صورتحساب برای کاربران فردی محاسبات ابری است. این سرویس نیز، میزان قابل تأییدی از استفاده کاربر از هر کدام از مولفه‌های یک اپن استک را نگهداری می‌کند.

### :Heat

Heat در واقع مولفه هماهنگ سازی و تنظیمات در اپن استک است که به تو سعه دهندگان این امکان را می‌دهد که نیازهای اپلیکیشن‌های cloud را که منابع لازم برای هر برنامه را تعریف می‌کند، در یک فایل ذخیره کنند. با این روش، می‌توان زیرساخت موردنیاز را که سرویس cloud باید بر روی آن اجرا شود، مدیریت کرد.

### :Mistral

Mistral خدمتی است که گردش‌های کاری را مدیریت می‌کند. به طور معمول کاربر با استفاده از زبان‌هایی مانند YAML که بر پایه‌ی گردش کاری کار می‌کنند، یک گردش کاری را می‌نویسد و تعریف گردش کاری مذکور را با استفاده از رابط برنامه‌نویسی کاربردی بر پایه REST برای Mistral ارسال می‌کند. سپس کاربر می‌تواند به صورت دستی یا با استفاده از پیکربندی یک یا چند رویداد، گردش کاری را فعال نماید.

### :Trove

Trove خدمت Database-as-a-Service را به عنوان موتور پایگاه داده رابطه‌ای (بر پایه SQL و همچنین غیر رابطه‌ای (NoSQL) را برای اپن استک فراهم می‌کند.

روش سنتی برای دسترسی به فایل‌ها معمولاً به خاطر اهمیت سرعت دسترسی به داده، ممکن است مورد بحث باشد.

### :Neutron

Neutron قابلیت شبکه سازی برای اپن استک را فراهم می‌کند. این ویژگی به هر مولفه یک پلتفرم اپن استک کمک می‌کند تا بتواند با دیگری به صورت کارا و سریع ارتباط برقرار کند.

### :Horizon

یک پنل کنترلی برای اپن استک است. این پنل تنها واسطه گرافیکی اپن استک است که در واقع می‌تواند اولین مولفه برای کاربرانی باشد که می‌خواهند اپن استک را تست کنند. توسعه دهندگان می‌توانند به تمام مولفه‌های اپن استک از طریق API دسترسی پیدا کنند ولی این پنل کنترلی شرایطی برای مدیران سیستم فراهم می‌کند تا نگاهی اجمالی به کل سیستم cloud داشته باشند و در صورت نیاز آن را مدیریت کنند.

### :Keystone

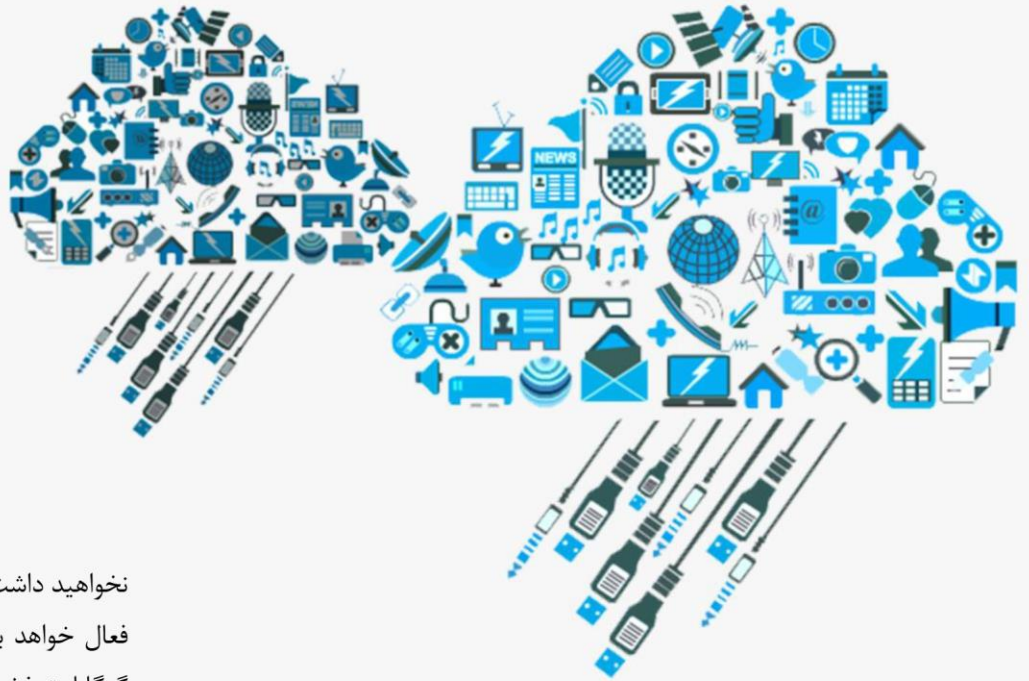
این ابزار در واقع سرویس‌های هویت را برای اپن استک فراهم می‌کند. در اینجا لیستی از کاربران اپن استک به همراه دسترسی‌های مجاز آنها نگهداری می‌شود. این سرویس، ابزارهای دسترسی چندگانه فراهم می‌کند به این معنی که توسعه دهندگان می‌توانند به آسانی دسترسی کاربر موجود را با استفاده از روش‌هایی در keystone نگاشت کنند.

### :Glance

وظیفه Glance، ارائه سرویس image در اپن استک است. در این مورد، image ها به image های موجود در هارد دیسک ارجاع داده می‌شوند. این image ها می‌توانند در زمان توسعه یک نمونه ماشین مجازی جدید، به عنوان الگو استفاده شوند.



# اپلیکیشن های شبه کلود



نخواهید داشت چرا که این سیستم بر روی حساب گوگل شما فعال خواهد بود. در سرویس رایگان آن شما می‌توانید تا ۵ گیگابایت ذخیره سازی را انجام بدهید و در سرویس پولی این میزان تا ۱۰۰ گیگابایت افزایش خواهد یافت. سرعت آپلود و راحتی کار با آن از دیگر مزایای این سیستم است.

## • وان درایو

این سرویس هم برای افرادی مناسب است که خواهان سرعت بالا در آپلود هستند و محیط آن مانند اینترنت اکسپلور است. تا ۲۵ گیگابایت قابلیت ذخیره سازی دارد و در آن تا ۵۰ مگابایت به طور همزمان می‌توان فایل ذخیره کرد. به این سیستم از طریق وبسایت می‌توان دسترسی پیدا کرد و اپلیکیشن ویندوزفون، آیفون و اندروید نیز برای آن منتشر شده است. همچنین قابلیت ویرایش فایل‌های آفیس به صورت مستقیم در آن برقرار است.

## • باکس

در باکس متناسب با نوع ثبت نام بین ۵ تا ۵۰ گیگابایت فایل به صورت رایگان می‌توان ذخیره کرد و قابلیت مهم آن راحتی درگ و دراپ کردن فایل‌ها از روی دسکتاپ برای آپلود است. اپلیکیشن‌های اندروید، آیفون، آپید و بلک بری برای باکس وجود دارند تا در هر کجا که هستید بتوانید به فایل‌هایتان دسترسی داشته باشید. قابلیت ویرایش مستقیم فایل‌های ورد و اکسل نیز به کمک افزونه‌هایی در آن وجود دارد.

## • ای درایو

ADrive از معدود سرویس‌های ابری است که در آن می‌توانید تا ۵۰ گیگابایت ذخیره سازی کنید. شما می‌توانید به راحتی با درگ و دراپ کردن فایل‌ها را آپلود کنید و با

حتی فکر کردن به اینکه شما در یک اتفاق تمام اطلاعات خود را از دست بدهید نیز ناگوار خواهد بود، به خصوص اگر شما از این اطلاعات پشتیبانی نیز تهیه نکرده باشید، در این جا اهمیت تکنولوژی ذخیره سازی ابری معلوم می‌شود.

به یاد می‌آورم که یک بار در یوتیوب ویدیویی را دیدم که در آن یک مجری از شهروندان آمریکایی سوال می‌کرد که آیا وضعیت آب و هوا بر روی اطلاعات آنان در سرویس‌های ابری تاثیر می‌گذارد؟ پاسخ بیشتر آمریکایی‌ها (که به پایین بودن اطلاعات عمومی‌شان معروفند) مثبت بود! اما فضای ابری در واقع یک سرویس ارائه فضای ذخیره‌سازی به کاربران است تا بتوانند اطلاعات خود را بر بستر اینترنت ذخیره‌سازی کنند تا هم خیالشان از بابت خراب شدن حافظه و از بین رفتن اطلاعاتشان راحت شود و هم این که بتوانند از هر کجا و به وسیله گجت‌های مختلف به اطلاعات خود دسترسی داشته باشند.

در این تکنولوژی شما صد در صد از سلامتی داده‌های خود اطمینان خواهید داشت و داده‌های موبایل که قبلاً جزو ناامن‌ترین‌ها به شمار می‌رفتند، اکنون با تمرکز ذخیره‌سازی ابری بر روی آنها می‌توانند از ایمنی و پشتیبانی مطمئنی برخوردار باشند. در این مقاله ۵ تا از بهترین سرویس‌های ذخیره‌سازی ابری را معرفی خواهیم کرد.

## • گوگل درایو

این سیستم را می‌توان از بهترین نمونه‌های فضای ابری دانست که توسط گوگل پشتیبانی می‌شود. می‌توان در آن به طور همزمان به ذخیره سازی و اشتراک گذاری اطلاعات و فایل‌ها پرداخت. شما نیاز به ثبت‌نام یا ساختن اکانت‌های متعدد

نرم افزار مخصوص خود سایت، ویرایش را بر روی فایل ها انجام بدهید. این سیستم قابلیت ذخیره سازی و پشتیبانی تمامی فایل ها از جمله موسیقی، ویدیو، عکس و... را دارد.

## • آی کلود

این سرویس برای دستگاه های با سیستم عامل iOS موجود است و به شما قابلیت ذخیره سازی تا ۵ گیگابایت را می دهد و اگر مورد نیاز شما نبود باید برای افزایش آن مبلغی را بپردازید. تمام کسانی که فایلی را از فروشگاه iTunes خریداری کرده و از طریق این سیستم اشتراک گذاری می کنند، از حجم ۵ گیگابایتی شان کم نمی شود. این سیستم نیز مانند سرویس قبلی قابلیت ذخیره سازی همه نوع فایل را به شما می دهد.

## • دراپ باکس

دراپ باکس یکی از قدیمی ترین و معروف ترین سرویس های فضای ابری است. استیو جابز پیش از راه اندازی iCloud سعی کرد دراپ باکس را خریداری کند که البته توافقی بین این دو حاصل نشد. رابط کاربری بسیار ساده، محیطی، تر و تمیز و کاربر پسند. امکان ذخیره سازی همه نوع فایل و به اشتراک گذاری آنها، قابلیت همگام سازی فایل ها از طریق یک فولدر اختصاصی در سیستم عامل دسکتاپ شما و پشتیبانی از همه پلتفرم های جدید موبایل آن را به یکی از برترین سرویس های ارایه فضای ابری بدل می سازد. شما در ابتدا ۳ گیگابایت فضای ذخیره سازی رایگان در اختیار خواهید داشت که البته با دعوت از دوستانتان می توانید این فضا را به ۱۶ گیگابایت هم برسانید. مزیت این سرویس این است که بر

خلاف وان درایو، گوگل درایو و آی کلاد شما را به یک اکوسیستم خاص سوق نمی دهد و مستقل از این که بیشتر از سرویس های کدام کمپانی استفاده می کنید می توانید از دراپ باکس استفاده کنید.

## Cloud چگونه وارد دنیای تکنولوژی شد؟

نقش مهندس شبکه طراحی یک شبکه است که به درستی کار کند. برای این کار، زمان شخص، مختص به این خواهد بود که بررسی کند چه دستگاه های (Device هایی) در شبکه هستند و چطور به یکدیگر متصل شده اند تا آنها را مدیریت و کنترل نماید. اما شبکه ها به شبکه های دیگر یا اینترنت هم متصل بودند. هنگامی که مهندسين شبکه، شبکه خود را بطور شماتیک طراحی می کردند برای نشان دادن این اتصال به عنوان بخشی از طراحی، سعی نکردند این شبکه ها را توصیف کنند! به این دلیل که بررسی های زیادی نسبت به این شبکه ناشناخته باید انجام می شد که این کار هم پیچیده بود هم زمان زیادی می برد و اصلا نیازی به این کار نبود! پس، این شبکه ها را با منحنی های بسته نشان دادند که رفته رفته تبدیل به شکل ابر یا همان Cloud شد و بیانگر چیزی بود که جزئیات آن ناشناخته است! به این صورت شکل ابر یا همان Cloud به دنیای تکنولوژی راه پیدا کرد!

درواقع Cloud استعاره ای از اینترنت شد و برای معرفی Cloud computing از عبارت "استفاده از داده ها از راه دور" استفاده گردید. اما اکنون اگر بخواهیم توضیح کامل و جامعی درباره این سوال که "Cloud computing" چیست؟ بدهیم باید گفت Cloud یک بستر برای سرویس دهی است! اما این که تنها بگوییم Cloud بستر سرویس دهی است یک دسته بندی کلی خواهد بود. پس، بخاطر وجود سرویس های مختلف، سرویس های Cloud به سه دسته بندی مهم تفکیک داده شده اند تا بتوان آنها را راحت تر بررسی نمود.



# میکروسرویس ها



## معماری Monolithic

این نوع معماری تحت عنوان معماری MVC شناخته می‌شود.

- Model
- View
- Controller

برای روشن‌تر شدن این مسئله، مثالی می‌زنیم. فرض کنیم کاربری از طریق مرورگر خود درخواستی به برنامه بفرستد. این درخواست به دست سرور می‌رسد و لایه‌ی مرتبط با Controller این درخواست را گرفته و برای لایه‌ی Model ارسال می‌کند و این لایه هم با پایگاه داده ارتباط برقرار ساخته و داده‌های مرتبط را فراخوانی کرده و در صورت نیاز یک سری کارها بر روی داده‌ها انجام می‌دهد و در نهایت نتیجه را در اختیار Controller قرار می‌دهد. Controller پاسخ را در اختیار View قرار می‌دهد و مسئول نمایش پاسخ به کاربر (به وسیله‌ی مرورگر) است.

## مشکلات معماری Monolithic

در معماری Monolithic یک هسته‌ی مرکزی داریم که کاربران با روش‌های مختلف می‌توانند با آن ارتباط برقرار سازند. اگرچه می‌توان کل سیستم را ماژول بندی کرد، اما همگی تحت یک پلتفرم واحد کنار یکدیگر قرار گرفته‌اند و امکان مجزاسازی این ماژول‌ها سخت و تا حدودی ناممکن است. در مورد استفاده از این معماری، باید مشکلات زیر را در نظر گرفت:

امروزه واژه **Microservice** بسیار مورد استفاده قرار می‌گیرد. مقالات و آموزش‌های زیادی پیرامون این موضوع تدوین شده و شرکت‌های بزرگی مانند Amazon بصورت بسیار گسترده از آن استفاده می‌کنند. اما ممکن است این مفهوم برای کسانی که تازه به دنیای برنامه نویسی وارد می‌شوند، کمی گنگ باشد. در این مقاله با این معماری و نقاط مثبت و منفی آن بیشتر آشنا می‌شویم.

## Micro services

معماری نرم‌افزار یک راه‌حل علمی برای پیاده‌سازی یک سیستم نرم‌افزاری می‌باشد. به عبارت دیگر معماری نرم‌افزار مجموعه‌ای از تصمیمات است بر پایه‌ی عوامل موجود. این تصمیمات می‌تواند بر سرعت، امنیت و کنترل‌پذیری کل پروژه و در نهایت موفقیت آن تأثیر داشته باشد.

یک سیستم خرید آنلاین را در نظر بگیرید. این سیستم باید بتواند حجم وسیعی از درخواست‌های کلاینت‌های مختلف را در مدت زمان مناسب، پاسخ بدهد. یعنی افزایش تعداد کاربران فعال و یا حتی تعداد درخواست‌های آنان در روزهای خاص، نباید بر عملکرد سیستم خدشه وارد کرده و یا آن را از دسترس خارج کند. علاوه بر آن این برنامه‌ها باید بتوانند به سادگی رشد کرده و به‌روز شوند.

## اکنون برای چنین سیستمی از چه معماری باید استفاده کنیم؟

برای ساخت و توسعه این نوع برنامه‌ها، دو مدل معماری رایج وجود دارد:

- (۱) معماری Monolithic
- (۲) معماری Micro services

باشند. این روش پیاده سازی قابلیت **Scaling** و **Testability** را بالا می برد و توسعه و نگهداری سیستم را آسان می کند.

### مزایای استفاده از **Micro Services**

به طور کلی، ایده ی میکرو سرویس ها این است که این امکان به برنامه نویسان داده شود تا اپلیکیشن های خود را از اجزا یا سرویس هایی که مستقل از یکدیگر هستند و به سادگی قابل تغییر، حذف و به روزرسانی می باشند. از مزایای دیگر این معماری می توان موارد زیر را نام برد:

- قابلیت **Fault Isolation** در سیستم افزایش می یابد.
- از آنجایی که سرویس ها از طریق زبان مشترک شبکه با یکدیگر در ارتباط هستند، می توان آنها را با زبان های برنامه نویسی مختلف و با فریم فرک های متفاوت نوشت.
- می توان هر سرویس را به صورت جداگانه ایجاد کرد و تغییر داد که باعث سرعت در به روزرسانی و فرآیند گسترش برنامه می شود.
- از آنجایی که این سرویس ها از طریق شبکه در تبادل هستند، می شود از آنها در سایر برنامه ها مجدداً استفاده کرد.
- ارتقاء و به روزرسانی هر یک از سرویس ها مستقل از کل سیستم بوده و به نسبت معماری **monolithic** زمان کمتری را صرف می نماید.
- درک عملکرد یک سرویس برای یک توسعه دهنده سیستم ساده تر از درک کل سیستم خواهد بود.
- مدیریت استفاده از منابع سخت افزاری در معماری میکروسرویس، بهینه تر است.

### محدودیت های **Micro Services**

- به دلیل ارتباط سرویس ها در بستر شبکه، انتظار کندی عملکرد آن ها دور از ذهن نیست.
- به دلیل ارتباطات شبکه ای، احتمال آسیب پذیری امنیتی بالا می رود.
- به دلیل مجزا بودن بخش های مختلف برنامه، مدیریت و ردیابی عملکرد سرویس ها، یکی از کارهای اصلی به حساب می آید.
- بدلیل تعدد سرویس ها و بخش های مختلف، توسعه و تنظیمات آن سخت و زمان بر است.

در آخر باید اضافه کرد انتخاب نوع معماری بر عهده ی توسعه دهنده است. شما به عنوان یک برنامه نویس باید سیستم خود را به طور کلی در نظر بگیرید و با برآوردی هزینه و زمان و همچنین در نظر گرفتن ویژگی های برنامه تان، معماری مناسب را انتخاب کنید.

- با هر گونه تغییر در برنامه های مبتنی بر این معماری، نیاز به **Build** و **Deploy** مجدد کل برنامه می باشد که باعث از دسترس خارج شدن موقت برنامه خواهد شد.
- وجود باگ در یکی از ماژول ها، کل پروژه را تحت الشعاع خود قرار خواهد داد.
- امکان اجرای بخش هایی از سیستم بر روی کامپیوترهای دیگر نیست و اگر بخشی از سیستم از کار بیوفتد احتمال توقف عملکرد کل سیستم وجود دارد.
- کلیه اجزای سیستم بر اساس یک زبان برنامه نویسی مشخص داشته و برای یک فریم ورک مشخص نوشته می شوند. این برنامه ها اصطلاحاً چند سکویی نیستند و کامپوننت های نوشته شده برای آن ها فقط در فریم ورک جاری قابل استفاده مجدد هستند.
- تنوع تکنولوژی ( شامل زبان های برنامه نویسی، دیتابیس های مختلف، لایبرری ها و فریم ورک های مختلف) وجود نخواهد داشت و اگر هم این گونه باشد، ارتباط برقرار کردن میان آنها دشوار خواهد بود.
- کامپوننت ها را نمی توان به سادگی با یک معماری جدیدتر و بهینه تر جایگزین کرد؛ چون کل معماری را تحت الشعاع قرار می دهد.

البته باید در نظر داشت که روند توسعه ی اپلیکیشن های مبتنی بر این معماری، بسیار سریع تر از سایرین بوده و برای سیستم های کوچک، راه حلی مناسب و کارآمد می باشد.

### معماری **Micro Services**

میکروسرویس روشی به منظور تقسیم بندی کردن یک برنامه به بخش ها یا سرویس های کوچک، سبک، مستقل است. به عبارت دیگر، میکروسرویس یک معماری توسعه ی نرم افزار **Distributed** (پخش شده) است. این نوع سرویس ها صرفاً به منظور انجام دادن یک تسک خاص طراحی می شوند؛ به طور مثال، یک سرویس صرفاً وظیفه ی مدیریت کاربران را بر عهده دارد و سرویس دیگر برای کار با پایگاه داده استفاده می شود.

سرویس های مختلف یک اپلیکیشن با معماری میکروسروسی با استفاده از درخواست هایی از جنس **HTTP** و **API** های به اصطلاح **RESTful** این ارتباط برقرار خواهد شد.

هر یک از این سرویس ها می توانند توسط تیم های جداگانه ای با پلتفرم توسعه و زبان برنامه نویسی و بانک اطلاعاتی جداگانه ای توسعه داده شوند و با یک مکانیزم سبک وزن مانند **Http** با یکدیگر در ارتباط



۶) فرم‌های سنگین و کند

چند روش برای طراحی یک تجربه کاربری خوب:

- ۱) تحقیق و پژوهش (Research)
- ۲) ساخت شخصیت کاربر (User Persona)
- ۳) ساخت مسیر کاربر
- ۴) ساخت وایرفریم و پروتوتایپ
- ۵) طراحی رابط کاربری و تحویل به توسعه دهنده
- ۶) تحلیل و بررسی - آزمون و خطا

چگونه رفتار کاربران را برای ایجاد یک تجربه کاربری موفق درک

کنیم؟



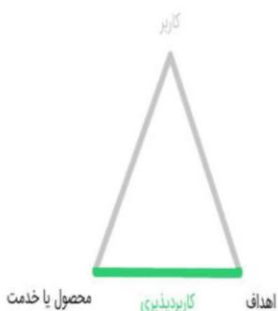
۱) تجزیه و تحلیل



۲) پشتیبانی مشتری



۳) آزمایش کاربرپذیری



مینیمالیسم (به معنای به کم قانع بودن) مفهومی است که در رشته‌های زیادی از جمله هنر، موسیقی، ادبیات، طراحی و فلسفه دیده می‌شود. این نوع طراحی، شیک بودن را در طراحی سادگی خود دارد و چیزی که مهمتر است را در طراحی برجسته می‌کند. برخی از موضوعات اصلی در طراحی مینیمالیستی عبارتند از:

- **طراحی تخت یا Flat Design:** سبکی خاص از طراحی است که در آن تمامی جزئیات گرافیکی نظیر دکمه‌ها و آیکون‌ها، از نوع دو بعدی کار شده‌اند.
  - **طراحی تک‌رنگ (Monochrome) یا با جعبه رنگی محدود:** این طراحی کمک می‌کند که تمرکز کاربر را بر روی محتوای اصلی نگه‌داریم.
  - **طراحی تایپوگرافی (Typography):** در این نوع طراحی، رنگ‌ها و فونت‌ها معمولاً به صورت «Bold» و کاملاً مشخص هستند.
  - **محدودیت انتخاب (Choice Limitation):** در این نوع طراحی، با حذف عناصر و توابع غیر ضروری، توجه کاربر را به موضوع اصلی جلب می‌کنیم.
  - **فضای منفی (Negative space) یا فضای سفید (whitespace):** این طراحی یکی از اصلی‌ترین طراحی‌های مینیمالیستی است. تضادی در این طراحی‌ها برقرار است که باعث می‌شود مطالب مهم سایت بیشتر در دید باشند.
- ۳) طراحی به روش فنی:
- یک طراح به هر روش یا هر سبکی که کار کند، هدف نهایی او این است که یک رابط کاربری ارائه بدهد تا مردم بتوانند با رایانه‌ها ارتباط برقرار کنند. کری ودهاوس (Carey Wodehouse) در این مورد می‌گوید: قبل از لپ‌تاپ‌ها، تلفن‌های هوشمند و اپلیکیشن‌های همراه؛ قبل از رابط‌های کاربری گرافیکی، مرورگرها و موتورهای جست‌وجو؛ قبل از رابط کاربری و تجربه کاربری، چیزی داشتیم به نام «تعامل انسان با رایانه»، دانشی برای ارتباط بهتر و بصری بین انسان و رایانه.

اشتباهات رایج در طراحی رابط کاربری:

- ۱) کنتراست پایین در طراحی
- ۲) واکنش‌گرا نبودن طراحی
- ۳) کپی برداری
- ۴) معماری اطلاعات (Information Architect) نامناسب
- ۵) ناهماهنگی در سبک طراحی



جاوا ...

جاوا چی چی؟

یه زبان برنامه نویسیه؟

شاید یه شاخه از زبان جاواست!!!

جاوا اسکریپت به شما امکان ساخت سایت‌های تعاملی را می‌دهد؛ پس کاملاً با جاوا متفاوت است. جاوا اسکریپت در کنار HTML و CSS به یک تکنولوژی ضروری وب تبدیل شده است و اکثر مرورگرها جاوا اسکریپت را اجرا می‌کنند؛ بنابراین، اگر می‌خواهید یک توسعه‌دهنده‌ی وب باشید نیاز است تا جاوا اسکریپت را بیاموزید. همچنین اگر می‌خواهید یک توسعه‌دهنده‌ی frontend باشید و یا از جاوا اسکریپت در توسعه‌های backend استفاده کنید باید بر جاوا اسکریپت تسلط داشته باشید. علاوه بر این موارد؛ امروزه جاوا اسکریپت به حدی گسترش یافته که در توسعه‌ی برنامه‌های موبایل، ویندوز و حتی بازی‌ها از آن استفاده می‌شود. در کل یک زبان بسیار محبوب و یادگیری آن یک مهارت بسیار کاربردی است.

سلام دوستان خوبم امروز می‌خواهم javascript رو به شما معرفی کنم.

جاوا اسکریپت چیست؟؟؟

جاوا اسکریپت زبانی برای نوشتن برنامه‌های ساده‌ایست که در مرورگر کاربر اجرا می‌شوند و ظاهر و امکانات وبسایت‌ها را متحول می‌کنند. جاوا اسکریپت تقریباً در هر وبسایتی کاربرد دارد و در عین حال کمتر نامش را شنیده‌ایم! سال‌ها پیش در دورانی که صفحات وب به صورت HTML و بسیار ساده طراحی می‌شدند، متن و عکس در کنار هم قرار می‌گرفت و حالت پویا نداشت، در واقع همه چیز حالت ایستا داشت. اما این روزها، این موضوع برعکس شده و کمتر سایتی را سراغ داریم که به طور کامل ایستا باشد، ولی وقتی صفحه‌ی شبکه‌های اجتماعی مثل فیس‌بوک و توییتر را باز می‌کنید، جاوا اسکریپت روی مرورگر شما اجرا می‌شود. به همین علت است که وقتی با استفاده از موس سراغ بخش‌های مختلف صفحه می‌روید، برخی عناصر عوض می‌شوند.



تاریخچه جاوا اسکریپت

در سال ۱۹۹۵ دو مرورگر اصلی در دنیای وب وجود داشت. مرورگر اکسپلورر که تازه کارش را شروع کرده بود و مرورگر دیگری به نام Netscape کمپانی نت‌اسکیپ علاوه بر معرفی یک مرورگر کامل و محبوب، به دنبال ایجاد یک زبان برنامه‌نویسی هم بود؛ که ساده باشد و برای توسعه‌دهندگان و حرفه‌ای‌ها، مفید واقع شود و در ادامه دنیای وب و وبسایت را متحول کند.

جاوا اسکریپت زبانی بود که پس از ۱۰ سال کار Brendan Eich متولد شد، کسی که بعدها برای مدتی کوتاه به عنوان رهبر موزیلا فعالیت کرد. زبانی که او ابداع کرده بود برای انواع برنامه‌نویسی شیء‌گرایی آبجکتیو و تابعی کاربرد داشت و از طرفی به زبان‌های متداول روز مثل C و C++ شبیه است.



جاوا اسکریپت شاخه‌ای از جاواست؟

خیر؛ ممکن است فکر کنید جاوا که در دوران گوشی‌های غیرهوشمند در اکثر گوشی‌ها متداول بود، همان جاوا اسکریپت است؛ اما این تصور کاملاً اشتباه است. جاوا زبانی است که توسط کمپانی Sun Microsystems توسعه داده شده است.

با توجه به این که سرویس‌های آنلاین در حال افزایش است؛ واضح است که جاوا اسکریپت به رشد خود ادامه خواهد داد و در سال‌های آینده هم همچنان باقی خواهد ماند.

در کل پیش‌بینی این که آیا مسیر جاوا اسکریپت به عنوان یک تکنولوژی همیشه در حال گسترش باشد یا نه دشوار است، اما نیاز به گفتن نیست که جاوا اسکریپت تا زمانی که تقاضا برای آن وجود داشته باشد پیشرفت خواهد کرد.

### و در آخر فرصت شغلی در جاوا اسکریپت

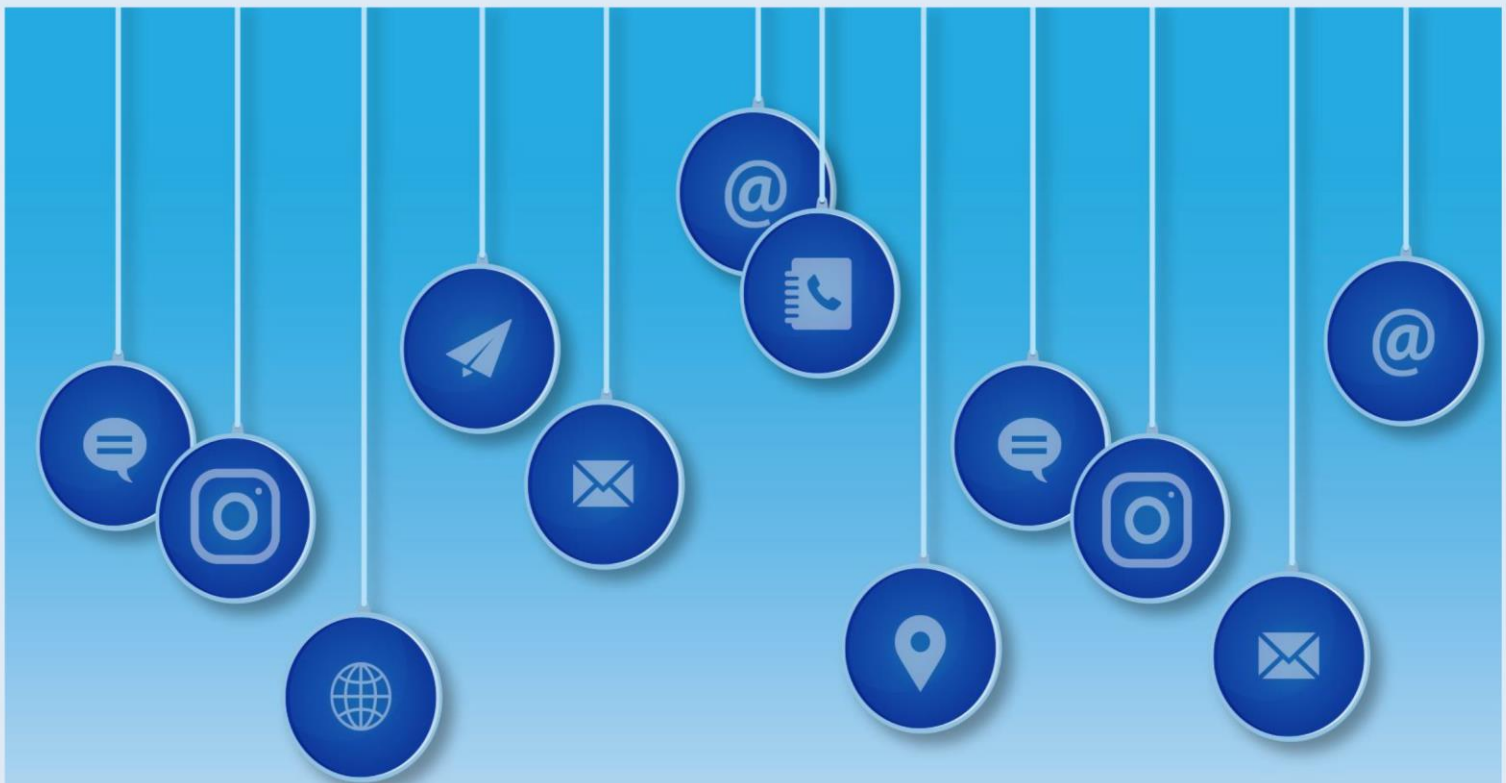
با توسعه‌ی روزافزون کسب و کارهای آنلاین، تقاضا برای توسعه‌دهندگان جاوا اسکریپت نیز در حال افزایش است. همچنین جاوا اسکریپت پرتقاضاترین مهارت در Angel List است. اگر می‌خواهید از جاوا اسکریپت برای توسعه‌های backend استفاده کنید، متوسط حقوق سالانه در ایالات متحده برای توسعه‌دهندگان Node.js حدود ۹۸،۹۶۲ دلار است.

مرورگر وب یکی از پرستفاده‌ترین برنامه‌ها هم روی دسکتاپ و هم روی دستگاه‌های موبایل است. با افزایش سرویس‌دهی‌ها در وب، به محبوبیت جاوا اسکریپت افزوده و به‌طور کلی باعث ابداعات تکنیکی زیادی در جاوا اسکریپت می‌شود. بسیاری از توسعه‌دهندگان و تاجران از جاوا اسکریپت برای گسترش محصولات وب خود در خارج از مرورگر استفاده کرده‌اند. این به این معناست که از جاوا اسکریپت می‌توان در تولید نرم‌افزارهای دسکتاپ و موبایل استفاده کرد.

جاوا اسکریپت هشتمین زبان پرطرفدار دنیاست در شاخص TIOBE.

توانایی یک زبان برای باقی ماندن و زنده ماندن بستگی به این دارد که آیا به زبان ایده‌های نو و خلاقانه وارد می‌شود یا نه.

علاقه به جاوا اسکریپت در سال ۲۰۱۵، ۳، ۲۲٪ افزایش یافت.



لطفا نظرات، پیشنهادات و انتقادات خود را  
از طریق راه‌های ارتباطی با ما در میان بگذارید

 @pardazesh\_magazine

 pardazesh\_magazine

 process.magazine1391@gmail.com





# مصاحبه با

## موفق



عنوان به محقق کامپیوتری داشتم فرایندی به اسم آزمایش داوینچی رو روی یک ربات جراحی تست می‌کردم و با حرکت دست‌هام و همین‌طور به کمک الگوریتم‌های افزایش سرعت و واقعیت مجازی، عملیات جراحی رو کنترل می‌کردم. جالب‌تر این‌که مسئول پروژه بهم گفت: (( من فکر می‌کنم که اگر تو توی فیلد پزشکی هم می‌رفتی، قطعاً یک جراح خبره می‌شدی!)) حرفش خیلی بهم چسبید! اعتقادم اینه که هرکس در هر رشته‌ای و هر دانشگاهی اگه بر اساس علاقه‌مندی رشته‌اش را انتخاب کنه، حتماً توش موفق می‌شه و می‌تونه به بهبود کار و شرایط گروه‌های دیگه‌ی شغلی هم کمک کنه.

من چهار سال از عمرم را صرف تحصیل در رشته‌ی مهندسی توی دانشگاه الزهرا (س) کردم و تعلق خاطر خاصی نسبت بهش دارم. دانشگاه الزهرا (س)، دانشگاه متفاوتیه و این تفاوت توی زندگی من تاثیر مثبتی داشته. از اساتید بزرگوارم خیلی چیزها آموختم و قطعاً بخش اعظمی از موفقیت‌هام رو مرهون راهنمایی‌های اون بزرگواران می‌دونم.

به علاوه دانشجویان این دانشگاه از سطح دانش بسیار بالایی برخوردارن و در زمان تحصیل من اکثر دانشجویها با رتبه‌های بسیار خوبی وارد این دانشگاه می‌شدن به نحوی که امکان پذیرش برای رشته‌هایی مثل علوم کامپیوتر و یا ریاضی در دانشگاه‌های شریف و یا تهران رو داشتن و اکثراً بنا به معیارهای مطلوب‌شون مهندسی رو انتخاب کردن و به این دانشگاه اومدن.

### از دوران تحصیل تون در دانشگاه

#### برامون بگید.

یکی از ویژگی‌هام که خیلی دوستش دارم اینه که همیشه سر همه‌ی کلاس‌ها حاضر می‌شدم و مطالب رو همون‌جا توی کلاس درس یاد می‌گرفتم، در نتیجه عمده‌ی وقت درسی خارج از کلاس، فقط صرف انجام تکالیف می‌شد. همین هم باعث می‌شد با حداقل زمانی که برای درس خوندن در کنار بقیه‌ی فعالیت‌های علمی، فرهنگی و تفریحی‌ام

کنجاوی زمانم رو صرف یاد گرفتن کوچک‌ترین جزئیات و یا پیچیده‌ترین ترکیب‌ها می‌کنم، بلکه استعداد تحلیل و یادگیری مفاهیم پایه‌ای این محث و همین‌طور توانایی پیاده‌سازی ایده‌هام به کمک این مفاهیم رو هم دارم. با شروع دوره‌ی دبیرستان در مدرسه‌ی استعدادهای درخشان فرزندانگن تهران، بیشتر درگیر برنامه‌نویسی شدم و بعد هم با شرکت توی مسابقات رباتیک فوتبالیست دوبعدی و سه‌بعدی، انگیزه‌ام برای ادامه دادن این شاخه دوچندان شد.

با توجه به علاقه‌مندی خاصی که به رشته کامپیوتر داشتم، تحصیلات دانشگاهیم را در این رشته شروع کردم. در سال ۱۳۸۹ در دانشگاه الزهرا (س) در رشته مهندسی کامپیوتر پذیرفته شدم و در سال ۱۳۹۳ فارغ‌التحصیل شدم و بعد برای ادامه تحصیل به آمریکا رفتم و دوره دوساله‌ی کارشناسی‌ارشد رو در رشته‌ی علوم کامپیوتر شروع کردم. بعد از فارغ‌التحصیلی از این مقطع، در سال ۱۳۹۶ تحصیلاتم رو در مقطع دکترای علوم و مهندسی کامپیوتر در دانشگاه University of California, San Diego شروع کردم و در حال حاضر مشغول به مطالعه و ساخت ربات‌های اجتماعی برای محیط‌های پزشکی در یک گروه تحقیقاتی هستم. ساختن نسل جدید ربات شبه‌بیمار که قابلیت انسان‌نمایی، ابراز احساسات، درد و علائم بیماری رو در ظاهرشون داشته باشن پروژه‌ایه که طی یک سال اخیر دارم روش کار می‌کنم.

### از دانشگاه الزهرا برامون بگید. اینکه

چجوری اومدید به این دانشگاه، چی شد که رشته‌ی کامپیوتر رو انتخاب کردید؟ درباره رشته و دانشگاه تون چه حسی داشتید؟

همون‌طور که قبلاً هم گفتم، از اول علاقه خاصی به رشته کامپیوتر داشتم تا جایی که حاضر نبودم با رشته‌هایی مثل پزشکی عوض کنم! البته الان خیلی علوم به هم مرتبط شدن و می‌شه لذت چندتاشون رو با هم برد! مثلاً امروز به

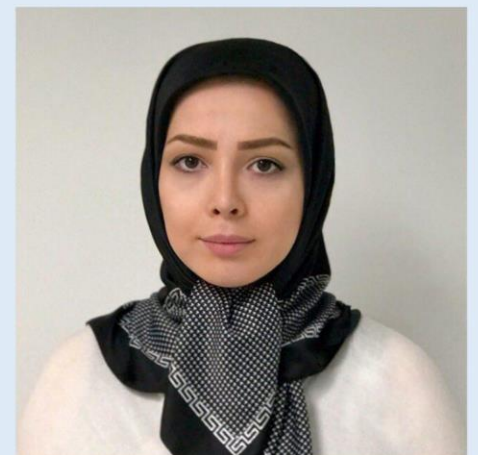
مصاحبه‌ی این شمارمون با سرکار خانم مریم پورعبادی، یکی از فارغ‌التحصیلان موفق دانشگاهمون هست، ایشون دوره‌ی کارشناسی رو در دانشگاه الزهرا گذروندن در حال حاضر محقق رشته‌ی کامپیوتر در دانشگاه کالیفرنیا، سن‌دیگو هستن. ایشون در طول تحصیلشون زمینه‌ی بسیاری از فعالیت‌های فرهنگی، علمی و آموزشی رو برای گروهمون فراهم کردن. صحبت‌های ایشون رو در این شماره می‌خونید.

### لطفاً به بیوگرافی کلی از خودتون برامون بگید.

مریم پورعبادی هستم. آذر ۱۳۷۰ در تهران متولد شدم. فرزند اول یک خانواده‌ی فرهنگی‌ام و یک برادر به اسم مهدی و یک خواهر به اسم معصومه دارم. میرزا قاسمی غذای مورد علاقه منه و تالش سرزمین محبوبمه.

یادم می‌آد نوروز سه سال پیش یکی از بستگانم بهم گفت «چهارم دبستان بودی که دیدم پای کامپیوتر نشستی و سعی می‌کنی از ساده‌ترین برنامه‌ی ویندوز، پیچیده‌ترین خروجی‌های ممکن رو بگیری. اونجا بود که مطمئن شدم تو حتماً به روز لیسانس مهندسی کامپیوتر می‌گیری!»

من اما فکر می‌کنم علاقه‌ام به رشته کامپیوتر به صورت جدی از سال دوم راهنمایی با شرکت توی المپیاد ریاضی و کامپیوتر و آوردن رتبه شروع شد و اونجا بود که فهمیدم نه فقط از روی



صرف می‌کردم، بازم بتونم در پایان دوره‌ی کارشناسی با معدل الف فارغ‌التحصیل بشم. این روند طی دوره‌ی کارشناسی‌ارشد هم ادامه پیدا کرد و به عنوان دانشجوی ممتاز با رتبه‌ی اول فارغ‌التحصیل شدم و به دوره‌ی دکترای راه پیدا کردم.

در کنار درس خواندن، درس دادن هم یکی از بزرگ‌ترین لذت‌های زندگی من و به خاطر همین تا به امروز، TA بیش از ۱۵ درس بودم. بخت با من یار بود که بعد از نشست سر چندین کلاس درس دکتر بهروز قلی‌زاده تونستم به سطحی از دانش برسم که از اون به بعد برای دو تا از درس‌های ایشان هم به عنوان حل تمرین انتخاب بشم و به دانشجویهاشون کمک کنم.

شما در دوران دانشجویی تون خیلی فعال بودید... چجوری خودتون و زمان تون رو مدیریت می‌کردید؟ فعال بودن در زمینه‌های غیر از درس در دانشگاه به نظرتون خوبه؟

زندگی ما آدم‌ها خواه‌ناخواه از ترکیب اجزای مختلف به هم تشکیل شده. مثل تحصیل، کسب درآمد، روابط خانوادگی، روابط اجتماعی، تفریح و خیلی چیزای دیگه. به خاطر وجود همین اجزای مختلف زندگی آدم‌ها (مستقل از این که کسی بخواد در کنار درس، فعالیت غیر درسی داشته باشه یا نه)، برای هر آدمی لازمه که مدیریت کردن شرایط، زمان و امکانات رو بلد باشه. مدیریت یک مهارته که بر اساس یادگیری و تکرار و تمرین به مرور زمان در افراد ایجاد و تقویت می‌شه.

یکی از بزرگ‌ترین نعمات زندگی من، پدر و مادرم هستن که هر دو از اساتید و مدیران موفق کشورند. به لطف داشتن‌شون و روش تربیتی‌شون، من در طی سال‌ها یاد گرفتم که باید چندین مهارت

اصلی رو یاد بگیرم و متوقف نشم تا زمانی که به چیزی که می‌خوام دست پیدا کنم. مهم‌ترین نکاتی که می‌تونم به همه‌ی شما هم توصیه کنم که تمرین‌شون کنین اینان:

### ۱) داشتن هدف و برنامه:

من یاد گرفتم اهداف نهایی‌ام رو از بین هدف‌هایی که برای من با ارزش انتخاب کنم تا در آخر کار احساس کنم برآیند هزینه‌هایی که در این راه کردم با دستاوردهایی که داشتم مثبت بوده و در مجموع چیزی به دارایی‌های من (اعم از سواد، شادی، سلامت، ثروت، اعتبار اجتماعی و یا احساس رضایت) اضافه شده.

برنامه‌ریزی‌های کوتاه‌مدت و بلندمدت و به‌روزرکردن‌شون بر اساس تغییرات در طی زمان به دو دلیل اصلی مهمن: اول به این خاطر که یک سری اقدامات پیش‌نیاز اقدامات دیگه هستن؛ و دیگه به خاطر این که تفاوت‌ها و یا تداخلی ممکنه بین برنامه‌ریزی‌های اشخاص مختلف درون گروه و یا سایر گروه‌ها وجود داشته باشه که باید قبل از شروع، یک مرحله شناسایی بشن. این برنامه‌ریزی کمک شایانی به ذخیره کردن زمان و استفاده‌ی بهینه از اون در آینده می‌کنه.

برای این کار لازمه مشخص کنیم چه مرحله‌ای برای رسیدن به یک هدف، اساسی هستن و در هر مرحله چه ویژگی شخصیت‌مون بیشترین کمک رو می‌تونه به پشت سر گذاشتن اون مرحله بکنه. پیش‌نیازش هم اینه که درباره‌ی خودمون، توانایی‌هامون و استعدادهامون به شناخت برسیم و ببینیم در چه قسمت‌های کار و زندگی می‌تونیم موثرترین باشیم.

### ۲) اولویت‌بندی کارها

می‌گن ایده‌های بد باعث از بین رفتن یک کسب و کار نمی‌شه، بلکه شروع کردن میلیون‌ها ایده‌ی خوب به صورت همزمان و تموم نکردن‌شون باعث شکست می‌شه.

کافیه کسی درباره‌ی موضوع شروع به حرف زدن با شما بکنه، بلافاصله حجم زیادی از فکرها و ایده‌ها به سمت مغز خوب‌تون هجوم می‌آرن! این درحالی‌که به نفر نمی‌تونه همه‌ی کارها رو همزمان به نحو احسن انجام بده و الزاماً همه‌ی ایده‌هاش شدنی یا درست نیستن.

برای همین من یاد گرفتم کارهام رو اولویت‌بندی کنم و زمان محدودم را صرف کارهایی که برام ارزش بیشتری دارن، بکنم. یه دفترچه دارم که همیشه همراهمه، توش برنامه‌هام رو می‌نویسم، بسته به اهمیت‌شون بهشون شماره می‌دم و بعضاً وقتی ددلاین هدف‌های اصلیم نزدیک می‌شه، مراحل باقی‌مونده رو روی استیکرها می‌نویسم و جاهایی که بیشتر جلوی چشممه می‌چسبونم تا زمانی که به اون هدف برسم. همین فرایند رو برای هدف‌های مربوط به گروه انجمن علمی دانشجویی کامپیوتر هم پی گرفتم که فکر می‌کنم همین هم یکی از دلایل اصلی نتیجه گرفتن و شکوفایی این گروه شد.

### ۳) داشتن انگیزه

روزهایی هست که ممکنه سرما خورده باشی، خسته باشی، از پروژه‌ات نتیجه‌ای که باید رو نگرفتی و یا حتی بی‌دلیل بی‌انگیزه باشی و ترجیح بدی برای یک مدت هیچ کاری نکنی. توی همچین شرایطی افراد اطرافت احتمالاً نمی‌تونن کار زیادی برات بکنن.

من یاد گرفتم چیزی که «مریم» رو از سایرین در شرایط مشابه متفاوت می‌کنه، اینه که مریم خودش، به خودش انگیزه می‌ده تا کاری برای تغییر شرایط بکنه. این یادگیری توی دنیای حرفه‌ایم هم خیلی برام کاربردی شده، چون اگر توی فرازونشیب‌های درس و یا کار قرار بگیرم، خودم رو به طور کامل مسئول نتایج کار خودم می‌دونم، در نتیجه به روش‌های مناسب به خودم انگیزه می‌دم که اصولاً باعث شادی درونم هم می‌شه!-) و این ناخودآگاه باعث می‌شه راه‌های جدیدی برای حل مسئله‌ی پیش روم پیدا کنم.

یه موردش که الان یادم می‌آد مربوط به زمانی می‌شه که به عنوان مدیر مسئول نشریه‌ی پردازش و با همکاری شبانه‌روزی جمعی از



دانشجوها، اولین شماره‌ی نشریه‌ی پردازش رو به منظور گسترش دانش روز مرتبط به علوم کامپیوتر چاپ کرده بودیم، ولی برخلاف انتظارمون استقبال چندانی در زمان فروش نمی‌شد. خاطریم هست که طی فروش دو روز اول نشریه شماره سوم سال ۱۳۹۲، بچه‌های تیم هر روز در ساعت‌های مشخص شده توی غرفه‌ای نزدیک به سلف دانشگاه حاضر می‌شدن، اما به خاطر استقبال کم دانشجوها، تیم هم داشت انگیزه‌اش کم‌رنگ می‌شد.

برای همین صبح روز سوم یک تعداد هندوانه‌ی سرخ تهیه کردم و خودم به تیم فروش ملحق شدم. تبلیغ اون روز تبدیل شد به «نشریه به شرط چاقو» و هربار به هرکسی که نشریه رو می‌خرید و همین‌طور به اعضای تیم یه قاچ هندونه تعلق می‌گرفت!-) با همین ایده‌ی ساده نه تنها تمام نسخه‌های اون شماره به فروش رفت، بلکه توی اون هوای گرم، خنکی اون هندوانه‌ی شیرین برای تیم «پردازش» خاطرهای موندگار و دلیل انگیزشی برای بهبود سطح رضایت‌شون از بیشتر دیده شدن ماحصل کارشون توسط دیگران شد.

#### ۴ تمرکز

یک شبانه روز برای همه‌ی آدم‌های روی کره‌ی خاکی زمین، ۲۴ ساعته. پس این به نظر یک بهانه می‌آد که برای انجام یک کار بگیم: وقت کم آوردیم، درحالی‌که آدم‌های دیگه‌ای هستند که تونستن همون کار یا مشابه اون رو انجام بدن. عاملی که باعث تفاوت در میزان کارآمدی ۲۴ ساعت آدم‌های مختلف می‌شه، اینه که آیا بر روی جهت درستی که مد نظرشونه، تمرکز می‌کنند یا نه. طی سال‌های کارشناسی‌ام سعی کردم که تمرکز کردن رو به عنوان یکی از هنرهای مربوط به مدیریت زمان تمرین کنم.

توی چهار سال، مشغول کارهای متفاوت زیادی شدم، از جمله عضویت در گروه درسی پروژه‌ی شش ماهه‌ی تولید نرم‌افزار، حل تمرین دروس متعدد، مدیر مسئول و طراح گرافیک نشریه، مدیر انجمن علمی دانشجویی کامپیوتر، تولیدگر گروه‌های تفریحی و مسئول برگزاری دوره‌های آموزشی. برای همین برای انجام کارهای مختلف، زمان‌های جدایی در طی روز و هفته اختصاص می‌دادم و به امور و وظایف مربوط به هر کار به صورت جداگانه می‌پرداختم که فکر می‌کنم همین

امر هم به پیشرفت کارهای گروهیم کمک شایانی کرد. می‌خوام بگم اگرچه می‌گن خانوم‌ها می‌تونن چندین کار رو به صورت هم‌زمان انجام بدن، اما این به این معنا نیست که الزاماً همه‌ی آن‌ها را به همان خوبی انجام خواهند داد که جدا جدا برای هر کدام وقت بگذارن!

#### ۵ قدرت تصمیم‌گیری

اینکه تصمیم‌گیری با چه کسی و در چه زمانی ملاقات بکنی، چه نیروهایی رو برای انجام چه فعالیت‌هایی جذب کنی، چه قدر زمان و منابع انسانی و مالی و سازمانی داری و چقدرش رو می‌تونی برای چه کاری اختصاص بدی، چه کاری رو به پایان برسونی و چه کاری رو رها کنی، همه‌ی



شناخت و تعامل با افراد و گروه‌های مختلف در سطوح مختلف (از جمله رییس دانشگاه، مدیر همایش‌های ملی کشوری و مسؤول حراست یا برنده‌های نوبل) دارین. برای همین خوبه که همیشه گوشه‌ی ذهنمون داشته باشیم که اقدامات و لینک‌های به ظاهر کوچک، یک زمانی می‌تونن نتایج خوب و قابل توجهی رو برامون به همراه داشته باشند. ملموس‌ترین مثالی که احتمالاً همه‌ی شما هم تجربه‌اش کردین، کمک‌های سره‌زنگ آقای حاج‌آقای هست، در شرایطی که شاید بعضاً جز شرح وظایفشون هم نبود، اما دریغ نمی‌کردند:)

در مجموع فعال بودن در زمینه‌های غیر درسی در دانشگاه می‌تونه کمک بزرگی در راستای تمرین کردن برای بهبود این توانایی باشه و به علاوه کمک کنه تا افراد توانایی (نه) گفتن در شرایط لازم رو در خودشون تقویت کنن.

#### ۷ مفیدتر بودن

اگر کسی می‌خواد خودش و کارش رو به بالاترین حد استانداردهای موجود برسونه، بایستی هر چیزی که از حد استاندارد اون شخص پایین‌تره به چالش بکشه. برای مثال اگه شخصی وظیفه‌ای رو به شما

محول می‌کنه اما شما به هر دلیلی فکر می‌کنید اون وظیفه، کاری مناسب با توانمندی‌ها یا خواسته‌های شما نیست، بایستی از کسی که بهتون این کار رو محول کرده درباره‌ی چرایی این انتساب سوال بپرسین و یا دلایل خودتون رو برای عدم تمایل برای انجام اون کار ابراز کنید. این باعث می‌شه که بخش زیادی از زمان شما برای کارهایی که برای شما مفیدتره و یا شما تخصص بیشتری در انجامش دارین، ذخیره بشه.

متأسفانه این یکی از شایع‌ترین اتفاقاتی هست که می‌افته: در واقع در این حالت فرد بالادست، کاری رو به شما محول می‌کنه، فقط به خاطر این‌که این آسون‌ترین اقدام ممکنه برای اون شخصه؛ نه به خاطر این‌که این بهترین و سنجیده‌ترین تصمیم برای شماست. بزرگی می‌گه «زمان با ارزش‌ترین دارایی هر انسانه». پس بهترین کار قبل این‌که بخواید زمانتون رو صرف کاری که بهتون واگذار شده بکنین، اینه که با توجه به عوامل مختلف ببیند، انجام این کار تا چه

این‌ها تصمیم‌های سختی هستن که گرفتارنشون در زمان مناسب مهمه، چرا که نه فقط روی زندگی خودمون، بلکه روی حال و آینده‌ی سایرین که باهامون در تعامل و همکاری هستند هم تاثیر می‌ذاره. برای من سخت‌ترین بخش در بین کارهایی که توی اون سال‌ها انجام دادم همین بود که سعی کنم تصمیماتی که می‌گیرم، منافع جمعی افرادی که باهاشون در تعامل اعم از حدود ۱۵ نفر عضو تحریریه‌ی نشریه، ۲۰ نفر فعال در انجمن، ده‌ها دانش‌آموز کلاس‌هایی که حل تمرینشون بودم و تمام دانشجویهای فنی و مهندسی دانشگاه و به صورت خاص گروه کامپیوتر رو تامین کنه.

#### ۶ تعامل اجتماعی با دیگران

برای موجودی اجتماعی به نام انسان، تقویت مهارت برقراری ارتباط و مذاکره کردن، لازمه‌ی پیشرفت توی زندگی اجتماعیه. شما برای شناخته شدن خودتون و کارتتون و همین‌طور گرفتن حمایت‌های مالی و انسانی و حتی عاطفی، نیاز به

حد شما رو به موفقیت نزدیک می‌کنه. (تعریف موفقیت به عهده‌ی خودتون!:-))

## ۸) مهار استرس

برای هرکسی در هر جایگاهی ممکنه شرایطی پیش بیاد که منطبق با پیش‌بینی‌هاش نبوده باشه و فرد دچار نگرانی و اضطراب بشه.

این اضطراب می‌تونه روی ذهن شما، عملکرد فعلی‌تون و تصمیم‌های بعدی که می‌گیرین اثر بذاره. همین‌طور این استرس توی ظاهر شما هم تاثیر خودش رو نشون می‌ده و این می‌تونه قضاوت ظاهری که دیگران درباره‌ی شما می‌کنن رو به سمت

همین امر باعث می‌شد به خودمون، توانمندی‌هامون، برنامه‌هامون و به پیشنهاداتمون اعتماد کنن، منابعشون رو در اختیارمون بذارن و حمایتمون کنن تا به هدف‌های پیش‌بینی شده نزدیک‌تر بشیم.

## ۹) کار گروهی

من همیشه فردی اجتماعی بوده و هستم:-) و این مسئله همیشه دو پیامد فوق‌العاده رو برام به همراه داشته: پیشرفت و شادی بیشتر!:-)

تجربه‌ی من این رو می‌گه که اگر تیم مناسبی تشکیل داده باشید، کاری که شاید به تنهایی امکان



انجام تمام و کمالش نبوده، با همکاری در فعالیت گروهی، با کیفیت بالاتر و در زمان کمتری انجام می‌شه. جالب‌تر از اون اینه که اگر چه افراد در یک تیم همه مثل هم نیستند و فکرها، خواسته‌ها و نیازهای بسیار متفاوتی با هم دارن، اما همه‌ی اعضای تیم یک‌سری ویژگی‌های مشترک با هم دارند که همین مسئله اون‌ها رو در کنار هم جمع می‌کنه و باعث افزایش روحیه‌ی همکاری‌شون می‌شه.

مثل ویژگی تمایل به خوشحال کردن و خوشحال شدن، گرایش به هنر و زیبایی و یا علاقه به یاد گرفتن مطالب اضافه بر درس. مشارکت در کارهای داوطلبانه که مزاد شرح وظایف کاری هر کسیه، می‌تونه لذت اعضای تیم رو از محیط کاری و یا درسشون افزایش بده و دل‌های آدم‌ها رو به هم دیگه نزدیک‌تر کنه.

یادمه یکی از کارهایی که خیلی مورد استقبال افراد تیم انجمن کامپیوتر و نشریه قرار گرفت، این بود

که هر از چندگاهی دوره‌های زنگ تفریح طور داشتیم! اون زمان‌ها که انجمن تازه پا گرفته بود، اگر می‌خواستیم مسابقه‌ای برای بیشتر درگیر کردن دانشجویها با امور علمی و فرهنگی برگزار کنیم، منابع مالی کافی برای تهیه‌ی جایزه‌اش نداشتیم، برای همین توی این دوره‌ها به همدیگه روش‌های درست کردن دستبندهای مختلف رو یاد می‌دادیم و وقتی که این دستبندهای رنگی خیلی زیبا آماده می‌شد، اون‌ها رو همراه به عنوان جایزه به برنده‌های مسابقات می‌دادیم. توی یکی دیگه از این دوره‌ها، یک روش برای بافتن شال‌گردن یاد گرفتیم و به همه‌ی اعضای تیم برگزارکننده‌ی برنامه‌ی «جشن روز مهندسی» که تازه بناش رو شروع کرده بودیم دادیم تا خوش‌رنگ‌تر و متحدتر بشن:-)

## ۱۰) تهیه‌ی گزارش

یکی دیگه از عواملی که به من کمک می‌کرد تا بتونم ذهنم و کارهام رو مرتب کنم، آماده و به‌روز کردن گزارش از تقریباً همه چیز بود! از رکورد کردن کتاب‌ها و صندوق مالی برای سیستم «جابه‌جایی کتاب‌های علمی بین دانشجویها و فارغ‌التحصیل‌ها» که طی سال آخر کارشناسیم راه اندازیش کردم، تا جمع کردن و آرشیو کردن تمام فرم‌های اداری که پر کرده بودیم و شرح وظایف افراد در هر تیم که توی دفتر ثبت شده بود.

داشتن یک سیستم برای نظم دادن به پرونده‌ها، ذخیره‌ی اطلاعات و گزارش‌ها، نه تنها باعث بهبود عملکرد خودمون و گروهمون در زمان حال می‌شه، بلکه افراد بعد از ما هم می‌تونن از این منابع استفاده کنن و کارشون رو بر پایه‌ی کارهایی که ما انجام دادیم بنا کنن و پیش ببرن. به این ترتیب با استفاده از این روش، زمان قابل توجهی رو برای آیندگان ذخیره می‌کنیم که در صورت عدم وجود این رکوردها می‌بایست مجدداً این راه‌ها رو آزمون و خطا طی کنن.

## ۱۱) صبر و بردباری

تمرین کنیم که صبور باشیم:-) مهارت مدیریت به معنی انجام دادن همه‌ی کارها نیست، بلکه به معنی به پایان رساندن کارهای مهمه.

عجله کردن احتمال بروز اشتباه در کار رو افزایش می‌ده و معمولاً جبران کردن این اشتباهات پیش اومده در نهایت منجر به صرف زمان بیشتری می‌شه در مقایسه با زمانی که همان کار با صبر و حوصله بیشتر انجام می‌شد.

## ۱۲) بخشش

بخشش چیزی نیست که ناخودآگاه خودش پیش بیاد، ما آدم‌ها باید انتخاب کنیم که ببخشیم.

بی‌اعتمادی بیره. پس سعی کنیم استرس‌مون رو مدیریت کنیم!:-)

برای من به عنوان مدیر انجمن علمی دانشجویی کامپیوتر، لازم بود تا به امور علمی، فرهنگی، مالی، اداری و مدیریتی آشنا باشم و متناسب با شرایط مختلف تصمیمات مناسب بگیرم. گاهی یک تصمیمی که گرفته می‌شد، ممکن بود منطبق با علایق فعالان فرهنگی باشه، اما برای گروه مالی به نظر جالب توجه و اجرایی نیاد. چیزی که کمک کرد تا اکثراً بتونم رضایت و همکاری گروه‌های مختلف رو برای اجرایی شدن برنامه‌هایی که انجمن تصمیم به برگزاریش داشت بگیرم، این بود که؛ همیشه در ملاقات با این گروه‌ها سعی می‌کردم هم خودم و هم اعضای تیم با اعتمادبه‌نفس بالا ظاهر بشیم و برنامه رو همراه با هدف نهایی و مشکلات احتمالی که ممکن بود به همراه داشته باشه باهاشون صادقانه در میون بذاریم.

من نیاز به این مهارت رو این روزها که در پروژه‌های بزرگتری مشغول به فعالیتیم، بیشتر از قبل حس می‌کنم، چرا که بنا به شرایط موجود با آدم‌های بیشتری که برای پروژه انتساب شده‌اند، همکاری دارم. یکی از جدیدترین تجربه‌ها مربوط به زمانیه که برای پروژه‌ی ساخت سخت‌افزار و نرم‌افزار یک ربات ۹ ماه با تیمم در دانشگاه UCSD کار کرده بودم و پروژه از سمت ما آماده‌ی تست بود، اما تیم با یه دانشجو از گروه هنرهای تجسمی هم همکاری می‌کرد که کار تولید پوست برای صورت ربات رو به عهده داشت. این شخص علی‌رغم تعهداتش نتونست کار رو طی زمان مشخص به ما تحویل بده، در نتیجه ما قادر نبودیم تست‌هایی رو که برای سنجش ربات در تعامل با انسان بود به موقع اجرا کنیم و نتایج رو برای چاپ در کنفرانس مورد نظرمون بفرستیم.

گاهی ممکنه شما اشتباهی کنین که در کارتون تاخیر و یا اختلال ایجاد کنه. به طور مشابه، افراد دیگه هم (به احتمال زیاد سهوی) ممکنه اشتباهی بکنن که باعث ایجاد مشکل در کار شما بشه. این امریه که می‌تونه به قدری زیاد روی شما تاثیر بذاره که تمرکز روی کارتون رو از دست بدید و باعث بشه که به صورت احساسی برخورد کنید؛ اما شما باید یاد بگیرید که اشتباهات رو ببخشید، بپذیرید که اتفاق افتادن اشتباه امری محتمل بوده و تمرکز کنید روی روشی برای درست کردن اشتباه اتفاق افتاده. این باعث می‌شه روابط شما با اطرافیان تون بهتر بشه و مقدار زمانی که ممکن بود صرف دنبال کردن اشتباه بشه رو به زمان مفید تبدیل کنید و ازش استفاده کنین.

برای پروژه‌های که براتون مثال زدم، خودم به شخصه وارد جزئیات کار هنری پروژه شدم و بعد از چندین جلسه با دانشجوی هنرهای بصری، محدودیت‌های کار رو پیدا کردم و با کمکش گزارشی دقیق از پیشرفت‌ها و اشتباهاتی که طی تولید پوست ایجاد شده بود تهیه کردیم. متوجه شدم نیاز به علوم زیست‌شناسی داریم پس لینک جدیدی با این دپارتمان زد، دانشجوی جدیدی رو با قابلیت‌های مورد نیاز به تیم اضافه کردم تا توی تولید پوست برای ربات کمک کنه و این بخش کار سرعت بگیره. در آخر به موازات این فرایند شروع به تست سیستم و الگوریتم نرم‌افزاری ربات ساخته شده کردم و الان نتایج تست نرم‌افزار رو برای همون کنفرانس و برای ددلاین تعیین شده آماده کردم. به علاوه برای تیم هنری زمان بیشتری خریدم تا رباتمون رو برای به دست آوردن نتایج تحقیق تعامل ربات با انسان تکمیل کنن. اگر از اون اشتباه به موقع نمی‌گذشتم و به موقع به فکر راه حل نمی‌افتادم، به احتمال قوی نه کار من و نه کار تیم پیشرفت نمی‌کرد.

## چی شد به فکر کار در انجمن و شروع نشریه توی دانشکده افتادید؟ اون موقع ترم چند بودید؟

متأسفانه از سال اول احساس کمبود تعاملات علمی خارج از کلاس، فعالیت‌های فرهنگی، آموزشی و گردهمایی‌های علمی پژوهشی رو توی گروهمون حس کردم. اون زمان ۳ نفر عضو انجمن علمی بودن که پایان اون سال تحصیلی هر سه فارغ‌التحصیل شدن و من و دو دوست همیشه همراهم مهندس مهسا شیخ‌حسینی و مهندس مریم مسعودی مسئولیت کارهای انجمن رو به عهده گرفتیم. در ابتدا مسئولیت انجام تمام امور علمی و اداری و فرهنگی و مالی به عهده‌ی ما بود، اما بعد از چندین مرحله تبلیغات در خصوص برنامه‌های انجمن و اهدافی که براش تعیین کرده بودیم، با سازمان‌دهی زیرگروه‌ها و عضوگیری، اعضای انجمن به حدود ۱۵ نفر رسید که هر کدوم بسته به علاقتون در زیرشاخه‌ها شروع به فعالیت کردند.

## کی فارغ‌التحصیل شدید؟

ترم هشتم، بهار ۱۳۹۳. ۴ سال درس مهندسی خوندن و فارغ‌التحصیل شدن به اندازه‌ی گرفتن نسخه‌ی فیزیکی مدرک فارغ‌التحصیلی سخت نبود!))

## خاطره‌ای دارید از دوران تحصیل تون که بخواین برامون بگید.

توی یک ظهر گرم تابستون ۱۳۹۳ که برای گرفتن مدرک فارغ‌التحصیلی‌ام به دانشگاه رفته بودم، یکی از دانشجویهای سال پایینی رو که اتفاقاً عضو انجمن علمی بود، دیدم. ازم پرسید حالا که فارغ‌التحصیل شدی و داری از این دانشگاه می‌ری چه حسی داری؟ جواب دادم حس نگرانی! نگرانم که

نتیجه‌ی کارهایی که توی این سال‌ها انجام دادم، موندگار نباشه و بعد از رفتنم از بین بره. جواب داد «تو انجمنی رو سرپا کردی و نشریه‌ای رو راه انداختی که ما دانشجویهای کامپیوتر بتونیم در اون حرف‌هامون رو بزنیم، تعلیم ببینیم و تعامل بیشتری با جامعه‌ی علمی و فرهنگی داشته باشیم. زحمتهایی که تو کشیدی امکاناتی رو برای دانشجویهای این دپارتمان فراهم کرده که قبلاً نداشتند. ما هیچ‌وقت تو و زحمتهات رو فراموش نمی‌کنیم.» جمله‌های صمیمانه‌اش جوری به قلبم نفوذ کرد که تا به امروز فراموش نکردم.

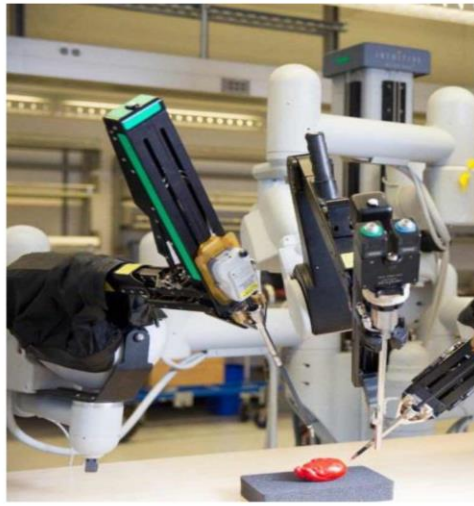
اون روز، آخرین روزی بود که به عنوان دانشجوی مهندسی دانشگاه الزهرا (س) به این دانشگاه رفتم و همین‌طور اولین روزی که تاثیر حقیقی سخت‌کوشی‌های چهار سال فعالیت‌های علمی و فرهنگی در این دانشگاه رو دیدم.

طی سالیان بعد بارها خبرهای خوشی از دوستانم و دانشجویهای سال پایینی درباره‌ی پیشرفتی که این گروه‌ها داشتن، بهم رسیده. آخرین خبر که بسیار خوشحالم کرد این بود که نشریه‌ی دانشجویی پردازش توی لیست پنج نشریه‌ی برتر کشوری قرار گرفته. خبرش برای یک هفته انرژی رو مضاعف کرد!-)»

## از اونجایی که هم در ایران و هم در خارج از کشور تحصیل کردید، دیدتون نسبت به دانشگاه‌های ایران چیه؟ این‌که در چه جایگاهی هستن؟ نقاط قوت و ضعف شون چیه؟

به نظرم اصلی‌ترین نقطه قوت دانشگاه‌های ایران، برخورداری از قابلیت و توانمندی و استعداد سرشار نسل جوانه که می‌تونه به مدد همت و تدبیر مسئولین





امر، فضای رشد و شکوفایی پیدا کنه. یکی از مواردی که دانشگاه‌های سطح کشور باید اهتمام بیشتری نسبت به اون داشته باشن، برنامه‌ریزی جهت کشف، شناسایی و پرورش استعدادهاست. استفاده از پتانسیل نیروی جوان در سطح دانشگاه هم می‌تونه برای دانشگاه، ارزش‌افزایی بکنه و برای دانشجویها هم موجب تجربه‌اندوزی و مهارت‌افزایی بشه.

مسئله بعدی تقویت ارتباط صنعت و دانشگاهه. دانشگاه و صنعت می‌تونن پروژه‌های مشترکی رو اجرا کنند. صنعت پتانسیل این رو داره که نیروهای مورد نیازش رو از بین دانشجویان مستعد، جستجو و استخدام کنه. پس چه بهتر که دانشگاه شرایط رو برای تربیت افرادی که مورد نیاز صنعت هستن، فراهم کنه و این امکان رو در اختیار دانشجویان و صنعت‌گران بذاره که با هم ملاقات و نشست‌های حضوری داشته باشن تا شناخت بهتری نسبت به نیازهای طرفین بوجود بیاد.

### ▪ شاغل هستید؟ حوزه فعالیت تون رو برامون می‌گین؟

من در ایالت کالیفرنیا، شهر سن-دیگو زندگی می‌کنم. با شروع دوره‌ی دکتری، کار و منبع درآمد تحقیق و انجام پروژه‌های علمی پژوهشی، حساب می‌شه. در حال حاضر در گروه contextual robotics institute در دانشگاه UCSD همکاری می‌کنم و ربات‌های انسان‌نما برای کاربردهایی در بهداشت و درمان می‌سازیم.

### ▪ توی ایران هم کار کردید؟ محیط‌های کاری ایران و خارج از کشور چه تفاوت‌هایی دارند؟

من بلافاصله بعد از فارغ‌التحصیلی از مقطع کارشناسی اقدامات لازم مثل امتحان تافل و جی ار ای رو انجام دادم و به فاصله‌ی کمی بعد از اون پذیرش گرفتم و برای شروع تحصیلات ارشدم مهاجرت کردم. بعد هم تا به امروز در محیط دانشگاهی مشغول تحصیل و تحقیق بودم، به همین خاطر تجربه کاری قابل‌ذکری ندارم.

### ▪ ایران در حوزه‌ی کاری ما در جهان چه جایگاهی داره؟ چه پتانسیل‌هایی داره؟ نقاط قوت و ضعف‌اش چیه؟

علم کامپیوتر از سرعت پیشرفت بسیار بالایی برخورداره و هر لحظه در حال تغییره. هر روز خبری از یه گوشه از دنیا می‌رسه از نتیجه‌ی تحقیقات فرد یا گروهی که تحولی اساسی در علم روز ایجاد

باور داشته باشیم و بدونیم هرکس که توانمند و در عین حال خلاق باشه با تلاش می‌تونه به موفقیت برسه. خانم دکتر نوشین ریاحی که در زمان دانشجویی من مدیر گروه کامپیوتر دانشگاه بودن، نمونه‌ی خوبی از بانوان کشور در مهندسی کامپیوتر هستن که سازماندهی و نظم‌شون تاثیر بسیار مثبتی در روند تحصیلی دانشجویها گذاشته.

### ▪ اگر برگردید به عقب همین مسیر رو انتخاب می‌کنید؟ (چرا؟)

همیشه به خودم می‌گم هدف خوب داشتن مهمه و باید حواسمون باشه انرژی خودمون رو صرف کارهای مفید کنیم. فکر می‌کنم تا به این لحظه انرژی و استعدادهام رو صرف راه و هدف درستی کردم، برای همین قطعاً اگر به عقب برگردم باز هم همین رشته رو انتخاب می‌کنم.

### ▪ نصیحت و پیشنهادی به بچه‌ها

به تمام دوستانی که قدم توی مسیر علم‌آموزی و یا علم‌پروری گذاشتن، پیشنهاد می‌کنم بیشتر به ندای درونی‌تون گوش کنید؛ استعدادهاتون رو شناسایی و تقویت کنید. از جمله مهارت‌های مهم در دنیای امروز، خلاقیت و قدرت حل مسئله هست، مهارت‌هایتان را افزایش دهید. برای خودتون هدف‌های کوتاه‌مدت و بلندمدت تعیین کنید و برای رسیدن بهشون برنامه‌ی مرحله به مرحله بریزید.

پدر، مادر و یا هرکسی که شما رو بزرگ کرده بیشترین شناخت رو نسبت به شما داره، ضعف‌ها و قوت‌هاتون رو می‌دونه. حرف‌هاشون رو بشنوین و اگر بهتون توصیه‌ای کردن بهش فکر کنین و اگر تونستین بهش عمل کنین.

### ▪ سوالی بود که دوست داشتید بپرسیم و ما نپرسیدیم؟

فکر کنم این طولانی‌ترین مصاحبه چند سال اخیرتون باشه، امیدوارم نکاتی که تا اینجا گفتیم برای خواننده‌های مجله جالب و مفید بوده باشه.:

### ▪ در نهایت صحبت دیگه‌ای دارید برامون؟

در آخر این‌که: بهترین‌ها رو برای جوانان سرزمینم آرزومندم.

کرده‌اند. من باور دارم که مهم‌ترین سرمایه‌های یک کشور، سرمایه انسانی و این سرمایه است که موجب کسب مزیت رقابتی می‌شه. کشور ما سرشار از سرمایه انسانی. کافیه این سرمایه‌ها را بیشتر دریابیم و فضای رشد و پرورش اون‌ها رو در داخل کشور فراهم کنیم تا بیش از پیش شاهد بالندگی نخبگان کشورمون در جای‌جای جهان باشیم. یکی از مناسب‌ترین روش‌ها برای این منظور می‌تونه افزایش دسترسی دانشجویها به منابع علمی و مقالات به روز دنیا باشه که شرایط رو برای آگاهی از علم روز و رقابت آگاهانه برای پیشرفت‌های علمی فراهم می‌کنه.

### ▪ در آمریکا چجوریه؟ بانوان در مهندسی کامپیوتر چقدر فعال و تاثیرگذار هستند؟ آیا اونجا هم بین بانوان و آقایون در این حوزه تفاوتی هست؟

به نظر می‌رسه تبعیض جنسیتی، به‌خصوص در فضاهای کاری مرتبط با علوم کامپیوتر امری شناخته شده در بیشتر کشورهای دنیاست که بسیاری از متفکرین (اعم از زن و مرد) در حال تلاش برای اصلاحش هستن. برای مثال ماه آینده توی تکزاس آمریکا بیش از ۲۰,۰۰۰ نفر در گردهمایی سالانه‌ای با محوریت زنان در علم کامپیوتر و تکنولوژی جمع می‌شن تا تعلیم‌های لازم برای برابری جنسیتی و نحوه‌ی افزایش موقعیت‌های علمی و شغلی برای بانوان رو آموزش و گسترش بدن.

با این اوصاف جهان داره به سمت و سویی می‌ره که توانمندی و مهارت فرد، تعیین‌کننده‌تر و پررنگ‌تر از مسائل جنسیتی می‌شه. این مسئله خاص کشوری هم نیست. افراد می‌تونن در هر حوزه‌ای حرفی برای گفتن داشته باشن به شرط این‌که به دانش روز اون حوزه اشراف داشته باشن. به همین خاطر یه راه معقول اینه که خودمون رو

# آشنایی با

## SDN

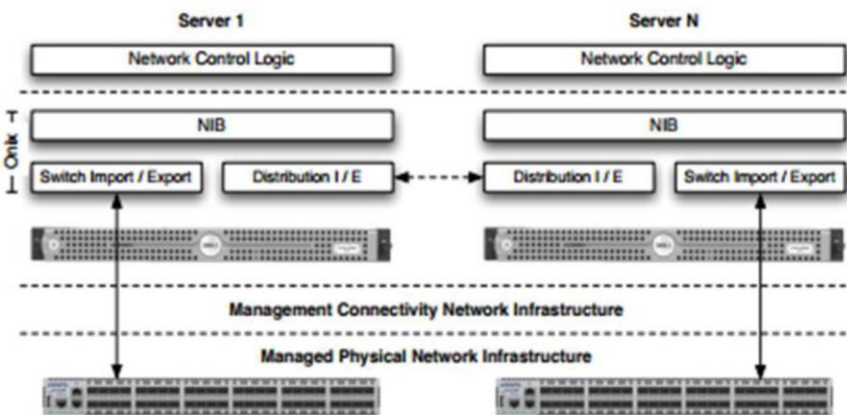
(قسمت دوم)

✓ زیربنای فیزیکی: شامل سوئیچها و مسیریابهای شبکه است که واسطی را پشتیبانی می‌کنند که آن واسط Onix را قادر می‌سازد وضعیتی را بخواند و بنویسد که رفتار عناصر (مثل درایه‌های موجود در جدول انتقال) را کنترل می‌کند. این عناصر نیازی به اجرای هیچ نرم‌افزار دیگری ندارند.

✓ زیربنای اتصال: ارتباط بین شبکه فیزیکی و Onix (ترافیک کنترل) انتقال را در اتصالات زیرساخت کنترل می‌کند. این کانال کنترلی هم می‌تواند به صورت In-band باشد (که در آن ترافیک کنترل همان عناصر ارسالی را به اشتراک می‌گذارد که ترافیک داده در شبکه به اشتراک می‌گذارد) و هم می‌تواند به صورت Out-band باشد (که در آن یک شبکه فیزیکی مجزا برای اداره ترافیک کنترل استفاده می‌شود). زیربنای اتصال باید ارتباط چند وجهی بین نمونه‌های Onix و سوئیچها برقرار کند و در صورت لزوم از همگرایی در خطای Link پشتیبانی کند. پروتکل‌های مسیریابی استاندارد) مثل IS-IS یا OSPF برای ساخت و نگه داری وضعیت ارسال در زیربنای اتصال مناسب هستند.

✓ Onix: یک سیستم توزیع شده است که در خوشه‌ای از یک یا چند سرور فیزیکی اجرا می‌شود. هر سرور هم می‌تواند شامل چند نمونه از سیستم‌های Onix باشد Onix. به عنوان پایگاه کنترل موظف است به منطق کنترل توانایی برنامه ریزی شبکه را بدهد (هم نوشتن و هم خواندن وضعیت شبکه). یک نمونه Onix همین‌طور برای منطبق شدن با مقیاس شبکه‌های بزرگ (با میلیون ها پورت) و برای فراهم کردن قابلیت برگشت پذیری ضروری برای توسعه تولید، موظف است

می‌شوند محیط‌هایی به گوناگونی WAN را هدف قرار می‌دهند. ابر عمومی (public cloud) و مرکز داده شرکت هم از محیط‌های هدف آن است. دوم، Onix اصول ابتدایی توزیع منطقی را فراهم می‌کند (مثل حافظه DHT و عضویت گروه) که طراحان اپلیکیشن را قادر می‌سازد بدون اینکه مکانیزم‌های توزیع را بازسازی کنند، اپلیکیشن‌های کنترل را به کار بگیرند و انعطاف‌پذیری را طوری نگه دارند تا تبدلات عملکرد/مقیاس پذیری همانند آنچه الزامات اپلیکیشن دیکته می‌کند انجام شوند.



شکل ۶: اجزای Onix

برای درک اینکه چگونه Onix به یک پایگاه کنترل تولید- کیفیت تحقق می‌بخشد لازم است درباره دو جنبه طراحی آن بحث کنیم: بستری که Onix در قالب آن در شبکه قرار می‌گیرد و API ای که Onix در دسترس طراحان اپلیکیشن قرار می‌دهد.

### ۱-۷-۱- اجزا

طبق شکل ۶ شبکه‌ای که با Onix کنترل شود از چهار جز تشکیل می‌شود که هر کدام نقش متمایزی دارند.

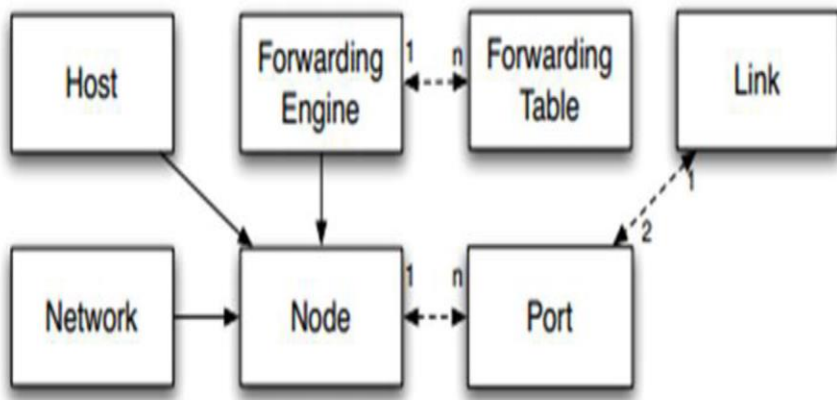
### ۷-کنترل‌کننده Onix

باینکه تاکنون سیستم‌هایی متعددی ارائه شده‌اند که از الگوی اصلی SDN پیروی می‌کنند، اما هنوز مطالب کمی منتشر شده که نشان دهد چگونه پایگاه کنترلی بسازیم که به همه الزامات و نیازها پاسخگو باشد. برای پر کردن این شکاف، سیستم‌عامل Onix معرفی شد.

همان‌طور که در شکل ۶ می‌بینید، چهار جز در یک شبکه کنترل Onix وجود دارد: زیربنای فیزیکی مدیریت شده، زیربنای اتصال، Onix و

منطق کنترل که توسط اپلیکیشن‌های مدیریتی به کار گرفته شده است. این شکل دو نمونه‌ی هماهنگ Onix را نشان می‌دهد که دیدشان را از وضعیت شبکه به اشتراک می‌گذارند و به منطق کنترل یک واسط خواندن/نوشتن به وضعیت پیشنهاد می‌کنند (با فلش تیره نشان داده شده است).

مهم‌ترین فایده Onix برای کاربردهای فعلی به دو بخش تقسیم می‌شود: اول، Onix نسبت به سیستم‌های قبلی API عمومی‌تری را در اختیار قرار می‌دهد؛ پروژه‌هایی که بر روی Onix ساخته



شکل ۷: کلاس‌های موجودیت‌های پیش فرض شبکه ارائه شده توسط API اونیکس

نمونه‌های Onix با استفاده از کدگذاری توزیع شده و الگوریتم‌های همه پذیر پخش شده است دیکته کند.

برای ساده کردن موضوع، ما فرض می‌کنیم NIB تنها موارد فیزیکی موجود در شبکه را شامل می‌شود. اگرچه از نظر عملی به سادگی می‌توان آن را برای پشتیبانی عناصر منطقی (مثل کانال‌ها) توسعه داد.

### 7-3- جزئیات مبنای اطلاعاتی شبکه

به طور کلی، NIB مجموعه‌ای از موارد موجود در شبکه را در برمی‌گیرد که هر کدام دارای مجموعه‌ای از زوج‌های key-value است و با یک شناساگر جهانی مسطح ۱۲۸ بیتی شناسایی می‌شود. این عناصر موجود در شبکه ساختار پایه را تشکیل می‌دهند و همه انواع دیگر موجود در شبکه از آن ناشی می‌شوند. Onix دسته بندی قوی‌تری را برای عناصر دسته بندی شده پشتیبانی می‌کند. سپس انواع دسته‌بندی شده مجموعه‌ای از پیش تعیین شده از ویژگی‌ها و روش‌ها را شامل می‌شوند (با استفاده

(NIB) نام دارد که بسیار شبیه به مبنای اطلاعاتی مسیریابی (RIB) است که مسیریاب‌های IP از آن استفاده می‌کنند. NIB هم پیشوندها را در مقاصد ذخیره می‌کند و هم گراف‌ها را از همه عناصر موجود در توپولوژی شبکه دارد. NIB قلب مدل کنترلی Onix است و مبنایی است برای مدل توزیع شده Onix. اپلیکیشن‌های کنترل شبکه می‌توانند در NIB بنویسند و از آن بخوانند (برای مثال حالت ارسال را اصلاح می‌کنند یا به شمارنده‌های پورت دستیابی پیدا می‌کنند) و Onix با تکرار و توزیع NIB بین موارد مختلف در حال اجرا قابلیت مقیاس پذیری و برگشت پذیری را (در چارچوبی که اپلیکیشن معین کرده) فراهم می‌کند.

Onix در زمانی که تکرارها و توزیع داده‌های NIB را اداره می‌کند، برای یافتن و فراهم کردن راه حل برای تضادها در وضعیت شبکه به منطق مخصوص اپلیکیشن متکی است. تضاد در وضعیت شبکه حین تبادل داده بین نمونه‌های Onix و همین طور بین یک نمونه Onix و یک عنصر شبکه روی می‌دهد. منطق کنترل می‌تواند ضمانت‌های سازگاری را برای وضعیتی که بین

وضعیت شبکه را به نمونه‌های دیگر درون خوشه ارسال کند.

✓ منطق کنترل: منطق کنترل شبکه در رأس API اونیکس قرار دارد. این منطق کنترل رفتار مطلوب و مورد انتظار شبکه را تعیین می‌کند؛ Onix صرفاً اصول ابتدایی لازم برای دستیابی به وضعیت مناسب شبکه را فراهم می‌کند.

این‌ها چهار جز اصلی شبکه مبتنی بر SDN هستند. فرض می‌کنیم که زیربنای فیزیکی می‌تواند بسته‌ها را بسیار سریع (معمولاً دو یا چند برابر سریع‌تر) انتقال دهد. سریع‌تر از آن که Onix (یا هر پایگاه عمومی کنترل دیگری) بتواند آن را پردازش کند. از این نظر استفاده از Onix برای اجرای عملکردهای مدیریتی مناسب نیست. عملکردهایی که برای شناخت تغییرات در وضعیت شبکه به ازای هر بسته (یا هر تغییر سریع دیگری) نیاز به منطق کنترل دارند.

### API-7-2

کامک اصلی Onix تعریف یک API مفید و عمومی برای کنترل شبکه است که توسعه اپلیکیشن‌های مقیاس پذیر را امکان‌پذیر سازد. در ادامه‌ی کارهای قبلی، API و اونیکس را حول یک شبکه فیزیکی طراحی شده است تا اپلیکیشن‌های کنترل را قادر باشند وضعیت را برای هر عنصر در شبکه بخوانند یا بنویسند. از این رو این API با محوریت داده، روش‌هایی را برای پایدار نگه داشتن شبکه بین عناصر درون شبکه‌ای و اپلیکیشن کنترلی) که بر روی نمونه‌های متعددی از Onix قابل اجرا هستند (ارائه می‌کند. به طور خاص، API اونیکس شامل یک مدل داده می‌باشد که زیربنای شبکه را با هر عنصر شبکه که به یک یا چند حامل داده مربوط می‌شود نشان می‌دهد. منطق کنترل می‌تواند: وضعیت فعلی مربوط به آن حامل داده را بخواند، با کار بر روی این حاملان وضعیت شبکه را تغییر دهد و گزارش‌های تغییر وضعیت این حاملان را ثبت کند. علاوه بر این، از آن جا که Onix باید طرح‌های کنترلی بسیاری را پشتیبانی کند، پایگاه به منطق کنترل اجازه می‌دهد مدل داده را تغییر دهد و بر استقرار و پایداری هر جز شبکه کنترل داشته باشد.

یک کپی از وضعیت شبکه که Onix مسیریابی می‌کند در یک ساختمان داده ذخیره می‌شود. این سازه داده‌ای مبنای اطلاعات شبکه

Category	Purpose
Query	Find entities.
Create, destroy	Create and remove entities.
Access attributes	Inspect and modify entities.
Notifications	Receive updates about changes.
Synchronize	Wait for updates being exported to network elements and controllers.
Configuration	Configure how state is imported to and exported from the NIB.
Pull	Ask for entities to be imported on-demand.



از زوج (key-value) و بر اساس این ویژگی‌ها عمل می‌کند.

برای مثال، یک کلاس موجودیت Port وجود دارد که می‌تواند به لیستی از پورت‌ها در موجودیت Node متعلق باشد. شکل ۷ مجموعه اصلی موجودیت‌های دسته بندی شده را نشان می‌دهد که Onix ایجاد می‌کند - موارد موجود در هر دسته در یک کلاس پایه‌ای مشترک محدود به دسترسی عمومی زوج key-value قرار می‌گیرند. چیدمان دسته‌ها در Onix ثابت نیست و اپلیکیشن‌ها می‌توانند این کلاس‌های اصلی را به کلاس‌های ریزتری دسته بندی کنند تا در صورت لزوم مدل داده Onix را توسعه دهند.

در شکل ۷، خطوط توپر نشان دهنده‌ی ارث بری و خط چین‌ها مربوط به ارتباط ارجاعی بین نمونه‌های موجودیت‌ها می‌باشند. اعداد روی خط چین‌ها نمایانگر رابطه‌های نگاشت کمی هستند (برای مثال یک لینک به دو پورت نگاشت می‌شود و دو پورت می‌توانند به همان لینک نگاشت شوند). گره‌ها، پورت‌ها و لینک‌ها توپولوژی (وضعیت مکانی) شبکه را شکل می‌دهند. تمامی کلاس‌های موجودیت از همان کلاس پایه‌ای ارث می‌برند که دسترسی زوج key-value را فراهم می‌کند.

NIB برای اینکه منطق کنترل بتواند به موجودیت‌های شبکه دسترسی یابد، راه‌های مختلفی را در اختیار می‌گذارد. همچنین فهرستی از تمام موجودیت‌ها بر اساس شناساگر آن‌ها تهیه و نگهداری می‌کند تا جستجوی مستقیم برای یک موجودیت خاص را امکان پذیر سازد. علاوه بر این از ثبت گزارش‌های تغییر وضعیت یا اضافه/حذف شدن موجودیت‌ها پشتیبانی می‌کند. اپلیکیشن‌ها بعداً می‌توانند با توجه به گزارش‌های ورودی یک

موجودیت در شبکه و نگهداری فهرست خودشان، قابلیت جستجو را توسعه دهند.

همه عملکردهای NIB ناهماهنگ (اسنکرون) هستند. به این معنی که به روز کردن یک موجودیت در شبکه فقط این را تضمین می‌کند که پیام به روز رسانی بالاخره به عنصر مربوطه در شبکه و/یا دیگر نمونه‌های Onix ارسال خواهد شد، اما تأخیر تضمین نمی‌شود.

## ۸- مقیاس پذیری و قابلیت اطمینان

Onix برای اینکه بتواند جایگزینی برای ساختار سنتی شبکه باشد، باید به الزامات مقیاس پذیری و قابلیت اطمینان شبکه‌های امروز و آینده پاسخگو باشد. از آن جا که NIB نقطه مرکزی وضعیت سیستم و رویدادها محسوب می‌شود، استفاده از آن در سطح بزرگ مقیاس پذیری و قابلیت اطمینان سیستم را مشخص می‌کند. برای مثال، همزمان با کاهش تعداد عناصر شبکه، NIB ای که توزیع نشده باشد باعث از بین رفتن حافظه سیستم می‌شود. یا تعداد رویدادهای شبکه (رویدادهایی که NIB ایجاد کرده (یا میزان کاری که برای مدیریت آن‌ها باید انجام شود زیاد می‌شود و CPU یک نمونه واحد Onix را اشباع می‌کند.

### ۸.۱- مقیاس پذیری

Onix از سه راهبرد برای بهبود مقیاس پذیری استفاده می‌کند: اول، اپلیکیشن‌های کنترلی را قادر می‌سازد بارکاری را تفکیک کنند. از این رو اضافه کردن نمونه‌ها بار کاری را کم می‌کند بدون اینکه صرفاً آن را تکرار کند. دوم، Onix تراکم را امکان پذیر می‌سازد. در تراکم، شبکه‌ای که با خوشه‌ای از گره‌های Onix مدیریت می‌شود به عنوان یک گره واحد در NIB برای یک خوشه مجزا دیده می‌شود. این امر دسته‌های موجود در Onix را متحدالشکل و ساختار آن را سلسله مراتبی می‌کند. از این رو باعث می‌شود میزان اطلاعات مورد نیاز درون یک خوشه واحد Onix کاهش یابد. در نهایت، Onix اپلیکیشن‌هایی را در دسترس قرار می‌دهد که بر روی پایداری و دوام وضعیت شبکه کنترل دارند. به طور کلی:

تفکیک: (Partitioning) منطق کنترل شبکه می‌تواند Onix را طوری پیکربندی کند که یک نمونه کنترلر به خصوص تنها یک زیرمجموعه از NIB را در حافظه نگه داری کند و آن را به روز نگه دارد. علاوه بر این، یک نمونه Onix می‌تواند اتصالاتی به یک زیرمجموعه از عناصر شبکه داشته باشد تا تعداد رویدادهایی که عناصر ایجاد می‌کنند و این نمونه باید پردازششان کند کمتر شود.

متراکم سازی: (Aggregation) در تنظیمات متشکل از چند Onix، یک نمونه از Onix می‌تواند گروهی از عناصر درون NIB خود را به عنوان یک عنصر متراکم به یک نمونه Onix

دیگر نشان دهد. این امر معمولاً برای نشان دادن پیچیدگی کمتر به لایه‌های بالایی در سلسله‌ای از کنترلرهای Onix انجام می‌شود. برای مثال، در یک محوطه بزرگ شبکه، هر ساختمان با یک کنترلر (Onix یا خوشه‌ای از کنترلرها) مدیریت می‌شود. این کنترلر تمام عناصر موجود در آن ساختمان را به عنوان یک گره متراکم به یک نمونه جهانی Onix که وظیفه مهندسی ترافیک بین محوطه‌ها را به عهده دارد نشان می‌دهد. این موضوع مشابه مفهوم الگوهای مدیریت جهانی کنترل در شبکه‌های ATM است.

**سازگاری و دوام (Consistency and durability):** وضعیت شبکه‌ای که آن را مدیریت می‌کند دیکته می‌کند. این کار با پیاده سازی هر کدام از الگوریتم‌های سازگاری مورد نیاز برای رسیدن به وضعیت سازگار انجام می‌پذیرد. برای وضعیت‌هایی که در آن استفاده از این الگوریتم‌ها، سازگاری را تضمین نمی‌کند، مسیریابی و تفکیک ناسازگاری به ایجاد پایداری کمک می‌کند. به طور پیش فرض، Onix دو نوع پایگاه ذخیره داده ارائه می‌کند که اپلیکیشن می‌تواند از آن‌ها برای حالاتی استفاده کند که ترجیحات متفاوتی برای سازگاری و دوام دارند. برای اپلیکیشن‌هایی که به دنبال دوام و سازگاری قوی‌تری هستند، Onix یک پایگاه داده تراکنشی تکرار شونده پیشنهاد می‌کند و برای وضعیت‌های فرار که تحملشان در برابر عدم پایداری بیشتر است یک DHT تک هاپ مبتنی بر حافظه ارائه می‌کند.

مکانیزم‌های مقیاس پذیری فوق می‌توانند برای مدیریت شبکه‌هایی استفاده شوند که بسیار بزرگ‌تر از آن هستند که تنها با یک نمونه‌ی Onix کنترل شوند. برای نشان دادن این موضوع، از یک مثال کاربردی استفاده می‌کنیم: اپلیکیشنی را در نظر بگیرید که می‌تواند بین سوئیچ‌ها در توپولوژی مدیریت شده مسیر ایجاد کند، با این هدف که مسیرهای کاملی درون شبکه به وجود آورد.

تفکیک: شبکه‌ای را فرض می‌کنیم که تعدادی متعادل از سوئیچ‌ها دارد و می‌تواند به سادگی توسط یک نمونه Onix اداره شود. در نتیجه، تعداد و سائز همه ورودی‌های ارسالی وضعیت بر روی شبکه، منابع حافظه یک سرور فیزیکی واحد را اشغال می‌کنند.





اطلاعات در دسترس تر هستند ولی ماندگاری در پایگاه داده را ممکن است نداشته باشد.

API هر دو مورد فوق در Onix پیش بینی شده است و بسته به انتخاب طراح و نیاز برنامه از هر کدام می‌خواهد می‌تواند استفاده کند.

معمولاً ایده کلی این است که اطلاعات توپولوژی و ساختار کلی شبکه و ویژگی‌های آن‌ها در یک پایگاه داده با ماندگاری بالا (SQL) ذخیره شود. اطلاعاتی که داخل مسیریاب‌هاست مثل جداول forwarding و چیزهایی که حالت موقتی دارند و زیاد تغییر می‌کنند در DHT ذخیره شود.

### ۹.۳- مدیریت وضعیت عناصر شبکه

طراحی Onix یک پروتکل خاص را برای مدیریت وضعیت ارسال عناصر شبکه تحمیل نمی‌کند. برعکس، نخستین واسط برای اپلیکیشن NIB است و هر پروتکل مناسبی که عناصر شبکه پشتیبانی کنند، می‌تواند برای هماهنگ نگه‌داشتن NIB مستقل با وضعیت واقعی شبکه استفاده گردد. برای مثال OpenFlow یکی از پروتکل‌های مدیریت وضعیت عناصر شبکه است که Onix آن را حمایت می‌کند و در بخش‌های قبل به توضیح داده شد.

### ۹.۴- سازگاری و هماهنگی

یک NIB مرکز تعامل منابع داده‌ای متعددی است (نمونه‌های دیگر Onix و نیز عناصر شبکه). مکانیزم‌های توزیع وضعیت به طور مستقیم با یکدیگر تعامل ندارند بلکه آن‌ها وضعیت را به NIB Import یا Export می‌کنند. برای پشتیبانی اپلیکیشن‌های متعدد با نیازهای مقیاس پذیری و قابلیت اطمینان احتمالاً خیلی متفاوت، Onix نیاز به اپلیکیشن‌هایی دارد تا مشخص کنند که چه داده‌ای به یک منبع خاص وارد یا از آن خارج شود. اپلیکیشن‌ها این کار را از طریق پیکربندی ماژول‌های Import و Export انجام می‌دهند.

NIB منابع داده را بدون نیاز به سازگاری زیاد ادغام می‌کند و در نتیجه وضعیت به روزرسانی می‌شود تا به NIB اضافه شود. در اینجا ممکن است به علت ناهماهنگی، وضعیت در هر منبع داده (DHT) یا به دلیل ناهماهنگی بین منابع داده‌ها تناقض ایجاد شود. بر این اساس Onix اپلیکیشن‌هایی را پیش بینی می‌کند که منطق

تفکیک ناهماهنگی را در سیستم عامل نشان می‌دهند.

### ۱۰- شبیه ساز Mininet

شبیه ساز Mininet شبکه‌ای مجازی همراه با کرنل‌های واقعی و سوودو کدهای برنامه است و می‌تواند سوئیچ‌های شبکه را بر روی یک ماشین (ماشین مجازی، ابر یا سیستم واقعی) به اجرا درآورد. این نرم‌افزار به دلیل امکان تعامل پذیری توسط خط فرمان و یا API می‌تواند در راستای توسعه، آموزش و تحقیق استفاده شود. همچنین این نرم‌افزار قابلیت‌های گسترده‌ای برای تست، ارزیابی و توسعه ابزارهای مبتنی بر شبکه‌های SDN و OpenFlow دارد.

### ۱۱- کارهای مرتبط

همان طور که بیان شد، Onix کار سیستم را کم می‌کند و در آن صفحه کنترل از صفحه داده مجزاست، ولی تمرکز Onix بر تبدیل شدن به یک پایگاه کنترل تولید - کیفیت برای شبکه‌های با مقیاس بالا، ما را قادر می‌سازد تا نسبت به سیستم‌های قبلی، بیشتر بر روی قابلیت اطمینان، مقیاس پذیری و عمومیت تمرکز کنیم. این سیستم اولین سیستمی نیست که کنترل شبکه را به عنوان یک مشکل سیستم توزیع شده در نظر می‌گیرد.

تحقیقات زیادی انجام شده که تمرکزشان بر ارائه یک صفحه ارسال توسعه پذیر به توسعه دهندگان شبکه بوده است (مانند Routebricks ، Click ، Onix ، XOPR) در ارائه یک صفحه کنترل توسعه پذیر، مکمل این سیستم‌ها محسوب می‌شود. به طور مشابه، Onix می‌تواند برای ساختارهای انعطاف پذیر مرکز داده شبکه، مثل SEATTLE ، vL2 و Portland پایگاهی برای مدیریت مراکز داده بزرگ باشد.

دیگر کارهایی که اخیراً انجام شده‌اند با توزیع وضعیت شبکه بین سوئیچ‌ها، از بار موجود در یک کنترلر متمرکز می‌کاهند Onix. به جای اینکه بر روی پیدا کردن یک رویکرد خاص در مقیاس مشخص تمرکز کند، بر روی مسئله فراهم کردن API‌های کلی مدیریت وضعیت توزیع شده متمرکز است.

Onix همچنین راه بسیاری از سیستم‌های توزیع شده قبل از خود را در پیش گرفته است.

سیستم‌هایی که برای پاسخ به الزامات پایداری به کمک اپلیکیشن‌ها متکی هستند تا بدین وسیله کارایی تکرار وضعیت را بهبود بخشند Bayou ، PRACTI ، WheelFS و PNUTS نمونه‌هایی از این سیستم‌ها می‌باشند.

### ۱۲- نتیجه گیری

SDN یک محصول یا مفهوم سخت‌افزاری/نرم‌افزاری نیست بلکه یک معماری و رویکرد جدید برای انعطاف پذیری و کنترل پذیری بیشتر شبکه‌ها و ظرفیت سازی برای استفاده از انواع برنامه‌های کاربردی، سرویس‌ها و خدمات نرم‌افزاری روی شبکه‌های کنونی است. از وظایف و نقش‌های پررنگ سخت‌افزار و تجهیزات شبکه می‌کاهد و به وظایف و نقش‌های لایه‌های نرم‌افزاری شبکه می‌افزاید و مدیریت و کنترل شبکه را ساده‌تر می‌کند.

الگوی SDN از پایگاه کنترل برای ساده سازی کاربردهای کنترل شبکه استفاده می‌کند. پایگاه کنترل به جای اینکه توسعه دهندگان را مجبور کند مستقیماً با جزئیات زیرساخت مواجه شوند، مسائل سطح پایین‌تر را خود رفع و رجوع می‌کند و توسعه دهندگان را قادر می‌سازد تا منطق کنترلشان را در یک API سطح بالا برنامه ریزی کنند. در این صورت، Onix مسائل شبکه را به مسائل سیستم‌های توزیعی تبدیل می‌کند تا با مفاهیم و الگوهای آشنا برای توسعه دهندگان سیستم‌های توزیعی قابل حل باشد.

لازم به ذکر است که Onix به تنهایی نمی‌تواند تمامی مشکلات مدیریت شبکه را حل کند. هنوز هم لازم است طراحان اپلیکیشن‌های مدیریتی کاربردهای مقیاس پذیری طراحی خود را درک کنند Onix. ابزارهایی عمومی برای مدیریت وضعیت فراهم می‌کند، ولی نمی‌تواند تمام مسائل و مشکلات مربوط به مقیاس پذیری و سازگاری را هموار سازد. ما هنوز درباره ساختن منطق کنترل در API و نپیکس در حال یادگیری هستیم، ولی در نمونه‌هایی که تا به حال با آن‌ها روبرو شده‌ایم ساخت اپلیکیشن‌های مدیریت با Onix بسیار ساده‌تر از ساخت آن‌ها بدون Onix است.

یکی از جدیدترین فناوری‌هایی که کاربرد پرداخت آن نیز به سرعت در حال گسترش است، فناوری RFID است. سیستم اصلی RFID شامل سه قسمت است: یک آنتن، یک فرستنده و گیرنده و یک RF آنتن یک پل بین برچسب، و فرستنده و گیرنده است RFID. در محدوده‌های گوناگون فرکانسی شامل ۱۲۵KHz، ۱۳۰۵۶MHz، ۲۰۴۵GHz، ۵۰۸GHz و ۹۵۰-۸۶۰ MHz کار می‌کند. هم‌چنین محدوده‌های خواندن را از چند سانتی متر تا ۵ متر ارائه می‌کند که به فرکانس دستگاه‌ها وابسته است. فناوری RFID از امواج رادیویی به منظور شناسایی اشیاء استفاده می‌کند. یک سامانه عبارت است از یک برچسب حاوی یک آنتن و

یک تراشه به همراه اطلاعاتی در مورد کالا یا اشیاء و هم‌چنین یک قرائت‌گر که دارای یک فرستنده و گیرنده رادیویی می‌باشد. کاربردهای اولیه این فناوری در بهبود کارایی سامانه‌های اطلاعاتی زنجیره‌های تامین و توسعه فروشگاه‌های بزرگ زنجیره‌ای بوده است. لکن کاربردهای این فناوری در حال گسترش بوده، به طوری که در ردیابی کالا، افراد، و حفاظت از مناطق امن از RFID استفاده شده است. نگهداری تراکنش‌ها و خدمات پرداخت فناوری RFID نیز از جمله کاربردهای آن می‌باشد که در خریدهای خرد روزانه شهروندان به کار گرفته شده است.

در این مقاله قصد داریم که با فناوری RFID مزایا، معایب، کاربرد ها و چالش‌های آن نظیر انواع حملات و راه‌های مقابله با این چالش‌ها بپردازیم.

## ۱- مقدمه

امروزه ضرورت شناسایی خودکار عناصر و جمع‌آوری داده مرتبط به آنان بدون نیاز به دخالت انسان جهت ورود اطلاعات در بسیاری از عرصه‌های صنعتی، علمی، خدماتی و اجتماعی احساس می‌شود. در پاسخ به این نیاز تاکنون فناوری‌های متعددی طراحی و پیاده‌سازی شده است. به مجموعه‌ای از فناوری‌ها که از آنان برای شناسایی اشیاء، انسان و حیوانات توسط ماشین استفاده می‌گردد، شناسایی خودکار و یا به اختصار Auto ID گفته می‌شود. هدف اکثر سیستم‌های شناسایی خودکار، افزایش کارایی، کاهش خطا و ورود اطلاعات و آزاد سازی زمان کارکنان برای انجام کارهای مهمتر نظیر سرویس دهی بهتر به مشتریان است.

تاکنون فناوری‌های مختلفی به منظور شناسایی خودکار طراحی و پیاده‌سازی شده است. کدهای میله‌ای، کارت‌های هوشمند، تشخیص صدا، برخی فناوری‌های بیومتریک، OCR (برگرفته شده از optical character recognition) و RFID نمونه‌هایی در این زمینه می‌باشند.

در حال حاضر شرکت‌های بزرگی مثل مک دونالد و وال مارت از این فناوری استفاده می‌کنند. در ادامه با فناوری RFID بیشتر آشنا خواهیم شد.

## ۱-۲ RFID چیست؟

RFID سامانه شناسایی امواج رادیویی (به انگلیسی: Radio Frequency Identification) (به اختصار RFID) سامانه شناسایی بی

سیمی است که قادر به تبادل داده‌ها به وسیلهی برقراری اطلاعات بین یک Tag که به یک کالا، کارت و... متصل شده و یک بازخوان (Reader) است. سامانه‌های RFID از سیگنال‌های الکترونیکی و الکترومغناطیسی برای خواندن و نوشتن داده‌ها بدون تماس بهره می‌برند.



## ۲-۲ آشنایی با فناوری RFID

تصور کنید که وارد یک فروشگاه زنجیره‌ای شده‌اید و اقلام مورد نیاز خود را داخل چرخ دستی قرار داده‌اید. صندوق دار با استفاده از بار کد باید تک تک اقلام داخل سبد را برداشته و اطلاعات آن را توسط بارکد خوان یکی یکی به داخل کامپیوتر وارد کند تا فاکتور اقلام انتخابی شما صادر گردد. بسیاری از اوقات به دلیل آنکه تعداد کالاهای خریداری شده بسیار زیاد می‌باشند؛ صف‌های طولانی‌ای در فروشگاه‌های زنجیره‌ای مشاهده می‌شود. گاهی اوقات نیز مخدوش شدن علائم بار کد، از خواندن اطلاعات جلوگیری می‌کند، که این خود موجب مشکلات بیشتری می‌شود. اما با این فناوری جدید یعنی RFID شما سبد کالای خود را برمی‌دارید و بدون اینکه مجبور به ایستادن در صف‌های طولانی شوید و یا حتی بدون اینکه مجبور باشید اقلام خریداری شده را به صندوقدار یا نگهبان نشان دهید، از در خارج می‌شوید.

حتما می‌پرسید چرا؟ چون شناسه روی کالا دیگر بارکد نیست بلکه از نوع RFID می‌باشد و خودش با فرستان علائم رادیویی کلیه اطلاعات جاری خود از قبیل تعداد، قیمت، وزن، ... را به کامپیوترهای موجود در درب‌های خروجی مخابره و منتقل می‌کند.

## ۳- اجزای یک سیستم RFID

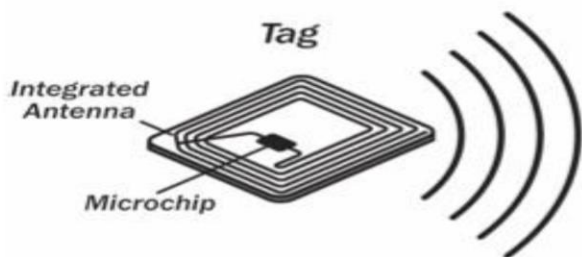
۱-۳ شناسه یا Tag

۲-۳ بازخوان برچسب یا Reader

۳-۳ آنتن

۴-۳ نرم‌افزار مدیریت اطلاعات

۳-۱ Tag یا شناسه: این شناسه‌ها دارای دو بخش تراشه و آنتن هستند و دارای عملکرد بسیار ساده‌ای می‌باشند؛ تراشه اطلاعات را از طریق آنتن منتشر می‌کند و حسگرهایی که در اطراف قرار دارند، این اطلاعات را دریافت می‌کنند.



شکل ۱- یک شناسه rfid

## تقسیم بندی انواع tag ها

### ۱-۱-۳ دسته بندی اصلی

Tag های غیر فعال، باتری ندارند و انرژی لازم برای فعال سازی پردازنده و ارسال اطلاعات خود را از سیگنالی که از آنتن Reader دریافت می‌کنند، تأمین می‌کنند؛ بدین ترتیب که ابتدا Reader امواجی را به سمت Tag ارسال می‌کند، سپس پردازنده Tag با استفاده از انرژی این امواج، بیدار شده و دستور دریافتی از سوی Reader را پردازش می‌کند و توسط امواجی که انرژی آن‌ها هم از امواج دریافتی تأمین شده است، نتیجه را به Reader مخابره می‌کند. بدین ترتیب واضح است که برد مسافتی این Tag ها بسیار محدود است؛ چرا که فاصله Tag از آنتن باید به اندازه ای باشد تا انرژی که Tag دریافت می‌کند، توان بیدار کردن پردازنده و تأمین انرژی موج برگشتی را داشته باشد. معمولاً برد Tag های غیر فعال از ۵ سانتی متر تا ۵ متر است.

### ۱-۲-۳ Tag های غیر فعال Passive Tags

که انرژی و برق مورد نیاز خود را از Reader ها به وسیله یکسری از روش‌های تراگسیل بدست می‌آورند.

### ۲-۲-۳ Tag های فعال Active Tags

که انرژی مورد نیازشان توسط یک باتری داخلی و جهت برقراری ارتباط دارای یک پردازنده، یک حافظه و حسگر می‌باشند.

### ۳-۲-۳ Tag های نیمه غیر فعال Semi-Passive Tags

که علاوه بر استفاده از باتری داخلی شان، می‌توانند از انرژی منتقل شده توسط بازخوان بر چسب Reader ها نیز بهره مند شوند.

### ۴-۲-۳ Tag های دو طرفه Two way Tags

که علاوه بر استفاده از باتری داخلی شان می‌توانند بدون کمک گرفتن از Reader ها دیگر اقسام هم شکل خود را نیز شناسایی کرده و با آن‌ها به گفتگو بپردازند.

## ۵- چالش ها

### ۱-۵ سازگاری نرم افزاری

سازگاری اطلاعات موجود بر روی برچسب های RFID با نرم افزارهای کاربردی موجود، یکی از جنبه های مهم معرفی فناوری RFID است. اطلاعات جمع آوری شده توسط RFID reader ها نیاز به تولید ارزش دارد. مثلاً وضوح بیشتر زنجیره عرضه و یا برنامه ریزی بهینه تر. این موضوع که بکارگیری RFID در ابعادی بسیار بزرگ در زنجیره تولید و عرضه ممکن است به انفجار اطلاعات جمع آوری شده منجر شود، امری بسیار حائز اهمیت است و این امر نیاز به نرم افزاری را که فاصله خالی بین اطلاعات دریافتی در RFID reader ها و اطلاعات موجود بر روی نرم افزارها را پر کند، به وجود می‌آورد؛ بنابراین جای تعجب نیست که شرکت های تولید کننده نرم افزارهای رابط (میان فرستنده ها و گیرنده ها)، موسسات برنامه ریزی ذخایر (ERP) و فروشندگان سیستم های مدیریت انبارداری، برنامه های خود را برای همکاری در زمینه تولید RFID اعلام کرده اند.

از طرف دیگر Tag های فعال، انرژی لازم برای پردازنده و ارسال امواج به سمت آنتن را از باتری همراه خود تأمین می‌کنند. بدین ترتیب برد این Tag ها در برخی موارد تا چند صد متر هم می‌رسد. قیمت این Tag ها در حدود چند ده دلار است که به مراتب از Tag های غیر فعال (با قیمتی در حدود چند ده سنت) گران قیمت تر هستند. یک پارامتر مهم در انتخاب Tag های اکتیو یا همون فعال، طول عمر باتری آن است، چرا که پس از اتمام باتری، شما مجبورید باتری یا اغلب خود Tag را تعویض کنید و این باعث افزایش هزینه ها می‌گردد. بدین ترتیب، طول عمر باتری یک عامل تعیین کننده در هزینه نهایی سیستم شما خواهد بود. Tag های RFID هر کدام دارای یک کد منحصر به فرد می‌باشند به طوریکه هیچ دو برچسبی در دنیا تولید نخواهد شد که کد یا ID یکسانی داشته باشند و کلیه تولید کنندگان Tag تحت نظر انجمن بین المللی مبادرت به ثبت آنها می‌نمایند. این Tag ها از دو بخش اصلی تشکیل شده اند که بخش اول یک آنتن مایکرواستریپ بوده و بخش دوم یک چیپ الکترونیکی می‌باشد. لازم به ذکر است که این بخش ها به صورت کاملاً مسطح و بسیار کوچک هستند به طوریکه می‌توانند آنها را در درون کاغذ یک برچسب معمولی قرار دهند و از دید کاربر، تنها یک برچسب معمولی کاغذی خواهد بود. در بخش چیپ الکترونیکی Tag ها یک فرستنده و یک گیرنده امواج رادیویی به همراه مدارهای پردازشگر الکترونیکی قرار دارد. در برخی از انواع این چیپها، حافظه نیز به مقدار دلخواه وجود دارد به طوریکه هر نوع اطلاعات مربوط به کنترل و دسترسی می‌تواند مستقیماً روی برچسب و درحافظه ای آن نوشته شود. در این صورت این برچسب درست مانند یک شناسنامه الکترونیکی همراه محصول عمل خواهد کرد.

### ۲-۳ دسته بندی با در نظر گرفتن منبع انرژی تأمین کننده

در صورتی که بخواهیم Tag ها را با در نظر گرفتن منبع انرژی تأمین کننده شان دسته بندی کنیم به ۴ دسته اصلی تقسیم بندی می‌شوند:

حوزه RFID پرداخته اند. گروس شادل و تیلیچ، RFID بر مبنای کلید عمومی و خصوصی را بررسی کرده اند.

جولز و همکارانش، بحثی در مورد برچسب‌های RFID دارند و راهکارهای افزایش امنیتشان را بررسی کرده اند. کارجوت و مسکوویتز، برچسبی را پیشنهاد داده اند که امنیت بالاتری نسبت به نسل قبلی خود دارد. لینگل، روش جدیدی برای پویش کردن حوزه‌های RFID ارائه داده است. بورک، به مساله تراشه های RFID و امنیت آن‌ها می‌پردازد.

هدف ما در این مقاله این است که در ابتدا، بدافزارهای موجود در سیستم‌های مبتنی بر RFID را معرفی کرده و در نهایت چالش‌های امنیتی ایجاد شده توسط آن‌ها را ارائه دهیم.

## دسته بندی بدافزارهای RFID

بدافزارها، نرم افزارهای مخربی هستند که هدف اصلی‌شان تخریب سیستم‌های کامپیوتری می‌باشد. بدافزار در RFID، از طریق برچسب RFID منتقل شده و اجرا می‌شود. در این قسمت بدافزارهای رایج RFID معرفی می‌شوند.

### ۱-۲-۵ استعمارگر RFID

استعمارگر RFID، یک داده برچسب RFID می‌باشد که قسمتی از سیستم RFID را که با آن مواجه است، مورد بهره برداری قرار می‌دهد. وقتی قرائت‌گر RFID، یک برچسب را پویش می‌کند، انتظار دارد که اطلاعات را در قالب معینی دریافت کند. رخنه‌گر می‌تواند داده‌هایی تولید کند که قالب و محتویات آن با قالب مورد انتظار دستگاه قرائت‌گر یکسان نباشد و این باعث می‌شود که نرم افزار RFID قرائت‌گر و حتی پایگاه داده آن را تخریب کند و یک حمله از نوع وقفه را شکل دهد.

### ۲-۲-۵ کرم‌های RFID

کرم برنامه‌ای است که در طول شبکه انتشار می‌یابد و در طول انتشار خود، نقص‌های امنیتی را در سرویس‌هایی که به طور گسترده استفاده می‌شوند پیدا می‌کند. کرم از ویروس قابل شناسایی می‌باشد چرا که کرم‌ها نیازی به فعالیت کاربر برای انتشار ندارند. کرم‌های RFID از اتصالات شبکه، سوء استفاده می‌کنند تا خودشان را تکثیر کنند. این کرم‌ها توسط سوء استفاده از سرویس‌های برخط در RFID یا حتی از طریق برچسب‌های RFID، منتشر می‌شوند. کدهای موجود در کرم‌ها، موجب می‌شوند تا کارسازهای RFID، برخی از فایل‌ها را از مکان‌های دور، بارگیری کرده و سپس آن‌ها را اجرا کنند. این فایل‌ها در واقع کارساز میان‌افزار می‌باشند. کرمی که نرم افزارهای RFID را آلوده کرده است می‌تواند برچسب‌های RFID را نیز آلوده کند.

### ۳-۲-۵ ویروس‌های RFID

یک ویروس RFID، به صورت خودکار، کدش را در برچسب‌های RFID، بدون نیاز به اتصالات شبکه تکثیر می‌کند. ویروس‌های RFID خواه ناخواه یک بار مفید دارند که عملکرد سیستم پشتیبان RFID را تغییر می‌دهند یا تخریب می‌کنند.

اگر به پیشینه مقوله رخنه‌گری نگاهی بیندازید، خواهید دید با آمدن هر فناوری جدید، رخنه‌گران به دنبال رخنه‌های امنیتی آن گشته و با سوء استفاده از حفره‌های موجود در آن، فناوری را به چالش می‌کشند. رخنه‌گران با دسترسی و کنترل سیستم می‌توانند دست به اقدامات خرابکارانه زنند و باعث شوند هزینه‌ای هنگفت از حیث مادی و معنوی و همچنین خسارت‌های جبران‌ناپذیری به فرد و سیستم تحمیل شود. بدافزارها، نرم‌افزارهای مخربی هستند که هدف اصلی‌شان تخریب سیستم‌های کامپیوتری می‌باشد. همان‌طور که انتظار آن می‌رفت، بدافزارها به سیستم‌های مبتنی بر RFID نیز راه یافته و آن‌ها را به چالش کشیدند. اهمیت این مقوله و این‌که فناوری RFID از بدافزارها در امان نیست، ما را بر آن داشت که به بررسی تاثیر بدافزارها و چالش‌های امنیتی ایجاد شده توسط آن‌ها بپردازیم. با بررسی صورت گرفته سه نوع اصلی بدافزار در سیستم‌های مبتنی بر RFID شناسایی شده است. این سه نوع استعمارگرها، کرم‌ها و ویروس‌ها می‌باشند. بدافزارها منشأ خارجی داشته و بر اساس برچسب‌های RFID به وجود می‌آیند. در نهایت، تاثیر بدافزارها و چالش‌های امنیتی ایجاد شده توسط آن‌ها را ارائه داده‌ایم.

RFID امروزه در بسیاری از صنایع دنیا استفاده می‌شود. به عنوان مثال، در فروشگاه‌ها از RFID به جای سیستم قدیمی بارکد (رمزینه) استفاده می‌شود. نمونه دیگر در صنعت مسافری، گذرنامه‌های مسافران می‌باشد. همان‌طور که می‌دانید گذرنامه‌ها حاوی اطلاعات ارزشمند شخصی هستند و اگر یک رخنه‌گر بتواند رخنه‌ای در سیستم RFID گذرنامه بیابد، سوء استفاده‌های فراوانی از آن شخص و به نام او می‌تواند انجام دهد. با این‌که داده‌های موجود در گذرنامه‌ها رمزنگاری می‌شوند ولی باز هم سوء استفاده‌هایی در این زمینه دیده شده است. رخنه‌گری و سرقت اطلاعات از همان روزی شروع شد که هیچ رایانه‌ای وجود نداشت. با پا به عرصه گذاشتن رایانه و شبکه، رخنه‌گری و سرقت اطلاعات شیوه علمی به خود گرفت و روز به روز متکامل‌تر شد. کلا چهار نوع حمله می‌توانیم داشته باشیم:

- حمله از نوع وقفه: بدین معنا که حمله‌کننده باعث شود شبکه مختل شده و مبادله اطلاعات امکان پذیر نباشد.
- حمله از نوع استراق سمع: بدین معنا که حمله‌کننده به نحوی توانسته است اطلاعات در حال تبادل روی شبکه را گوش داده و بهره برداری نماید
- حمله از نوع دستکاری داده‌ها: یعنی حمله‌کننده توانسته است به نحوی اطلاعاتی که روی شبکه مبادله می‌شوند را تغییر دهد. بدین صورت که داده‌های دریافتی در مقصد، متفاوت با آن چیزی باشد که از مبدأ، ارسال شده است.
- حمله از نوع افزودن اطلاعات: یعنی حمله‌کننده اطلاعاتی را که در حال تبادل روی شبکه است، تغییر نمی‌دهد بلکه اطلاعات دیگری را که می‌تواند مخرب یا بنیان‌گذار حملات بعدی باشد، به اطلاعات اضافه می‌کند.

بر اساس این چهار نوع حمله‌ای که معرفی شد، حملات زیادی با نام‌های گوناگون به وجود آمده است. تحقیقات فراوانی در مورد RFID و امنیت آن صورت گرفته است که هر یک وارد بحث تفصیلی در این مقوله شده است.

فلدهوفر و همکارانش به بررسی الگوریتم رمزنگاری AES در







# 3D sensors

## سنسور چیست؟

حسگر یا سنسور، المان حس کننده‌ای است که کمیت‌های فیزیکی مانند فشار، حرارت، رطوبت، دما و... را به کمیت‌های الکتریکی پیوسته (آنالوگ) یا غیرپیوسته (دیجیتال) تبدیل می‌کند. در واقع سنسور یک وسیله‌ی الکتریکی است که تغییرات فیزیکی یا شیمیایی را اندازه‌گیری و آن را به سیگنال الکتریکی تبدیل می‌کند. سنسورها می‌توانند عمق فاصله یا محدوده را اندازه‌گیری کنند.

سنسورها در انواع دستگاه‌های اندازه‌گیری، سیستم‌های کنترل مانند plc مورد استفاده قرار می‌گیرند. به طور کلی دو نوع سنسور وجود دارد:

- **projected-light-sensor**: که این نوع سنسور نور را با الگوی دوربین دو بعدی استاندارد ترکیب کرده و عمق را اندازه‌گیری می‌کند.
- **Time-Of-Flight**: سنسورهای زمان‌سنجی هستند که عمق را با تخمین زدن تأخیر زمانی اندازه‌گیری می‌کنند. (تأخیر زمانی از انتشار نور تا تشخیص نور).

سنسورهای سه بعدی خطوط ارتفاع را تعیین می‌کنند. داده‌های سه بعدی می‌توانند با حرکت دادن اشیاء یا با حرکت سنسور ایجاد شوند. داده‌های سه بعدی اندازه‌گیری شده می‌توانند برای پردازش خارجی یا پردازش در سنسور برای اندازه‌گیری یا تشخیص مناسب، شبیه‌سازی شوند.

حسگرهای سه بعدی همچنان در مراحل اولیه‌ی توسعه قرار دارند. سنسورهای سه بعدی کاربرد وسیعی در زندگی مدرن دارند که در محدوده‌ی هواپیماهای بدون سرنشین، ربات‌ها؛ و به طور کلی برای دسترسی به زندگی واقعی استفاده می‌شوند.

ما می‌توانیم اقدامات خود را با حرکات ساده، حرکات بدن؛ و حتی تشخیص چهره، کنترل کنیم. لیزرهای پیشرفته به ویژه لیزرهای دیود و سنسورهای سه بعدی در حال تغییر، چگونگی تعامل با تکنولوژی هستند. واقعیت مجازی، واقعیت افزوده و وسایل رانندگی مستقل؛

تنها نمونه‌ای از برنامه‌های جدید در افق هستند. با گسترش سریع بازار سه بعدی و با افزایش نیاز به سنسورهای سه بعدی جای تعجب نیست که این سخت‌افزار در حال تبدیل شدن به قابلیت دسترسی بیشتر است.

حسگر سه بعدی را می‌توان با استفاده از تعدادی فناوری‌های مختلف به دست آورد. هر یک از این فناوری‌ها که مورد استفاده قرار می‌گیرند، دارای نقاط قوت جداگانه هستند

(Jabil optics) توانایی طراحی، توسعه و استفاده از سخت‌افزار حسگر سه بعدی بر اساس هر یک از سه تکنولوژی که در ادامه به توضیح آن خواهیم پرداخت، یا با استفاده از یک پلتفرم موجود یا طراحی ماژول به صورت جداگانه را دارد.

سه تکنولوژی فوق به شرح زیر است:

(۱) **بینایی استریوسکوپي**: این تکنولوژی از عملکرد چشمان انسان برای دیدن تصاویر گرفته شده است. دو دوربین در موقعیت‌های کمی انداز (درست مثل چشم‌های انسان) قرار می‌گیرند. سپس دو تصویر گرفته شده به یک تصویر متحد می‌شوند. تفاوت‌های کوچک ناشی از موقعیت‌های مختلف دوربین باعث ایجاد تصویر کلیشه‌ای، یعنی تصویر سه بعدی می‌شود. (دوربین استریو) یک سنسور است که به دید استریو متکی است و یک نقشه تقریبی از اطلاعات عمق در آن ذخیره می‌شود. مزیت دید استریو آن است که قادر به کار کردن در خارج از منزل است، اما دوربین‌های فعلی هنوز بسیار دقیق نیستند. علاوه بر این داده‌های تصویری استریو نیاز به زمان زیادی برای پردازش دارند و الگوریتم‌ها معمولاً بسیار محاسباتی هستند. چشم انداز استریوسکوپي یک کمک اضافی از یک ماژول نقطه در شیء یا صحنه برای کمک به تمرکز دوربین است.

بر اساس جدیدترین گزارش Bloomberg اپل مشغول توسعه یک سنسور سه بعدی مبتنی بر لیزر است که به احتمال زیاد در برترین آیفون سال ۲۰۱۹ به کار خواهد رفت. کاربرد این سنسور در اپلیکیشن‌ها و سرویس‌های واقعیت افزوده خواهد بود که تکنولوژی آینده متعلق به آن بوده و اپل بر روی آن سرمایه‌گذاری زیادی کرده است.

این سیستم با منعکس کردن لیزر از دستگاه و سپس اندازه‌گیری زمانی که بازگشت اشعه به طول می‌انجامد، یک نقشه تشکیل خواهد داد. چنین اطلاعات تفصیلی و عمیقی را می‌توان با استفاده از دوربین‌های دوگانه نیز به دست آورد، چنان‌که اپل نیز در حال حاضر این کار را در دستگاه‌های دارای دوربین دوگانه خود انجام می‌دهد. سیستم مشابه دیگر اپل، فیس آیدی است که با ایجاد یک نقشه دقیق از صورت کاربر، آن را از میان میلیون‌ها نفر تشخیص می‌دهد.

سیستم مشابه دیگر، در دوربین‌های پیکسل‌های جدید گوگل دیده می‌شود؛ اما سنسور سه بعدی مبتنی بر لیزر آتی اپل به مراتب قوی‌تر و دقیق‌تر خواهد بود. پیش از معرفی فیس آیدی نیز چندین گوشی‌ساز همچون سامسونگ از سیستم تشخیص چهره مخصوص خود در گوشی‌های موبایلشان استفاده می‌کردند، اما ویژگی تکنولوژی تشخیص چهره فیس آیدی، پیشرفته‌تر بودن آن است.

در برترین آیفون سال ۲۰۱۹، فیس آیدی همچنان حضور خواهد داشت؛ اما در پشت دستگاه نیز یک سنسور سه بعدی مبتنی بر لیزر به کار می‌رود که کاربرد مخصوص به خود را دارد.

اپل علاقه و توانایی خود را در تکنولوژی واقعیت افزوده به خوبی نشان داده است. یکی از اصلی‌ترین و جدیدترین ویژگی‌های آی‌اواس ۱۱ به‌عنوان نسخه سال ۲۰۱۷ پلتفرم موبایل اپل، قابلیت‌های مربوط به واقعیت افزوده است. این سیستم عامل در حاضر توانایی بسیار بالایی در این تکنولوژی دارد، اما با افزوده شدن یک سنسور سه بعدی مبتنی بر لیزر و توسعه دستگاه به‌صورت سخت‌افزاری، شاهد جهشی بزرگ در دنیای واقعیت افزوده خواهیم بود.

۲) **الگوی ساختار نور:** یک الگوی نور از هر دو خط، مربع (ساختارهای دوره‌ای) یا نقطه‌ها بر روی یک شی یا یک صفحه با یک ماژول پیش‌بینی می‌شود. یک دوربین مستطیلی ماژول طرح‌ریزی و سپس نور منعکس را ضبط می‌کند.

۳) **الگوی ایجاد شده توسط مثلث:** بین طرح‌ریزی و ماژول بین دو دوربین برای به دست آوردن جسم یا صفحه است.

۴) **زمان پرواز:** فلاش‌های مستقیم کوتاه یا فلاش‌های کوتاه سینوسی از طریق یک ماژول طرح‌ریزی منتشر می‌شوند و سپس توسط یک ماژول دوربین گرفته می‌شوند. زمان سفر نور از امیتر (emitter) به جسم و بازگشت به دوربین محاسبه می‌شود. سپس مختصات اندازه‌گیری شده یک تصویر سه بعدی ایجاد می‌کند.

هر تکنولوژی نقص‌هایی دارد. یکی از نقص‌های اصلی در ارتباط با تکنولوژی سنسورهای سه بعدی، میزان مصرف انرژی در هنگام مصرف و استفاده از آن است.

تولیدکنندگان سنسورهای تصویربرداری سه بعدی برای راه‌اندازی حسگرهای سه بعدی که دیدشان از دیوار می‌تواند نفوذ کند، تلاش می‌کنند. این ویژگی که هدف شرکت‌های کابلی، پهنای باند و بخش‌های هوشمند خانه است، راه طولانی را برای پیشرفت خواهد داشت.

شرکت‌های تلفن‌های هوشمند هم سعی دارند تا حسگرهای سه بعدی را در دستگاه‌های خود قرار دهند. به تازگی سامسونگ گلکسی نوت ۸ در نسخه ی خود دوربینی با تکنولوژی سه بعدی قرار داده است.

(تشخیص حرکت برای رابط کاربر نیز با حسگر سه بعدی انجام می‌شود. برخورد با یک منبع نور مادون قرمز در یک عنصر بصری در یک الگوی ساختار یافته یا یک ورق نور، باعث می‌شود که شما بتوانید بازی‌ها یا دستگاه‌های سرگرمی را با حرکات خود کنترل کنید.)

ما در جهان سه بعدی زندگی می‌کنیم که به معنای تجزیه و تحلیل رفتار انسان است. پس نیاز به وسایلی با درک اطلاعات سه بعدی است. با استفاده از سنسورهای سه بعدی می‌توان وضعیت واقعی پیام‌های محیطی را شناسایی کرد.

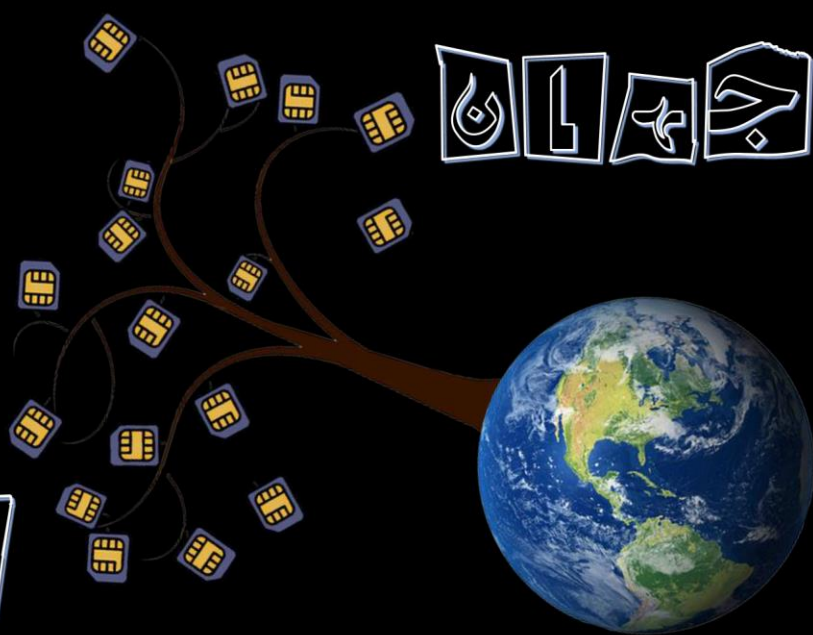
### کاربرد سنسور سه بعدی در گوشی همراه اپل (apple)

سنسور سه بعدی مبتنی بر لیزر جدیدترین تکنولوژی موبایلی است که اپل مشغول کار بر روی آن است. این سنسور موبایل در سرویس‌ها و برنامه‌های واقعیت افزوده کاربرد خواهد داشت.

اپل بعد از معرفی آیفون ایکس، که مهم‌ترین و پیشرفته‌ترین سیستم آن سیستم تشخیص چهره (فیس آیدی)

است، اعلام کرد توسعه این گوشی موبایل چند سال زمان برده است. پس بعید نیست این شرکت هم‌اکنون نیز مشغول توسعه یک تکنولوژی نوین برای آیفون‌های سال‌های بعد باشد.



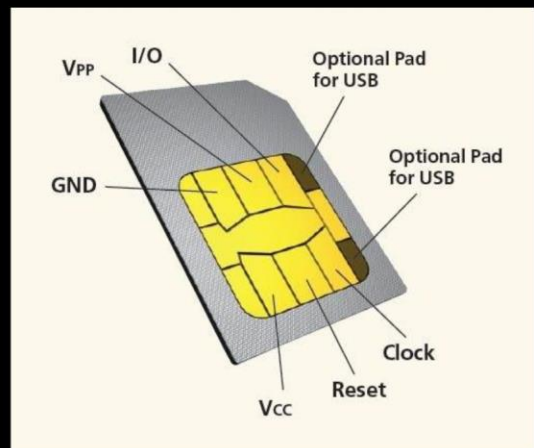


سیم کارت، یک کارت هوشمند، با دنیای پیچیده‌ی درونش، خیلی شناخته شده نیست. شاید بسیاری از شماها تا به حال فکر نکرده بودید که درون سیم کارت چند سانتی متری، چه دنیایی وجود دارد. در این مقاله قصد داریم که یک گوشه از دنیای درون سیم کارت را به شما نشان دهیم.

پس با ما همراه باشید...

#### • تعریف مقدماتی سیم کارت

ابتدا به سراغ قسمت سخت افزاری می‌رویم. سیم کارت‌ها ابزارهایی سخت افزاری هستند که تلفن‌های ما را قادر به برقراری ارتباط با اپراتورها می‌کنند و فضای بسیار کمی نیز برای ذخیره شماره‌ها و اطلاعات خود دارند. حال اگر بخواهیم بصورت تخصصی آن‌ها را بررسی نماییم، ابتدا نگاهی می‌کنیم به ظاهر سخت افزاری سیم کارت و تشریح بخش‌های مختلف آن:



VCC: منبع تغذیه

RST: ری‌استارت ارتباط و سیگنال

CLK: ایجاد سیگنال کلاک برای سیم کارت

GND: اتصال زمین

VPP: اختصاص یافته برای ولتاژهای برنامه نویسی

I/O: ورودی و خروجی

C4 - C8: که به رنگ تیره مشخص شده اند و برای ارتباطات خارجی از طریق یو اس بی و دیگر کاربردها طراحی شده اند.

یک سیم کارت شامل یک بخش چیپ می‌باشد که در واقع یک مموری ROM از ۶۴ تا ۵۱۲ کیلوبایت می‌باشد که سیستم عامل بر روی این بخش قرار دارد و در بخش EEPROM اطلاعات کاربری مانند تماس‌ها و اس ام اس و... ذخیره می‌شود. بسیاری از این سیم کارت‌ها با ولتاژ ۱.۸ - ۳ تا ۵.۵ ولت کار می‌کنند. کارخانه‌های تولید کننده سیم کارت غالباً از سه الگوریتم برای تولید سیم کارت استفاده می‌کنند:

- ۱) COMP128v1
- ۲) COMP128v2
- ۳) COMP128v3

امروزه نیز صرفاً سیم کارت‌های COMP128v1 قابل کلونینگ یا کپی می‌باشند، چرا که فعلاً صرفاً الگوریتم این نوع از سیم کارت کشف شده و هنوز در مورد دیگر الگوریتم‌ها اطلاعات چندانی در دسترس نمی‌باشد و قابل ذکر است که ۷۰٪ سیم کارت‌های امروزی از این الگوریتم بهره می‌برند.

#### • بخش نرم افزاری سیم کارت

ساختار زیر یک چارت کلی از فایل سیستم‌های موجود داخل سیم کارت است و این فایل سیستم‌ها، که تعدادی از آن‌ها اجباری و تعدادی اختیاری هستند، هر کدامشان یک وظیفه را باید انجام بدهند که این وظایف شامل: ذخیره شماره سریال منحصر به فرد برای هر کدام از سیم کارت‌ها، ذخیره اطلاعات در مورد سرویس‌هایی که یک سیم کارت می‌تواند آن‌ها را به اجرا در بیاورد. البته که این قابلیت سیم کارت‌ها هم وابسته به نوعشان است که می‌تواند، sim یا usim باشد که اولی برای نسل دوم سیم کارت‌هاست و دومی برای نسل سوم سیم کارت‌ها (البته تفاوت‌های دیگری هم دارد)، مورد بعدی در نوع سیم



و در مورد هر کدام از کدهای ذکر شده در بالا داریم:

**pin: (Personal Identification Number)** شماره شناسایی شخصی - ۲ پین وجود دارد (پین ۱ و PIN2) یک PIN (شناسه شخصی) یک کد عبور ۴-۸ رقمی است که برای تأیید هویت کاربر به یک سیستم استفاده می شود.

پین تلفن یک دستگاه ذخیره سازی داخل سیم کارت است و همچنین یک اقدام امنیتی برای محافظت از سیم کارت به سرقت رفته است. اگر "عملکرد امنیتی PIN" توسط گوشی شما پشتیبانی می شود و فعال شده است، هر بار که شما تلفن خود را شروع می کنید، باید PIN را برای باز کردن قفل سیم کارت وارد کنید. هنگامی که هر دو پین و فعال هستند، ابتدا باید PIN و سپس رمز عبور قفل صفحه را برای باز کردن قفل گوشی خود را وارد کنید.

پین توسط اپراتور مخابراتی ارائه می شود و می تواند مجدداً تنظیم و اصلاح شود.

نکته: اگر پین بیش از سه بار اشتباه وارد شود، هر دو سیم کارت و تلفن قفل خواهد شد. آنها می توانند با وارد کردن PUK باز شوند.

اگر پین را نمی دانید و یا آن را فراموش کرده اید، برای دستیابی به اپراتور مخابراتی با ما تماس بگیرید (به طور کلی، ۱۲۳۴ یا ۰۰۰۰ پین پیش فرض برای سیم کارت های سیم کارت است، اما فقط برای مرجع، با دقت استفاده کنید).

**Puk:** یک PUK (کلید باز شناسایی شماره شخصی) نیز PUC (Pin Unlock Code) نامیده می شود.

یک puk رشته از هشت اعداد نامنظم است که نمی توانند توسط کاربر تنظیم مجدد یا اصلاح شوند. Puk فقط زمانی تغییر می کند که سیم کارت جایگزین شود.

در نظر داشته باشید که puk برای باز کردن قفل پین استفاده می شود و برخی از PUK ها همراه با سیم کارت هایی هستند که کاربر خریداری می کند.

اگر PUK، ده بار در یک ردیف اشتباه وارد شود، هر دو سیم کارت و تلفن به طور دائم قفل شده و شما باید گواهینامه معتبر خود را به فروشگاه اپراتورهای مخابراتی ببرید تا یک سیم کارت جدید دریافت کنید. بنابراین، هنگامیکه سیم کارت و تلفن با پین قفل می شوند، شما باید به طور همزمان در تماس تلفنی با اپراتور مخابراتی باشید.

پس از تایید ساده برای داده های کاربر، می توانید PIN یا PUK را برای باز کردن قفل سیم کارت و تلفن خود بدست آورید.

ADM هم یک سطح دسترسی است که همان طور که گفته شد فقط اپراتور و سازنده، آن را دارد و این یک کد است که اپراتور با زدن این کد می تواند از این سطح دسترسی استفاده نماید.

همچنین اطلاعاتی دیگری که این فایل ها در اختیار ما می گذارند شامل:

### International Mobile Subscriber Identity or

**IMSI:** عددی ۱۳ تا ۱۵ رقمی می باشد که برای شناسایی اپراتور استفاده میگردد. رقم اول کشور اپراتور و رقم بعدی خود اپراتور را نشان میدهد و اعداد بعدی نشان دهنده اطلاعات و جزئیات اپراتور میباشد این عدد بر روی لاشه سیم کارت قابل مشاهده است.

### Authentication Key or Ki:

کیلیدی بین اپراتور و کاربر سیم کارت میباشد که به اپراتور اجازه میدهد تا کاربر را شناسایی و اطلاعات و هویت وی را احراز نماید. این اطلاعات در حین صدور سیم کارت از کاربر گرفته شده و به این کلید اختصاص می یابد و باید با روش های خاص و از طریق خواننده کارت بازیابی شود.

### Location Area Identity or LAI:

اطلاعاتی درباره موقعیت مکانی، که هر سیم کارت، آخرین اطلاعات موقعیت مکانی بر حسب **BTS** وصل شده را در خود ذخیره میکند. در دیگر بخش ها نیز پیام های اس ام اس و شماره تلفن ها و تماس ها ذخیره میشوند.

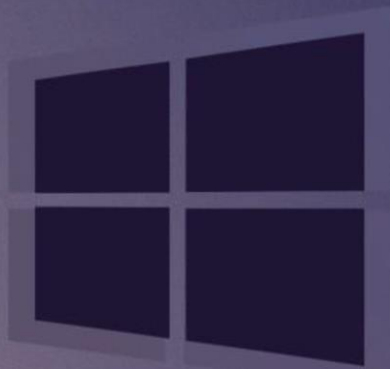
خوب شاید در ذهنتان هزاران سوال ایجاد شده باشد، چرا که این موضوع خیلی آشنا نیست، اما برای آشنایی بیشتر توصیه می کنم در استانداردهای ETSI به دنبال پاسخ سوالاتتان بگردید.

از این سیم کارت ها و اطلاعات داخل آن، می توان استفاده های زیادی کرد؛ از کیف پول های دیجیتال که سیم کارت بتواند اطلاعات را نگه دارد، تا اپلت های **over the air** که بدون دسترسی های مستقیم اپراتور، کارهایی روی سیم کارت به صورت **over the air** انجام می دهد.



## Load نشدن search در ویندوز 10

دهید تا قابلیت Run باز شود. بعد از آن، عبارت services.msc را تایپ کنید تا صفحه Services باز شود. از این جا، Windows Search service را پیدا کنید. سپس، روی آن راست کلیک کنید و عبارت properties را انتخاب کنید. وقتی صفحه Windows Search Properties باز شد، عبارت Startup Type را کلیک کنید و آن را در حالت Automatic قرار دهید. پس از آن، روی OK کلیک کنید تا تغییرات ذخیره شود.



**راه حل سوم:** چنان چه با انجام دو مرحله بالا Search ویندوز شما مجدداً load نشد، با روش ذیل باید درست شود. ابتدا Control Panel را بیاورید. عبارت Indexing Options را جستجو کنید و آن را کلیک کنید تا صفحه‌ای در برابر شما باز شود. روی Troubleshoot search and indexing در زیر همان صفحه کلیک کنید و مراحلش را دنبال کنید. اگر بلافاصله مشکلاتان درست نشد کامپیوتر را reset کنید.

### آیا تعمیری این دفعه برایتان مفید بود؟

تعمیری‌های خود را برای ما به ایمیل مجله پردازش ارسال کنید. تعمیری شماره بعد مجله: [آیا تاج پد لپتاپ شما کار نمی‌کند؟](#)

حدوداً یک ماه بود که search ویندوز بالا نمی‌آمد و من برای پیدا کردن برنامه‌هایی به مشکل برخوردی بودم. خیلی هم برای رفع مشکل تلاش کردم و زمان زیادی صرف رفع مشکل کردم اما درست نشد؛ تا اینکه یک فکری به ذهنم رسید!

خوب؛ حتماً هزاران نفر هم این مشکل را داشته‌اند و از آن هزار نفر، حداقل ۵ نفر، مشکلی را حل کرده‌اند و جواب را با بقیه به اشتراک گذاشته‌اند. پس، در اینترنت گشتم و راه حل را پیدا کردم.

### • search ویندوز 10 با مشکل همراه است؟

Search ویندوز 10 همراه با یک دستیار شخصی به نام "کورتانا" ارائه می‌شود. عامل اصلی اختلال مربوط به تنظیمات کورتانا و لینک شدنش به حساب مایکروسافت است. اما راه حل این مشکل چیست؟؟

### • راه حل رفع مشکل load نشدن search در

#### ویندوز 10 چیست؟

**راه حل اول:** کورتانا را غیر فعال کنید! برای این کار بر روی نوار وظیفه (taskbar) کلیک راست کنید و گزینه TaskManager را انتخاب کنید. مطمئن شوید که در بخش Processes قرار دارید. سپس دنبال Cortana یا search بگردید. روی آن راست کلیک کرده و سپس آن را غیر فعال کنید. مشکل حل می‌شود اما شما نمی‌توانید از دستیار شخصی کورتانا استفاده کنید.

### راه حل دوم: برای این که Search در ویندوز 10

درست عمل کند، باید مطمئن شوید که Windows Search service فعال شده باشد. برای فعالسازی، روی کیبورد خود همزمان دکمه Windows و حرف R را فشار

