۳

پویش

مهرماه ۷۶
۴۴ صفحه
۱۴۴ تومان

نشریه دانشجویان کامپیوتر دانشگاه صنعتی امیرکبیر

به‌نام آفریدگار قلم

فهرست

سرمقاله ۱
استاد-دانشجو: دونگاه ۳
اصفهان از روی شماره ۷
نگاه ۱۰
معرفی نشریات دانشجویی ۱۱
کوتاه ۱۲
جاوا چیست؟ ۱۷
تراشه‌های جاوا ۲۰
تعریفهای پیچیده اشاره‌گرها ۲۴
بهبود کارایی ویندوز۹۵ ..... ۲۶
آینده؛ ذخیره‌وبازیابی‌داده‌ها ۲۹
چندکارگی در یونیکس ۳۳
یک سیستم شیءگرا ۳۶
یادمان ۳۷
نتیجهٔ‌نظرسنجی‌شماره‌دوم ۴۱
مسابقه ۴۱

مسئول پویش:
امین چاروسه

هیأت تحریریه:
علیرضا بذرافشان
علی حاجی‌زاده‌مقدم
محمدرضا خجسته
علیرضا ماندگار

طرحها از:
محسن طریقیان

همکاران:
مهدی رازپوش‌نظری
رویا رحیم‌زاده
مهدی رومی
مصطفی سلمانی
آرش سیف‌هاشمی
مهدی کاظمی
طیبه کمالی
علی‌اصغر کومرشی
مرتضی محمودزاده

و با تشکر از:
استاد گرامی
جناب آقای مهندس شجری
و نیز هیئت رئیسه و
مدیرعامل محترم شرکت
کوارتز کامپیوتر

راههای ارتباط با پویش:
صندوق پستی ۶۱۳ - ۱۳۱۴۵
پست الکترونیکی pooyesh@ce.aku.ac.ir

پویش روی WEB: http://194.165.1.41/pooyesh/

سخن اول

در ابتدای ترم جاری، دکتر محمد مهدی همایونپور به ریاست دانشکدهٔ کامپیوتر برگزیده شدند.

باتوجه به امیدواریهای فراوانی که پس از این انتخاب در دانشکدهٔ ما ایجاد شده، تعدادی از اعضای شورای صنفی دانشجویان و تحریریهٔ پویش طی گفتگویی نظرات ایشان را در مورد مسائل جاری دانشکده و فعالیت‌های دانشجویی جویا شدند.

دکتر همایونپور در این ملاقات صمیمانه، ضمن برشمردن پاره‌ای از مسائل، بر لزوم فعالیت و همکاری مستمر استادان و دانشجویان برای برطرف کردن موانع تاکید کردند و از دانشجویان خواستند تا انتقادات و پیشنهاداتشان را بطور مستقیم یا از طریق نمایندگانشان با ایشان درمیان بگذارند.

ما نیز در این جلسه از ایشان خواستیم تا نظرشان را در مورد مشکلات دانشکده و برنامه‌های آینده، برای دانشجویان بیان کنند. آقای دکتر با وجود کمی وقت و مشغلهٔ فراوان، درخواست ما را اجابت کردند. با تشکر از دکتر همایونپور، سرمقاله این شماره را به پاسخهای ایشان اختصاص می‌دهیم.

● برخی از مشکلات اساسی دانشکده که تا حد زیادی در سطح کل دانشگاه مطرح می‌باشند:

○ عدم وجود فضای آموزشی و پژوهشی کافی

○ عدم حضور استاد و دانشجو بطور تمام وقت در دانشگاه (البته نه بطور عام)

○ کافی نبودن منابع لازم برای آموزش وپژوهش در کتابخانه

○ کافی نبودن منابع محاسباتی پیشرفته

○ چشمگیر نبودن فعالیتهای فوق برنامه برای دانشجویان، هیأت علمی و کارمندان

○ چشمگیر نبودن ارتباط با صنعت در زمینه پژوهش

مشکلات فوق دارای عواقبی است که از آن جمله می‌توان به موارد زیر اشاره نمود:

○ تضعیف آموزش و پژوهش،ناشی از:

۱ ـ ارائه کلاسها در زمانها و مکانهای نامناسب و در وضعیت ناراحت کننده

۲ ـ توقف و یا کندی فعالیتهای علمی-پژوهشی و دلسردی استاد و دانشجو بر اثر کمبود فضای آزمایشگاهی

۳ ـ نداشتن دفتر مستقل برای استادان ، که به دلیل تداخل کاری سبب عدم تمایل استادان برای حضور آنها در دفاترشان می‌شود

○ پایین آمدن روحیه و سطح علمی دانشجو و استاد وکم شدن دلبستگی آنها به محیط کار

○ بالا نرفتن کیفیت ارائه درسی و کندی امور پژوهشی و عدم امکان انجام تحقیقات در حد مرزهای دانش، به دلیل نبود کتابها و منابع علمی جدید در کتابخانه و فراهم نبودن امکان دسترسی راحت به آنها

○ عدم توفیق در استخدام و جذب نیروهای قوی به دلیل نبود امکانات رفاهی جذب‌کننده

○ کندی و دشواری مراحل اولیه استخدام برای هیأت علمی و کارشناس، در دانشگاه

○ از دست دادن استادان قوی به علت عدم امکان استخدام سریع، و عقب افتادن از دانشکده‌های رقیب که با سرعت استادان قوی را استخدام می‌کنند.

○ کندی گزینش دانشجویان دکترا

❁ برنامه‌هایی که به منظور ارتقاء سطح علمی،آموزشی و پژوهشی دانشکده مهندسی کامپیوتر مد نظر می‌باشد :

○ بالا بردن کیفیت علمی-آموزشی دانشکده به کمک استخدام هیأت علمی جدید و ارزیابی مستمر فعالیتهای آموزشی و محتوای دروس تدریس شده

○ بالا بردن سطح کیفی فعالیتهای تحقیقاتی و پژوهشی با

۱ ـ گسترش و تجهیز آزمایشگاههای تحقیقاتی

۲ ـ ارتباط با صنعت و مراکز تحقیقاتی

۳ ـ تشویق پروژه‌های تحقیقاتی مستقل در جهت نیازهای مملکتی و پیشبرد مرزهای دانش، ضمن درگیر کردن دانشجویان در انجام اینگونه پروژه‌ها

۴ ـ انتشار مقالات علمی و تألیف و ترجمه کتاب

○ ارزیابی مستمر کتابخانه دانشکده و سعی در بهبود کیفی و کمی آن با خرید منابع و کتب و مجلات جدید مورد نیاز در امر آموزش و پژوهش و فراهم آوردن امکان دسترسی راحت به منابع مذکور

○ ارزیابی مستمر آزمایشگاههای دانشکده و تلاش در بهبود کیفیت آنها

○ افزایش فضای دانشکده وتأمین مکانهای مناسب و کافی برای بر قراری کلاسها، آزمایشگاهها، مکانهای پژوهشی، کتابخانه، سالن مطالعه، دفاتر اساتید و امور اداری

○ ارائه دوره‌های تخصصی-کاربردی کوتاه مدت و دوره‌های معادل، در راستای رفع نیازهای صنایع و ارگانهای مختلف، که از محل درآمدهای حاصل از آن می‌توان با پشتوانه مالی قویتری به گسترش امکانات دانشکده و تقویت آن از جنبه‌های مختلف اقدام نمود.

○ جذاب نمودن هر چه بیشتر محیط دانشکده برای دانشجویان و اساتید، و تشویق آنان به حضور بیشتر و مفیدتر با تأمین موارد مذکور در بندهای فوق

استاد . . . . دانشجو

اشاره

آنچه در پی می‌آید، نوشتهٔ یکی از دانشجویان دانشکدهٔ ما است که در اواخر ترم گذشته در تابلوی شورای صنفی نصب شده بود. نصب این مطلب در تابلو، موجب شد باب گفتگو بر سر موضوع "روابط استاد و دانشجو" گشوده شود. ولی پایان ترم تحصیلی و تعطیلی دانشگاه مانع از ادامهٔ این بحث شد.

با توجه به اهمیتی که پویش برای مقولهٔ ارتباط گرم و صمیمانه توام با احترام متقابل میان دانشجویان و اساتید قائل است، با درج این مطلب و پاسخ یکی دیگر از دانشجویان به آن، به استقبال مطالب آتی اساتید و دانشجویان در این زمینه می‌رود.

لازم به‌ذکر است مطالب مطرح‌شده لزوماً نظر پویش نیست و بدون هرگونه ویرایش یا دخل و تصرفی منتشر می‌شود.

نگاه اول

استاد گرامی

"هر یک از ما روزی با معیارمان نسبت به زندگی
مورد قضاوت قرار خواهیم گرفت...
نه با معیار زندگی کردن؛ با میزان بخشیدن...
نه با میزان ثروت؛ فقط با میزان خوب بودنمان،
نه با بزرگی ظاهری."

"ویلیام آرتور وارد"

استاد گرامی ...

ما دانشجویان انتظار زیادی از شما نداریم. از اینکه رفتاری مطابق شأن یک دانشجو با ما داشته باشید صرفنظر کردیم. حداقل رفتاری آنگونه که شایستهٔ برخورد با یک انسان است، با ما داشته باشید!!

این قلم را شخصی در دست گرفته که زمانی نه چندان دور، آن هنگام که پشت نیمکت های دبیرستان بهترین دوران عمر خود را سپری می‌کرد، آرزوی ورود به دانشگاه را داشت و از بهترین ساعات عمر خود مایه می‌گذاشت تا شاید روزی بتواند به این آرزو برسد

و اما اکنون...

این تراوشات ذهنی شخصی است که همان‌روزی که نتیجهٔ کنکور در روزنامه اعلام شد و نام خود را در بین قبول شدگان دانشکدهٔ کامپیوتر دانشگاه امیرکبیر یافت، چنان شور و شوقی به وی دست داد که در پوست خود نمی‌گنجید و برای آمدن به دانشگاه، این به اصطلاح بهشت شداد لحظه شماری می‌کرد...

راستی یادتان هست که از بس اسامی قبول شدگان ریز نوشته شده بود به زحمت می‌توانستیم نام خود را بیابیم. اما ای کاش آن لحظه از این موضوع غافل نبودیم که نام کوچک هر دانشجو در روزنامهٔ اعلام نتایج، روزی در دانشگاه چنان کوچک و کوچکتر می‌شود و هر کس از راه می‌رسد او را خوار و تو سری خور می‌کند. از استادش گرفته تا کارمند و کارگر دانشگاه همه وی را در زیر پاهای خود له خواهند کرد.

و اما اکنون این مطلب را دانشجوی بیچاره‌ای می‌نگارد که با ذهنی آشفته و روحی خسته از هر چه دانشجو، دانشگاه و درس است، نفرتی عجیب پیدا کرده است...

روزی با این آرزو وارد دانشگاه شدم تا درس خدمت به اجتماع و مردم کشورم را فراگیرم.

خواستم درس روابط اجتماعی صحیح را بیاموزم؛ اما ...

دیدم که استادم حتی حاضر نیست جواب سلام دانشجویانش را بدهد؛ دیدم که استادم هر بار به بهانه ای از زیر بار جواب به سؤالات دانشجویان شانه خالی می‌کند؛ دیدم که آقای دکتر X همانگونه که روی صندلی، رو به کامپیوترش و پشت به در اتاقش نشسته بود، جواب دانشجویی که دم در اتاقش ایستاده را می‌دهد.

از این عدم احترام متقابل وی به دانشجو که بگذریم؛ تک تک کلمات جوابهایش پر از گوشه و کنایه است به‌گونه ای که تا می‌تواند دانشجو را سبک و سرافکنده‌از همان راهی که آمده بود، برگرداند؛

آقای Y بارهاو بارها دیدم که وقتی دانشجو در اتاقش ایستاده و با او صحبت می‌کند، بدون توجه به صحبتهای دانشجو از اتاق بیرون رفته و با اشارهٔ دست به دانشجو می‌گوید که "فعلاً کار دارم؛ وقت ندارم؛ بیا بیرون می‌خواهم در اتاقم را ببندم و بروم"!...

به دانشگاه آمدم تا درس صداقت، راستی و درستی فراگیرم؛ اما...

دیدم که استادم حتی این حق را به دانشجویش نمی‌دهد که نمره‌اش رابداند. برای اطلاع از وضعیت نمره‌ات که به اتاقش می‌روی

می‌گوید: "من نمره‌ای به شما نمی‌گویم؛ فقط می‌گویم که آیا درس را پاس کرده‌ای یاخیر"! از حق اعتراض به نمره که بگذریم؛ آیا این حق مسلم یک دانشجو نیست که بداند چه نمره‌ای‌گرفته ؟!!

دیدم که استادم چنان نمراتی به نتیجهٔ زحمت دانشجویان می‌دهد، گویی که‌اصلاً استاد این کلاس نبوده... حقیقت این است که‌تاکنون کسی نتوانسته برای نمره دادن این استاد فرمول خاصی کشف کند. یکی که انتظار نمرهٔ‌پایینی دارد، ناگهان بالاترین نمرهٔ‌کلاس می‌شود و برعکس...

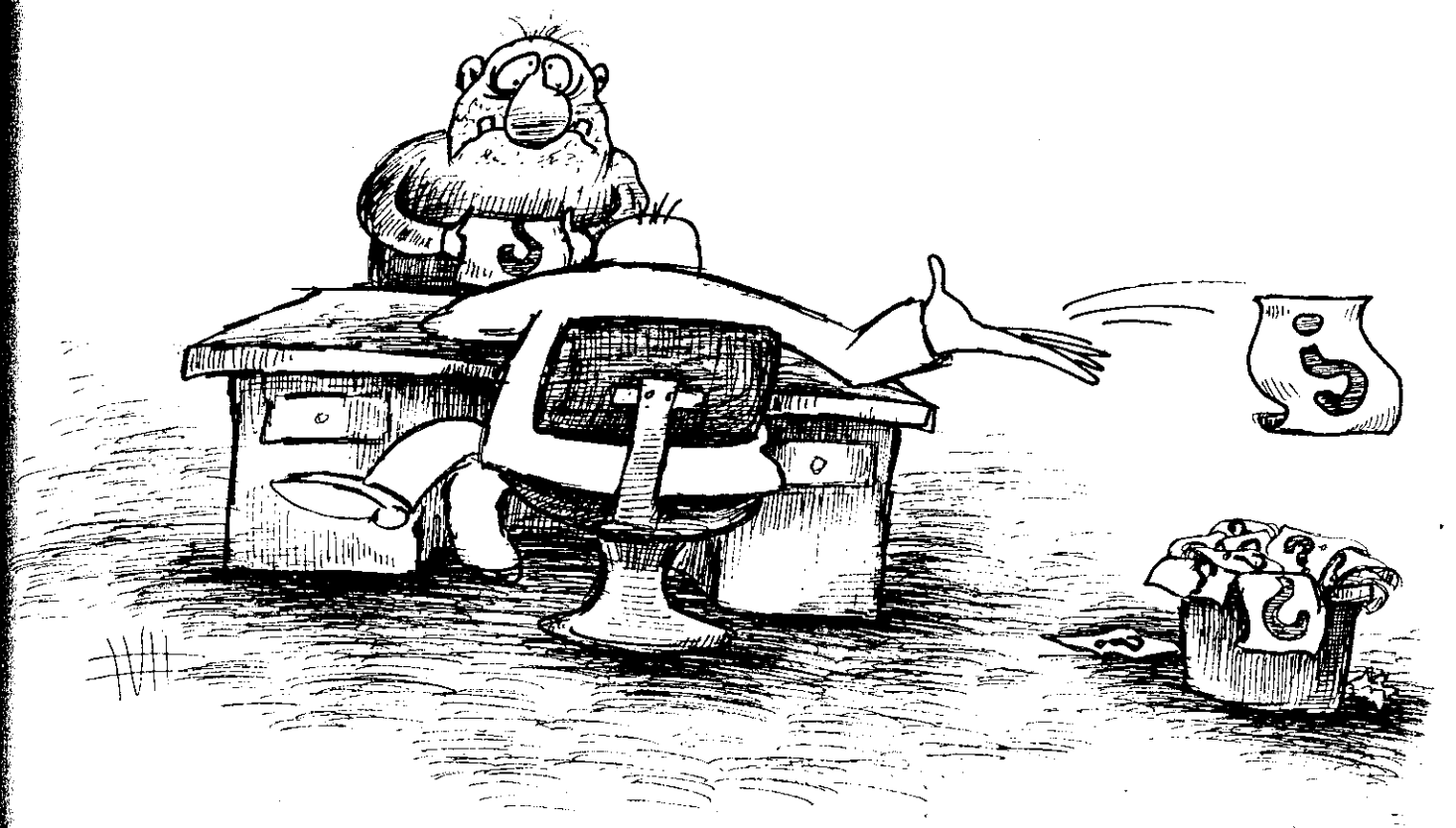
آمدم دانشگاه تادرس احساس مسؤلیت و تعهد فراگیرم؛ اما...

استادم به من آموخت که می‌توان جلسات زیادی از یک کلاس‌درس راهدر داد. ۲۰دقیقه‌ای‌از اول کلاس دیر آمد، نیم ساعتی نیز زودتر کلاس را تعطیل کرد. آخر ترم با یک کلاس ۳-۴ ساعته جبرانی تمام‌مطالب درس راگفت و کل امتحان‌پایان ترم رااز همان کلاس جبرانی انجام داد.

استادم به من آموخت که می‌توان شب امتحان و یا شاید نیم ساعتی مانده به امتحان چنان سؤالات اشتباه و سردرگمی طرح کردو بارها سر جلسهٔ‌امتحان آنها را اصلاح کرد و آخر هم مشخص شود که مساله از ریشه و بنیاد اشتباه بوده است. یا می‌توان‌برای راحتی بیشتر، تعدادی تست که چندین سال متوالی به عنوان سؤال امتحانی داده شده است را به دانشجوی بیچاره قالب کرد و حتی زحمت این را به خود نداد که تست های‌غلط راکه هر سال تکرار می‌شود، تصحیح نماید.

استاد به من آموخت که میتوان هر که زیر دست انسان است؛ مثلاً دانشجو را هیچ حساب کردو ارزشی برایش قائل نشد. دانشجوی مفت‌خور دانشگاه دولتی چه ارزشی می‌تواند داشته باشد که من استاد، بخواهم جزوه‌ای را که سر کلاس تدریس می‌کنم، کمی اصلاح کنم، بجای مطالب کهنه عهد بوق موجود در جزوه مطلب جدیدی جایگزین کنم، و یا حتی قبل از آمدن سر کلاس برای تدریس، مطلب مورد تدریس را مطالعه کرده‌باشم تا به سؤالات دانشجویان بتوانم در حد قانع کردن آنهاجواب دهم.

استاد گرامی: ترا می‌ستایم به‌خاطر این همه آموختنی‌هایی که به من یاد دادی !!!



می‌گفت: شاعر! آخر زمانی روح تو وسعت بی‌پایان داشت... بر وسعت روح تو چه گذشت!

فریادکردم: خاموش! بامن دیگر از وسعت روح حرف مزن!...

همه، هر چه تنگ‌نظری دیدم، در وسعت روح خودکم کردم!...

آنقدر کم کردم، تا وسعت روحم پر شد...

پر شد از یک مشت تنگ‌نظری‌های کم شده!

نگاه دوم

دوست ... من

دوست همکلاسی من

از اینکه می‌بینم اینقدر ذلیل و خوار شده‌ای (آنطور که خودت میگی) بسیار متاسفم. ولی الان من نمی‌خواهم با تو همدردی کنم، برای تو و دیگرانی که مثل تو هستند تأسف بخورم. قصد هم ندارم از اساتیدمان دفاع کنم. آنچه باعث شد دست به قلم شوم مشکلی است که گریبانگیر بیشتر ماست و در نوشتهٔ تو هم موج می‌زند. دردی که یک جامعه‌را و یک ملت را فلج می‌کند و آنها را بی‌ناموس و خسته وذلیل می‌سازد. و این‌درد اگر درمان نشود و اگر بر ما غالب شود، نقطهٔ پایانی خواهد بود بر تاریخ ما و فرهنگ ما و اصلاً همه چیز ما. نمی‌دانم اسم این درد چیست. شاید بشود گفت "درد انتظارات بزرگ و همتهای کوچک". درد کسی که خوابهای قشنگی می‌بیند ولی در بیداری از جایش تکان نمی‌خورد تا اینکه دوباره به خواب رودو...

دوست لهشدهٔ من

نوشته بودی که‌هر کس از راه می‌آید، به گونه ای تو را خوار و توسری‌خور می‌خواهد. بر فرض که چنین باشد. تو چرا می‌نشینی تا دیگران توی سرت بزنند. چرا هیچ کاری نمی‌کنی؟ چرا از جمعی که تو (و بسیاری مثل تو) تشکیل‌دهنده‌اش هستید جز بیحالی و سستی چیزی به چشم نمی‌آید؟ وقتی تو از پیش تسلیم شده و افتاده‌ای، چطور انتظار داری دیگران به حسابت بیاورند و احترامت رانگه دارند؟

تو که از زمین و زمان طلبکاری، تابحال برای مردم و اجتماع و کشورت چه کرده‌ای؟

بگذار تو را تجسم کنم: دانشجویی که هر روز صبح دیر بیدار می‌شود و دیر به کلاس می‌رسد. سر کلاس یا چرت می‌زند، یا با چشمهای مات و گرد شده به استاد و تخته نگاه می‌کند بی‌آنکه بفهمد موضوع درس چیست. یک ساعت از شروع کلاس نگذشته، صدایش بلند می‌شود که: "استاد خسته نباشید" و بعد از کلاس باچند نفر دیگر روی یک لبهٔ‌صندلی می‌نشیند و پشت سر استاد جوک می‌گوید، یا سریال دیشب و فوتبال پریروز را برای دوستش تعریف می‌کند. ظهر در سلف، سر غذا، اولین حرفی که از دهانش بیرون می‌آید بد گفتن از غذاست و بعد هم گفتن حرفهایی از قبیل اینکه امروز تعداد گربه های دانشگاه کم شده و چمن ها را زده‌اند و آه و ناله از بلاهایی که این غذاها سرش آورده‌اند و... کلاسهای عصر هم مثل کلاسهای صبح می‌گذرند و بعد، از عصر تا شب یا در پارک نشسته، یا در خوابگاه چرت می‌زند، یاپای تلویزیون لم می‌دهد، و یا اینکه با چند نفر شبیه خودش گوشه‌ای جمع می‌شوند و پشت سر مسؤلین جامعه و دانشگاه و اساتید و دوستان دیگرش (که غایبند) بدگویی می‌کنند. اگر فردی سیاسی و اجتماعی (!) باشد به یکی از روزنامه ها هم نگاهی می‌اندازد و اگر خیلی روشنفکر باشد، یکی دو مجله را هم ورق‌می‌زند. اگر فردی رومانتیک باشد، گوشه‌ای کز می‌کند و شعرهای یأس آلود می‌سراید. اگر هم کسی باشد که در مدرسه درسخوان به حساب می‌آمده به جزوه‌هایش هم نگاهی می‌اندازد (که این یکی صحنهٔ‌نادری است) و... همین! (و البته باید به آن چیزهای دیگری هم اضافه کرد. ولی چون ترویج منکر محسوب می‌شود قابل نوشتن نیست!) وآنوقت چنین کسی، از خدمت به اجتماع و مردم وکشور و از روابط صحیح اجتماعی و از صداقت و راستی ودرستی و احساس مسؤلیت و تعهد و... دم می‌زند.

و تازه خطاب به چه کسانی؟

دوست تو سری خوردهٔمن

در مورد استادهایت چه می‌دانی؟

آیا میدانی دکتر X مدرکش را از چه دانشگاهی گرفته و در کدام دانشگاه آمریکا یا اروپا به تدریس مشغول بوده؟ و فکر می‌کنی این آدم (به قول تو) بی مسؤلیت را کدام درد و کدام نیاز به وطنش برگردانده؟ و کدام انگیزه او را در اینجا (که دانشجویانش اینطور قدرشناسند ومسؤلان بالاترش، طوری دیگر) نگه داشته؟

هیچ می‌دانی تخصص دکتر Y در ایران نظیر ندارد؟ و می‌دانی که دانشگاههای خارج در آرزوی جذب امثال اویند و شرکتهای داخلی هم برای یک ساعت کارش، حاضرند بیش از حقوق یک ماه استادی پول بدهند؟

نمی‌دانم تا بحال با آن استاد برخورد نکرده‌ای که موقع سلام به دانشجویان، دستش را روی سینه‌اش می‌گذارد و خم می‌شود، یا آن استادی که همیشه در سلام پیشدستی می‌کند، یا آن‌یکی که در جمع دانشجویانش مانند یکی از آنهاست و یا آنکه چنان اوراق امتحانی را با دقت و کلمه به کلمه تصحیح می‌کند که آدم مات می‌ماندو...

گفتم که قصد دفاع از کسی را ندارم. آنچه نوشتم هم به قصد انکار برخی نقایص و نارسائیهاو کم کاریها نیست. اینها را گفتم تا ببینی چقدر بی‌انصافی کردن و یک طرفه به قاضی رفتن، زشت و در عین حال آسان است. و اینکه با این قبیل حرفها مشکلی حل نمی‌شود که هیچ، امید حل مشکل هم از بین می‌رود. گفتم تا شاید کمی به خود بیایی و دست از غرزدن و ایرادگرفتن و ناله و نفرین کردن برداری، که با ناله و نفرین، حتی دل آدم هم خنک نمی‌شود.

دوست کوچک شدهٔمن

تو که از هر چه دانشجو و دانشگاه و درس است (و لابد از خودت هم) نفرتی عجیب پیدا کرده‌ای، تو که همه چیز را یأس آور و خاکستری می‌بینی، تو که از عدم‌وجود روابط صحیح اجتماعی و گم شدن صداقت و راستی و درستی و فراموش گردیدن احساس مسؤلیت و از یاد رفتن تعهد رنج می‌بری، به من بگو، تاکنون چه کرده‌ای؟ چقدر تلاش‌کرده‌ای صادق باشی؟ نسبت به چه چیزها و چه کسانی احساس مسؤلیت می‌کنی و این احساس (اگر وجود دارد) چطور تجلی پیدا کرده؟ اصلاًبگو بدانم، به نظر خودت وجود تو چه تأثیری بر جامعه و اطرافیانت داشته؟ و اگر تو را در نظر نگیریم چه لطمه ای برپیکر اجتماع می‌خورد؟ کجای کار لنگ می‌ماند؟ چه اشکالی پیش می‌آید؟ و چقدر درد آور است اگر پاسخ همه این سؤالات "هیچ" باشد.

دوست من، دانشجو:

مهمترین مشکل من و تو، درد بزرگ ما و بلای عظیم جامعهٔ ما،"بی عملی" است.

نگاه: در حاشیه افتتاح ساختمان ابوریحان!

برای درج در این شمارهٔ پویش، گفتگویی با آقای نادر ابراهیمی، نویسنده معاصر، صورت گرفته بود که به علت کمبود جا موفق به استفاده از آن نشدیم.
به خواست خدا، این مطلب، در شماره آینده چاپ می شود.

موضوعات مربوط به رشتهٔ کامپیوتر چندتاست؟

آیا یک نفر می‌تواند در مورد همهٔ این موضوعات مطالعه کند؟

پاسخی که شورای صنفی به آن رسید این بود:«نه ،نمی‌شود». به همین جهت تصمیم گرفتیم اقدام به برگزاری کلاسهای کوتاه مدت آموزشی کنیم تا در آنها، هر هفته یکی از دانشجویان(که در مورد موضوع خاصی مطالعه و کار کرده) سایر دانشجویان را با آن موضوع آشنا کند. اولین قدم در این راه، اواخر ترم گذشته برداشته شد (کلاسهای آموزش اسمبل کردن کامپیوتر) در ترم جاری هم تابه‌حال کلاسهای اسمبل کردن و دیباگ کردن برنامه ها برگزار شده است.

شورای صنفی از همه کسانی که در مورد هر یک از موضوعاتِ علمِ کامپیوتر مطالعه یا کار کرده‌اند و مایلند که دانسته هایشان را به سایرین هم منتقل کنند، برای تشکیل این کلاسها دعوت به همکاری می‌کند.

شورای صنفی-آموزشی دانشجویان
دانشکده مهندسی کامپیوتر

اصفهان از روی شماره

۱ ◘

-قربون همهٔ شما... مرسی،مرسی... آخری‌ها ببخشن صدامون بهشون نمی‌رسه... حسنِ خوشه‌چین، خوشه‌چینِ حسن.

- سبز سبزم ریشه دارم، در زمستان هم بهارم... سبز سبزی ریشه داری، در زمستان هم بهاری... سبز سبزند ریشه دارند، در زمستان هم بهارند...

-حسن!... به تو می‌اندیشم... حسن یک، حسن دو، حسن سه...حسن ده

حسن ده، حسن نه، حسن هشت...حسن یک

حسن حسن حسن (دومرتبه)

- خدایا، خداوندا، دانشکده ما را بزرگتر بفرما. آمین.

- خدایا، خداوندا، به دانشکده ما pc عنایت بفرما. آمین.

- خدایا، خداوندا، اساتید ما را ملایمتر بفرما. آمین.

- خدایا، خداوندا، کتابدار کتابخانه را خوش‌اخلاق‌تر بفرما. آمین.

۲ ◘

بعد از ظهر یک روز تیرماه است. عدّه‌ای جلوی درب رشت دانشگاه میرزاتقی خان جمع‌شده‌اند. اتوبوسی منتظر، گرما بیداد می‌کند و سردی بین آن جمع را می‌پوشاند. این عدّه چه‌کاره‌اند؟ بیکاره‌اند، باکاره‌اند، سرکاند...؟ برای چه سوار اتوبوس می‌شوند؟ سفرکنند؟ سفرکنند که چه شود؟

گروه تدارکات اردو سخت درحال بسته‌بندی وسایل اردو است. درآن هوای گرم واقعاً کار سختی است. گروه تا اینجای کار مراحل مشکلی را پشت‌سر نهاده مثلاً تهیهٔ بودجه و گرفتن اتوبوس که انجامشان یعنی از هفت‌خوان رستم گذشتن و در خوان هشتم گرفتار آمدن:

○ساعت ۸ صبح، یکروز مانده به اردو:

مسؤول مالی دانشکده می‌گوید «اگر معاون دانشجویی بودجه بدهد من هم میدم». معاون دانشجویی می‌گوید «اگر بتونی از مسؤول دانشکده بگیری منم بهت می‌دم». بیچاره مسؤول اردو؛ از معاونت دانشجویی به امور مالی، از امور مالی دانشکده به معاونت دانشجویی:

مسؤول شماره یک: «اصغر اصغر، اکبر»

مسؤول شماره دو: «اصغر به گوشم»

مسؤول شماره یک: «اصغرجان بهش بودجه دادی؟»

مسؤول شماره دو: «نه، تو چی؟»

مسؤول شماره یک: «مرد حسابی چه حرفها می‌زنی، تو مرام ما دفعه اول و دوم کار راه نمی‌افته. هَه هَه هَه... بذار یه‌خورده بدَوه»

○ساعتی بعد:

مسؤول شماره یک: «اصغر اصغر، اکبر»

مسؤول شماره دو: «اکبر جان به‌گوشم»

مسؤول شماره یک: «مسؤول اردو کجاست؟»

مسؤول شماره دو: «درمیانه میدان، سرگردان»

مسؤول شماره یک: «عجب سعی وتلاشی. راستی راستی باورش شده که می‌خواد اردو ببره»

مسؤول شماره دو: «منم می‌خوام باهاشون برم نظرت چیه؟ هم فاله هم تماشا. بریم ببینیم تو این اردوهای کامپیوتر چه خبره که اینقدر سر وصدا می‌کنه!... چند تا از اساتید هم میان»

مسؤول شماره یک: «حال می‌کنی‌ها»

○دفتر معاونت دانشجویی:

مسؤول اردو: «آقا بودجه بده»

مسؤول شماره یک: «نمی‌دم. برو اول از اصغر بگیر»

○دفتر امورمالی:

مسؤول اردو: «آقا بودجه بده»

مسؤول شماره دو: «نمی‌دم. برو اول از اکبر بگیر»

مسؤول اردو: «ای هوار، ای بیداد، اگر بودجه ندی همهٔ استادا رو با خود شما می‌برم هتل چمن، بجای مرغ آبگوشت، بجای نوشابه آب‌زرشک...»

مسؤول شماره دو: «ببین پسرم چرا عصبانی می‌شی؟ مایه میخوای از اول بگو. چقدر میخوای؟ ۶۰تا بسه؟ بیا اینم ۶۰ هزارتومن، برو پیش اکبر بگو هرچقدر می‌خوای بهت بده... راستی آقای حسن، حالا هتل چند ستاره هست؟»...(مسؤول اردو خارج می‌شود)

- اکبر اکبر، اصغر. آقا بده، بده بده پول بده، مسأله تقابل مرغ و آب‌گوشته، هرچی خواست بده.

مسؤول اردو بالاخره در دقیقه نود پول را می‌گیرد. پایان پیش درآمد سفر.

○اتوبوس در آستانه حرکت است:

صدا: «لطفاًخواهران عقب اتوبوس، برادران جلوی اتوبوس» یاهاهای های‌های‌های. چه نوای روح بخشی؛ اصولا سنت چیز خوبی‌است... یاهاهای های‌های‌های نمی‌دونم چرا سرم داره درد می‌گیره؟!... یاهاهای های‌های‌های ...آقا اگه می‌شه یکی نوار رو عوض کنه!

- نه جونم نمیشه.

- آقا جون مادرت عوضش کن سرمون داره ..

- گفتم که نمیشه!اِهِه!

- مگه موزیک خالی اشکال داره؟

سکوت سرشار از ناگفته‌هاست! موسیقی کلاسیک؟ پناه‌بر خدا! بچه به تو چه که موسیقی اونور آب به کجا رسیده؟ اصلاً چرا کلاسیک اجنبیا رو گوش می‌کنی، مگه غربزده‌ای؟ لیبرال! کلاسیک خودمون که بهتره: گل از باغ می‌روید- می‌گذرد رهگذر از کوچه‌ها- آی ای نسیم مارا به کوچه‌تون ببر. اصلاً لازم نکرده موسیقی گوش بدی.

۳

اِ ساندیسها آمد، کیک‌هام پشت سرش. پذیرایی: کیک هست، ساندیس،هست، مهربانی هست...

۴

زمزمه بچه‌ها زیر لب: «الا ای آهوی وحش کجایی؟ آبی دریا قدغن، تو قدغن، من قدغن...»

۵

۷۵ی‌ها در اتوبوس غریبند. ۷۳ای‌ها و ۷۴ی‌هابا هم حرف‌می‌زنند و صحبت می‌کنند...مسؤول اردو: «آقایان ۷۵ی شلوغ نکنید سروصدا زیاده!...شما ۵نفر بفرمایید جلو بنشینید»

مسؤول اردو جای تعدادی از ۷۴ی ها را عوض می‌کند، ما که نفهمیدیم چرا جای آنها عوض‌شد در واقع بعدها فهمیدیم هیچ کس حتی مسؤول اردو نیز نفهمیده چرا جای آنها عوض شد.

۶

یک استراحتگاه کویری.

مسؤول اردو: «آقایان ۷۵ی تا ۵ دقیقه دیگه اینجا باشین... آقایان و خانومها از این اطراف دورنشن گم میشن»

۷

هبوط در کویر. مجدداً به را افتادیم. -اسم من شهرامه -اسم من علیرضاست -اسم منم آرشه -منم ۰۰۷ -راستی خودمم آق‌بابام.

۸

بالاخره نیمی از جهان رؤیت‌شد. گفتیم چی باشه دیگه. اتوبوس به انتظار گروه راهنما توقف‌کرد. مسؤول اردو: آقایون ۷۵ی از اینجا جلوتر نرن... خانومها و آقایون جایی نرن، اتوبوس می‌ره گم‌میشن. راهنماها آمدند. این دونفر نیز پُر زحمت کشیده‌بودند. از دوسه روز قبل جهت تهیه خوابگاه و هماهنگی با فولاد و ذوب‌آهن و غیره به اصفهان آمده بودند تاجایی که $\frac{۴}{۱۷}$ شان به $\frac{۱۷}{۴}$ تبدیل‌شده‌بود.

۹

به یه جایی رسیدیم که بهش می‌گفتن خوابگاه. آقایون اینور خانومها اونور. بعداز شام به نظر می‌رسید دیگر درمحوطه خوابگاه گم نمی‌شویم پس چرخیدن آزاد. خیلی چرخیدیم، آنقدر که سرمان گیج رفت و مجبور شدیم بخوابیم... صبح که از خواب بلندشدیم چی‌شد؟ ها پس Listen ،

آقای دکتر با تیپ ورزشی روی موکت خوابیده‌بودند.

-بچه ساکت!اگه بیداربشه ساختمون افتادی‌ها...
مهندس دست‌پاک (جایش سبز)، مهندس شجری و آقای عبدی تیز در اردو حضور داشتند.

۱۰

سکانس بعدی دیدار از نقاط دیدنی نیمی از جهان بود: چهل ستون، عالی‌قاپو، موزهٔ ملی، هشت بهشت و بازار امام. درچهل ستون n نفر دنبال یک خانوم بودند! (خانم راهنما)

۱۱

قراره بچه‌های دانشکده کامپیوتر از فولاد مبارکه و ذوب‌آهن اصفهان بازدید علمی-تفریحی کنند... یاهاهای. وای بازم توفیق اجباری... جلوی اتوبوس، مهندس دست‌پاک (هی روزگار) و دکتر صادقیان نشسته‌اند. به‌سمت فولاد مبارکه.

○داخل مجتمع:
راهنما که از قضا مهندس کامپیوتر است برایمان کلی ناز و عشوه می‌کند.
کارگاه اول؛ کارگاه دوم؛ کارگاه سوم... خلاصه با اتوبوس یک دور شمسی قمری زدیم. خدا می‌داند کسی‌چیزی فهمید یا نه. اتوبوس به‌سلف کارخانه رسید؛ جذاب‌ترین بخش بازدید (ناهار جوجه‌کباب). بخش علمی-رایانه‌ای این بازدید مربوط می‌شد به کامپیوتری که بنابه گفتهٔ راهنما وظیفه‌اش تنظیم ساعت ورود و خروج کارکنان بود.

۱۲

بعد از ظهر قرار بود به ذوب آهن برویم. فولادمبارکه آنقدر مفید بود که نیمی از بچه‌ها ترجیح دادند در خوابگاه بمانند.
خُب حالا می‌ماند قسمت تفریحی ماجرا. چه‌کنیم؟ ها! قرارشد شام را در سی‌وسه پل بخوریم. این هم قسمت تفریحی. خوب مگه از یک سفر علمی-تفریحی چه‌می‌خواهیم؟ واقعاً که بعضی‌ها چه توقع‌هایی دارند! اصلاً چه معنی داره تو سفر آدم با انسانهای مختلفی آشنا بشه؟ چه معنی داره کسی با کسی دوست بشه؟

۱۳

ساعت از دو و نیم شب گذشته، جلسهٔ دادگاه کاملاً رسمی است.
شاکیان: مسؤولین اردو. مجرم: Xهفتاد و چهاری و دوستانش.
وکیل مدافع: Xهفتاد و چهاری و دوستانش. شاهد: یک اتوبوس آدم.

○بیانیهٔ وکیل مدافع:
عده‌ای از برادران در میانهٔ اتوبوس نشسته‌اند. با خود صحبت می‌کنند. زیر لب زمزمه می‌کنند. می‌گویند و می‌خندند. آیا جرم است؟ مسؤول اردو جای آنها را عوض می‌کند. مجرم شاکی است! رفع اتهام؛ روبوسی؛ والسلام... چندتا لکهٔ‌سیاه کوچولو روی صورت بعضی از مسئولین اردو پیدا شده بود که با ماچیدن صورت همدیگر برطرف شد.
هوای اردو و آسمان رو به روشن شدن می‌رفت.

۱۴

دکتر صادقیان، مهندس دست‌پاک (هی؛ چه سود نیکی را) و مهندس شجری در شب آخر از مسئولین اردو شدیداً تشکر کرده کارشان را بس بزرگ معرفی می‌کنند. بجابود. منظورم اینه که: بابا ای‌والله، دمتون گرم.

○آخرین ناهار:
به میمنت خوش گذشتن در این اردو قرار بر آن شد که یک‌آقایی را سر و ته به دار بیاویزند که آویختند. حالا مدام اینور اونورو نگاه می‌کنند که یک بزی، قوچی چیزی از آسمون برسه. ای خدا پس چرا هیچی نمی‌فرستی، یارو مُرد. بعد از آن تقریباً تمامی بچه‌ها به‌زاینده‌رود پرتابیده شدند... مسؤول اردو: «از اینکه به شما می‌گفتم اینور اونور نرید گم می‌شید، دلگیر نباشید»

○چند دقیقه بعد:
مسؤول اردو: «آقایون هفتاد و پنجی! از اینجا دورتر نرید گم می‌شید.»

○سکانس آخر:
آشنایی بچه‌ها خیلی بیشتر شده. تقریباً همه همدیگر را می‌شناسند و با هم اختلاط می‌کنند. از گز اصفهان و نقاشیهای سانسورشده عالیقاپو گرفته تا اساتید مظلوم دانشگاه(ای دل غافل)... نوار ترکی از ضبط اتوبوس پخش می‌شود... آسون نشو ای همسفر (چی شد؟) هوا رو به تاریکی است. عده‌ای روی ردیف‌های جلوی اتوبوس سرهایشان را درهم کرده درحال پچ‌پچ‌اند. کم‌کم صداها بلند و بلندتر می‌شوند.
-قربون همهٔ شما... مرسی،مرسی... آخری‌ها ببخشن صدامون بهشون نمی‌رسه... حسنِ خوشه‌چین، خوشه‌چینِ حسن.
- سبز سبزم ریشه دارم، در زمستان هم بهارم... سبز سبزی ریشه داری، در زمستان هم بهاری... سبز سبزند ریشه دارند، در زمستان هم بهارند...
-حسن!... به تو می‌اندیشم... حسن یک، حسن دو، حسن سه...حسن ده
حسن ده، حسن نه، حسن هشت...حسن یک
حسن حسن حسن حسن (دومرتبه)
- خدایا، خداوندا، دانشکده کامپیوتر را بزرگتر بفرما. آمین.
- خدایا، خداوندا، به دانشکده کامپیوتر pc عنایت بفرما. آمین.
- خدایا، خداوندا، اساتید دانشکده را ملایمتر بفرما. آمین.
- خدایا، خداوندا، کتابدار کتابخانه دانشکده را خوش‌اخلاق‌تر بفرما. آمین.

۱۵

اتوبوسی به‌نام سردی، بی‌تفاوتی، سوءتفاهم، تفاهم، دوستی، محبت، دوست‌شدن، دوست‌داشتن، سلام، به‌اسم صدا کردن (با قید شروط) و یک عمر آشنایی پایدار.

علی‌اصغر کوهردهی

پویش ۳/ صفحهٔ۹

نگاه: سرزمین خورشید! (Sun land)

معرفی نشریات دانشجویی

احتمالاً دانشکدهٔ کامپیوتر دانشگاه صنعتی امیرکبیر همهٔ دنیانیست. اگر این حرف را بپذیریم، باید این احتمال را بدهیم که پویش هم تنها نشریه دانشجویی دنیا نباشد. اگر احتمالات فوق درست باشند، بد نیست که این نشریات دانشجویی از حال و روز هم باخبر بوده با هم آشنا شوند که پیمودن این مسیر مشترک با یک آشنا بسیار آسان‌تر و امیدبخش‌تر است. با این فکر شروع‌کردیم به جستجو. کار زیاد سختی نبود. با یکبار چرخاندن نگاه در دانشگاه خودمان اقلاً نیم‌دوجین نشریهٔ دانشجویی به‌چشممان خورد:
مصاف (نشریهٔ دانشجویان صنایع)، تلنگر (مکانیک)، بسپار (پلیمر)، حرکت (شیمی)، بلور (معدن)، پیک‌هوافضا (هوافضا)، سرا (عمران) و.... آخرین آنها هم جولا (این نوزاد خوش‌قدم دانشکدهٔ نساجی).
ولی بعد فکر کردیم همهٔ اینها که کم‌وبیش برایمان آشناهستند، پس سرمان را بالاتر گرفتیم. نگاهمان رفت تا.....

دانشگاه تهران
دانشکدهٔ داروسازی
نشریهٔ دانشجویان داروسازی:

موج

این موجی‌است که تابحال پنج‌بار به ساحل رسیده. اولین‌بار در آذر۷۳ و آخرین بارهم در همین مهرماه۷۶ (بعلاوهٔ یکی دو ویژه‌نامه؛ از جمله ویژه‌نامهٔ نوزدهمین سالگرد هجرت دکتر شریعتی).
این شمارهٔ موج با گزارشی مفصل شروع می‌شود به‌نام:
"چرا دانشجویان اغلب انگیزه ندارند؟"
در ابتدای این گزارش می‌خوانیم:
«یکی از نکات مهمی که جلوی توسعه را می‌گیرد عدم خلاقیت و نو آوری است و این عدم خلاقیت و نو آوری به این خاطر است که ملت ما مدتها تحت سیستمهای استبدادی بوده و این فرهنگ خواه‌ناخواه در جامعهٔ ما جا باز کرده و مردم عادت کرده‌اند هرچه را از آنها بخواهند انجام دهند و هیچگاه عادت نکرده‌اند تفکر داشته‌باشند و نوآوری کنند... دیگر کاملاً واضح است که فرزند این فرهنگ چیست. آری! نالیدن و تلاش نکردن برای بهتر شدن وضعیت موجود...»
مطلب جالب توجه دیگر "تذکرة‌العلما"یی است که به سبک "گل‌آقا" و "مهر" به رییس دانشکده ابراز ارادت کرده!
"چه باید کرد؟" مقاله ایست که به مشکلات نظام آموزشی می‌پردازد و سه علت اساسی ضعف آنرا (که مصداق بارزاش "کم‌کاری و فرسایش روح تلاش علمی دانشجویان" است) اینطور بیان می‌کند:
۱) ابهام در فلسفهٔ تعلیم و تربیت
۲) سردرگمی در اعمال یک متد علمی منسجم در آموزش دروس کلاسیک توسط اساتید
۳) عدم توانایی سیستم آموزشی -از جمله اساتید- در ایجاد «انگیزش»

"چرا انگلیسی؟"
«امروز زبان انگلیسی کلید در گشا و چراغ راهگشای عرصهٔ پژوهش و روادید گذر از مرزهای علم است... متأسفانه در نظام آموزشی ما زبان انگلیسی از محوریت و رسمیت لازم برخوردار نیست و در جامعهٔ دانشگاهی ما نوعی حالت استغنا و بی‌اعتنایی (و گاهی نوعی رمیدگی و دلزدگی) نسبت به آن مشهود است. بنظر نگارنده بی‌مهری نسبت به این مقوله در ایجاد جو افسردهٔ حاکم بر دانشگاههای ما بی‌تأثیر نبوده است. در فضایی که ذوق و شوق «آموختن» در مواجهه با محتویات کم مایه و مندرجات مغلوط جزوه‌ها و یادداشتهای کلاسی جان می‌دهد و رنگ می‌بازد، وضع اسفبار فعلی آنچنان هم غیرمنتظره نیست...»
عناوین سایر مقالات این شماره:
"اتحادیهٔ دانشجویی، تحقق یا آرمان؟"، "یک پایان‌نامهٔ جالب"، "لزوم‌ایجاد گرایشهای مختلف در دورهٔ دکترای عمومی داروسازی"، "آشنایی با انجمن متخصصین علوم دارویی ایران"، "طرح ژنریک"، "وجه تسمیه داروهای تجاری"، "خاطرات پنی‌سلین"، "جدول"، "سندرم کمبود نمره" و "معرفی یک داروی جدید"

«موج هربار که‌به‌ساحل می‌آید با خود گوهرهای درخشانی را از دریای افکار و دلهای شما دوستان به‌ارمغان می‌آورد تا تلألو آن شادی‌بخش دلها باشد. حدیث غربت موج و غروب ساحل دریا چشمان را به آینده‌ای دور می‌دوزد تا دستان پرتلاش دیگری حدیث پویش آب را تکرار کند.»

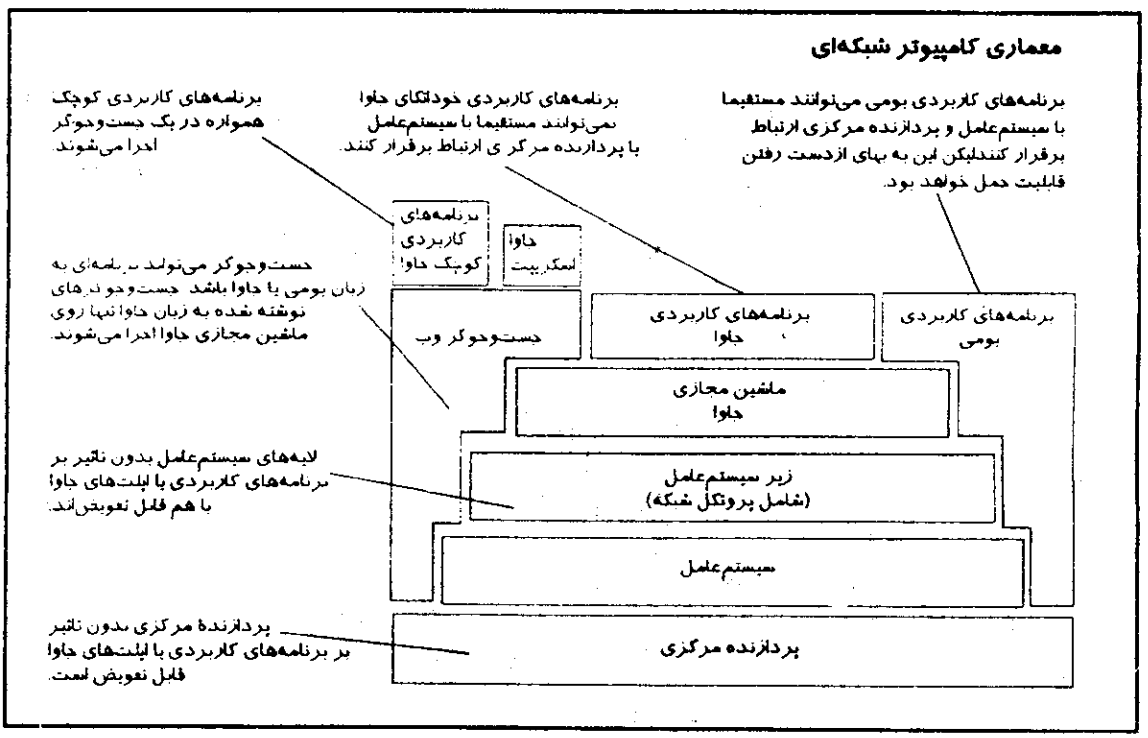
وسرانجام آنکه:
امیدواریم حکایت این امواج، تا ابد در ذهن صدفهای ساحل بماند (که‌می‌ماند).

کامپیوتر شبکه (NC) چیست؟

NC مخفف Network Computer به معنای کامپیوتر مخصوص شبکه است. در واقع این اصطلاح به نوع یا مدل خاصی از کامپیوترها اشاره نمی‌کند. بلکه همانند کامپیوتر شخصی (PC)، NC هم یک اصطلاح عام است که یک سیستم استاندارد را بیان می‌کند. دلیل ایجاد چنین استانداردی، گسترش روزافزون کاربردهای شبکه در دنیای کامپیوتر است. درواقع امروزه بخش قابل توجهی از استفاده‌کنندگان کامپیوترهای شخصی، از آن برای برقراری ارتباط با شبکه (معمولاً اینترنت) استفاده می‌کنند و بخشی از قابلیتهای PCها در این نوع استفاده کمتر بکار می‌آیند. بنابراین تولیدکنندگان کامپیوتر به این نتیجه رسیدند که با حذف این قابلیتها، می‌توانند کامپیوتری تولید کنند که به خوبی از عهدهٔ کارهای مربوط به شبکه برآیند و در عین حال قیمت پایینی هم داشته‌باشند. این فکر پایه ایجاد استاندارد جدید NC توسط شرکت Oracl شد. بر اساس این استاندارد، یک کامپیوتر شبکه‌ای باید دارای توانایی‌های خاصی باشد. قابلیت اتصال به اینترنت (دیدن صفحات وب، دریافت و فرستادن نامهٔ الکترونیکی، شرکت در بحث‌های گروههای خبری و...) و امکان اجرای برنامه‌های جاوا مهمترین این توانایی‌ها هستند. برای ایجاد این قابلیتها برخی امکانات باید ضرورتاً در یک NC تعبیه شوند: یک صفحهٔ نمایش حداقل ۶۴۰×۴۸۰ نقطه‌ای، یک وسیلهٔ اشاره‌گر (مثل ماوس)، وسیله‌ای برای وارد کردن متن (مثلاً صفحه‌کلید) و خروجی صدا. بعلاوهٔ یک اتصال شبکه‌ای که بتواند بسته‌های IP (پروتکل اینترنت) را انتقال دهد. همچنین نرم‌افزارهای کاربردی از قبیل مرورگر وب، برنامه‌هایی برای پشتیبانی از قالبهای چندرسانه‌ای اینترنت (مثل پرونده‌های Gif , Jpeg , Wav , Avi و...)باید بر روی این کامپیوترها اجرا شوند.

توجه کنید که استاندارد فوق به هیچ مدل خاصی برای پیاده‌سازی این قابلیتها تکیه نمی‌کند و هر کامپیوتری (با هر نوع پردازنده و...) در صورت سازگاری با استاندارد می‌تواند به‌عنوان یک کامپیوتر شبکه‌ای شناخته شود. مثلاً ممکن است برخی از کامپیوترهای شبکه‌ای، جعبه‌های روتلویزیونی باشند که از تلویزیون بعنوان صفحهٔ‌نمایش استفاده‌می‌کنند و از طریق یک تلفن یا مودم به شبکهٔ اینترنت متصل می‌شوند.

شکل زیر معماری یک کامپیوتر شبکه‌ای را نشان می‌دهد.



خلاصه‌شده از: خبرنامه انفورماتیک شماره ۶۴ / تیر ۱۳۷۶

استفاده از FEDها برای ساخت نمایشگرهای قابل حمل

کامپیوترهای کیفی قابل حمل،امروزه بخش مهمی از بازار را اشغال کرده‌اند.در راه توسعه این نوع از کامپیوترها، موانعی وجود داشت که باید برطرف می‌شد. یکی از این موانع، مشکلات LCDها بود. LCDها (که بطور سنتی برای ساختن نمایشگر کامپیوترهای کیفی مورد استفاده قرار می‌گرفتند) کنتراست کافی برای استفاده در حالتهای نورانی مختلف را، حتی با صرف انرژی فراوان تأمین نمی‌کنند. بعلاوه LCDها حساس به دما بوده به زمانی برای گرم شدن نیاز دارند و در دماهای پایین، سرعت عکس‌العملشان پایین می‌آید. برای حل این مشکل، راههای مختلفی پیشنهاد شد که یکی از امیدوارکننده‌ترین آنها،تکنولوژی FEDهای کاتد سرد بود.اصول کار FEDهای کاتد سرد همانند CRT است،ولی بجای استفاده از یک تفنگ الکترونی بزرگ با ولتاژ بالا،FEDها از آرایه‌ای از گسیل کننده‌های ولتاژ پایین برای تحریک نسفرهای مولد نور (که صفحه را روشن می‌کنند)استفاده می‌نمایند.FEDهای کاتد سرد،کنتراست نمایشی روشن‌تر از LCDها تولید کرده،نسبت به دمابی اعتنا هستندو انرژی مصرفی آنها به میزان قابل توجهی کمتر از LCDها است.(یک FEDده اینچ تمام رنگ به توانی کمتر از ۵ وات نیاز دارد).

شرکتهای مختلف، انواع مختلفی از FEDها را ارائه کرده‌اند.SI diamondبر روی یک FEDکار میکند که از یک روکش الماس برای یک سطح گسیل کننده میدانی استفاده مینماید.در حالی که بیشتر گسیل کننده‌های مرسوم فلزی هستند.(گسیل میدانی :گسیل الکترونها از فلز یا نیم رسانا به خلأتحت اثر میدان الکتریکی قوی).البته الماس بکار گرفته شده، از نوع الماس سنتی و گرانقیمت نیست، بلکه الماس توسط یک لیزر با شدت بالا از گرافیت بدست می‌آید.از دیگر مسائل پیش‌روی FEDها، جهازگردانهای گرانقیمت مورد نیاز برای قطع ووصل ولتاژ وهمگن نبودن تصاویر نمایش داده شده بودکه شرکتهابرای غلبه بر آنها تلاش زیادی کردند.

ترجمه: علی‌حاجی‌زاده

Active X چیست؟

Active X فن آوری تازه ای است که مایکرو سافت به دنیای web عرضه کرده و با استفاده از آن، پایگاههای وب می‌توانند براحتی از طیف وسیعی از آثار چند رسانه‌ای، اشیاء محاوره‌ای و برنامه های پیچیدهٔکاربردی استفاده کنند. در واقع Active Xکارهای انجام شده وزبانهای برنامه نویسی گوناگون (مثل delphi,c++,visual basic,...)و اسناد غیر HTML(مثلاً فایلهای تهیه شده توسط EXCELیاWORD)و برنامه های جاوا و...را به هم و به صفحات وب می‌چسباند و آنها را در یک سند HTML قرار می‌دهد. به این ترتیب توسعه دهندگان وب می‌توانندبدون آنکه درگیر یادگرفتن زبانهای مختلف شوند، خلاقیتشان را بکارگیرند و با استفاده ازهیر زبان برنامه‌نویسی که دوست دارند، صفحات وب محاوره‌ای خود رابسازند. بعلاوه با استفاده از Active X تهیه‌کنندگان صفحات وب مجبور نیستند از صفر شروع کنند و همه اجزای پایگاه وب محاوره‌ایشان رادستی بسازند، چرا که کنترلهای Active X بسیاری برای کارهای گوناگون در بازار موجودند ومی‌توانند مورد استفاده قرار بگیرند.

با وجود این همه مزایا، چرا Active X در این مدت عالمگیر نشده؟ در واقع مشکل به محدود بودن دامنهٔ سیستم عامل‌هاو مرورگرهای سازگار با Active X برمی‌گردد Active Xتنهاتوسط سیستم عامل ویندوز پشتیبانی می‌شود.بعلاوه بر روی همین سیستم عامل هم از میان تمام مرورگرهای وب موجود، تنها با مرورگر شرکت مایکروسافت(Internet Explorer 3)سازگار است.این در حالی است که تعداد بسیار زیادی ازکاربران اینترنت، از مرورگر Netscape Navigator استفاده می‌کنند وشرکت Netscape هم‌که در زمینه‌وب درگیر رقابتی شانه به شانه با مایکروسافت است مایل نیست با پشتیبانی‌از Active Xزمینهٔ رشدو استانداردشدن آنرا فراهم کند و بجای اینکار، ترجیح می‌دهدبا ترویج Java,Java Scriptبه نیازهای استفاده کنندگان پاسخ گوید. (هر چند ظاهراً Netscape سرانجام تصمیم گرفته در نسخهٔ آتی Navigator قابلیت سازگاری با Active X را بگنجاند) در واقع Active X هم گرفتار همان مشکلی است که پیش از این Visual Basic Script شرکت مایکروسافت (که به عنوان رقیبی برای Java Script عرضه شده بود)به آن دچار شده بود.

مایکروسافت راه درازی در پیش دارد و تا چشم کار می‌کند،Java افق را پوشانده!

ترجمه:حمیدرضا مختاریان

HTTP 1.1

مصاحبه با Jim Getty دانشمند علوم کامپیوتر

س:پیشرفتهای اصلی http در نسخهٔ ۱.۱ چیست؟

ج:نسخهٔ ۱.۰ http (پروتکلی برای انتقال Hypertext) بخوبی از TCP استفاده نمی‌کند. http بر مبنای بالاترین لایهٔ پروتکل انتقالی TCP کار می‌کند و شیوهٔ کار نسخهٔ ۱.۰ در این زمینه زیاد ماهرانه نبود. برای هر شیء کوچک گرافیکی که به روی صفحه نمایش خود می‌دیدید،نسخهٔ ۱.۰ ارتباط TCP مستقلی ایجاد می‌کرد. صفحه های شبکه (Web Page) فراوانی که به آنها دسترسی دارید، تعداد بی شماری اشیاء کوچک را شامل می‌شود، و TCP برای استفاده در این حالت ویژه چندان مناسب نیست. از آنجا که ارتباطهای TCP اغلب و بسیار سریع دور ریخته می‌شوند، این مسئله بر الگوریتهای تراکم گریز (Congestion Avoidance) TCP فائق می‌آید.

س:http نسخهٔ ۱.۱ چه روشهایی را برای رفع این مشکل به کار می‌برد؟

ج:نسخهٔ ۱.۱ این مشکل را با استفاده از دو روش حل می‌کند. در روش نخست، که ارتباطات پایا (Connections Persistent) نامیده می‌شود، بجای دور ریختن ارتباطات پس از هر استفاده، ارتباط باز نگه‌داشته می‌شودو شما از آن برای درخواستهای بعدی استفاده می‌کنید. روش دیگر که لوله کشی (Pipeling) نامیدو می‌شود، اجازه می‌دهد که سرویس گیرنده در آن واحد درخواستهای چند گانه‌ای داشته باشد. با استفاده از روش لوله کشی مانند این است که شما لیستی از چیزهایی را که می‌خواهید سرویس دهنده به شما تحویل دهد، آماده می‌کنید، بجای اینکه در هر بار یکی را درخواست کنید. لوله کشی کردن، در مقایسه با ارتباطات پایا به شما اجازه می‌دهد که اجرای دلخواه خود را بگیرید.

س: هنگام آزمایش http 1.1 باچه فوایداجرایی موجه شدید؟

ج: اگر کاری که انجام می‌دهید Dwonload کردن یک تصویر بزرگ باشد،هیچ جایگزینی برای پهنای باند وجود ندارد. http 1.1 نسخهٔ نمی‌تواند یک مودم ۲۸/۸ را به اجرای سریع‌تر وا دارد. با این وجود برای download کردن صفحه شخصی (Home Page) آزمایشی خودمان روی یک خط ۲۸/۸از یک سرویس دهنده، که روی اینترنت بسته شده بود، برای اولین بار، درمقایسه با http 1.0، ۲۰ درصد پیشرفت مشاهده کردیم. این یک برآورد محافظه کارانه است و حداقل آن چیزی است که مردم می‌توانند انتظار داشته باشند به دست آورند. درعمل ممکن است کارآیی برای بیشتر مراکز (Sites) بیش از این پیشرفت کند چراکه http 1.1 بطور گسترده در حال گسترش است و تراکم ایجاد شده توسط http 1.0 روی اینترنت کاهش می‌یابد. با این وجود http 1.1 برای Cash Validation [که هنگام دوباره دیدن یک صفحه، زمانیکه چیزی عوض نشده، انجام میشود.] حتی بر روی یک خط تلفنی، بطور فوق العاده ای بهتر از نسخهٔ ۱/۰ کار می‌کند. در آزمایشها درحدود ۴ برابر پیشرفت در مورد Cash Validation مشاهده کردیم. برای سایر محیطهای شبکه، در کل کارهای بهتری انجام داده‌ایم. برای نمونه، دریک گذر میان کالیفرنیا [در آزمایشگاه Lawrance Berkley] و MIT، مادر نخستین آزمایشهای بازیابی دوبرابر پیشرفت و همچنین ۸ برابر پیشرفت در زمینهٔ Cash Validation بدست آوردیم. [در مقایسه با http 1.0]. سپس در آزمایشهای اترنت محلی خودمان مادوبرابر برای Cash Validation پیشرفت حاصل کردیم. بنابراین http 1.1 دگرگونی عظیمی در احساس مردم نسبت به اینترنت بوجود خواهد آورد.

ترجمه: علیرضا ماندگار

برای اطلاعات بیشتر در مورد http 1.1 به آدرس زیر مراجعه نمایید :

http//www.w3.org/pub/www/Protocols/HTTP/Preference/pipeline.html

دو پیشرفت فنی خیره‌کننده در صنعت تراشه‌ها

دانشمندان کامپیوتر، همانندشخصیت رمانهای روسی، به دو گروه تقسیم می گردند:گروه خوشبینهاوگروه بدبینها.
گروه دوم همواره درکتابها، کنفرانسها و در مجلات، ابراز میدارند که علم کامپیوتر تا آخرین حد ممکن، پیش رفته‌است.
آنهااستدلال می‌کنند که اندازهٔ اتم و الکترونهای دور آن میزان فشرده سازی ترانزیستورهادر یک تراشهٔ سیلیکونی را محدود می‌کنند.

امـا گـروه دیگـر کـه گـوردن مـور (Gordon Moore) شاخصترین آنهااست، معتقدندکه تراشه هاباسرعتی قابل پیش بینی کوچکتر و سریعتر می‌شوند. (بر اساس نظریهٔ سال ۱۹۶۵ مور که می‌گوید در هر ۱۸ ماه ظرفیت تراشه ها دو برابر خواهد شد.)

موفقیت صنعت کامپیوتر تا حدود زیادی به حقیقتی که مور آنرا ارائه داده است، بستگی دارد. اماحتی گـروه خـوشبین بـه صـنعت ریزپردازنده نیز، به این نتیجه رسیدند که اگر قرار باشد قانون مور همچنان معتبر باقی بماند باید تحول عظیم و سـریعی در صـنعت ساخت تراشه اتفاق افتد.

به تازگی دو نمونه از این تحولات رخ داده است:

۱) شرکت اینتل (بزرگترین شرکت جهانی نیمه هادیها) اعلام کرده است که مهندسین آن ترفندی زیرکانه برای ذخیره‌سازی دو برابـر حجم اطلاعات دیجیتالی در همان فضای فیزیکی پیشین، بر روی یک تراشه کشف کرده‌اند.

۲) هفتهٔ گذشته شرکت آی.بی.ام مطلبی را فاش کرد که شاید خیلی چشمگیرتر باشد: مـحققین ایـن شـرکت راهـی بـرای جـایگزینی هـادیهای آلومـینیومی بـا مس (کـه ارزانـتر و سـریعتر است) در ریزپردازنده های خود یافته‌اند.

این کاری عظیم است اما نه فقط به این علت کـه مس ارزانـتر است، بلکه تحول اساسی در این است که هـادیهای مسـی ریـزتر ساختن تراشه هارا ساده می‌کنند. این مورد راه حل مناسبی بـرای تراشه سازان است که از نظر بدبینها زمان بسیار طاقت فـرسایی را صرف می‌کردند تا الکترونها را در مسیرهای آلومـینیومی هـر چـه باریکتر (که هر قدر باریکتر می‌شوند، رسانایی‌شان کمتر می شود) به جریان در آورند.

یک دههٔ قبل، از زمانی که دانشمندان پی بردند که برای رسیدن به مراحل بالاتری در کوچک سازی (با سـیمهایی بـه قـطر ۰/۲۵ میکرون حدود ۱/۴۰۰ قطر موی انسان)، باید آلومینیوم را به کلی کنار گذارد. مس جانشینی مشخص و مناسب بود. اما دو مشکل به ظاهر غیر قابل حل داشت. اولین مشکل هـنکامی ظـاهر شـد کـه دانشمندان سعی داشتند لایه ای از مس را روی سطح سیلیکون قرار دهند. اتمهای ریز مس در سیلیکون متخلخل مانند عبور قهوهٔ داغ ازصافی قهوه جوش نفوذ می‌کنند. مس به قدری رسانا است که حتی یک اتم فعال آن می تواند تمام سطح سیلیکون را غیر قابل استفاده نماید.

مشکل دوم قدری ظریفتر بود: سیمی که قطر آن ۰/۲۵ میکرون است آنقدر نازک است که پس از ساخته شدن نمی‌توان آنرا جابجا یا لمس کرد. پس مهندسین ریزپردازنده بجای آنکه سیمهای نازک را در تراشه های سیلیکونی قرار دهند، رویهٔ نازکی از فـلز را روی آن می‌کشند و سپس کلیهٔ قسمتهای غیر لازم را پاک می‌کنند. چیزی که باقی می‌ماند مسیرهای میکروسکوپی فلزی است که فقط آنقدر قطر دارند که جریان مورد نظر را عبور دهند. اما در حالیکه تراشه‌سازان راههای بسیاری برای پاک کردن آلومینیوم اضـافی از روی تـراشـه یافته‌اند، هیچکس تا کنون راهی برای پاک نمودن مس اضافی نیافته است. انجام این عمل، به نـظر آی.بـی.ام نـیاز بـه اخـتراع شـاخهٔ جـدیدی از عـلم شـیمی داشت. درو پک (Drew Peck) -از متخصصین نیمه هادی در .Cowen and Co- اظهار میدارد :

«انجام این عمل به صورت هدفی مقدس در آمده بود.»

یک دهه بعد، بعد از تمام موانع موجود، شرکت Big Blue هر دو مشکل را حل کرد. برای اولی آنها پوششی مخصوص که بین مس و سیلیکون باقی می‌ماند اختراع نمودند (نوعی از ریز تراشه Gortex که تنها اجازه عـبور جـریان مشـخصی را بـین لایـه هـا می‌دهد.) آنها همچنین فن‌آوری پاک نمودن بخشهای اضافی را که نه تنها مس را پاک می کرد، بلکه حتی دسترسی تراشـه سـازان را بـه دنیاهای کوچکتر از ۰/۲ میکرون تا ۰/۰۵ مـیکرون سـپیش بـینی بعدی مور- را ممکن می کرد.

شرکت اینتل که مور هنوز هم هفته ای سه روز را در آن کار می‌کند، فن آوری ساخت تراشه را در بعد دیگری پیش برده است. دو هفته قبل این شرکت اعلام کرد که نو آوری مهمی در تراشه های حافظه فلاش (نوعی حافظه که در اکثر وسایل از دوربینهای دیجیتالی گرفته،تا وسایل ضبط پیام فاقد نوار استفاده می شود) داشته است.

حافظه های فلاش مثل اکثر حافظه های کامپیوتر اطلاعات را به صورت دودویی ذخیره می کنند که معمولا" با دو حالت صفر و یک نمایش داده می شوند. نوآوری اینتل پیدا کردن راهی برای ذخیره سازی دو برابر اطلاعات دودویی در هر ترانزیستور حافظهٔ فلاش است. این مطلب دقیقاً مثل این است که شما در اتومبیل خود دو برابر ظرفیت مسافر سوار کنید بدون اینکه بنزین اضافی مصرف شود و یا احساس تنگی جا نمایید! مهندسین اینتل این اعجاز را با وسایل فوق‌العاده حساس جدید که وضعیت الکتریکی تراشه های آنها را تشخیص می‌دهند انجام داده‌اند.

اینتل می‌گوید که اولین سوپر تراشه های خود را تا قبل از پایان امسال به همراه یک کارتریج حافظه فلاش ۶۴ بیتی که تا دو سال پیش غیر ممکن به نظر می‌رسید، روانهٔ بازار خواهد کرد. آی.بی.ام نیز که فن آوری مس خود را پنهان داشته است اولینهای آن را در سال ۱۹۹۸ خواهد فروخت.

کاربران معمولی کامپیوتر بااین قدرت فوق العاده جدید چه خواهند کرد؟ هر دو شرکت بر این عقیده‌اند که کامپیوترهای سریعتر و باهوشتر می‌توانند در آینده امتیازات زیادی را به ارمغان آورند، از کامپیوتر هایی که فرمان می‌پذیرند گرفته تا ماشینهای رومیزی، بدون هیچ زمان طاقت فرسای راه اندازی، آماده به کارند. اما هیچ کدام از این دو فن‌آوری تراشه به خودی خود نمی‌توانند کامپیوتر شخصی گران قیمت شما راکه هفته پیش خریده‌اید از دور خارج کنند. کاری‌که قانون مور با گذشت زمان خواهد توانست... .

ترجمه:**احد هراتی**

CD قابل خواندن و نوشتن

امروزه دیگر CD پدیدهٔ جدیدی نیست. پس از CD-ROMهای اولیه که دیسکهایی با ظرفیت زیاد (۶۴۰ مگابایت) ولی فقط خواندنی (Read Only) بودند،CD-R به بازار آمد. این نوع از CD قابلیت آنرا داشت که خریدار پس از خرید، یکبار برروی آن بنویسد ولی پس از آن دیگر امکان تغییر محتویات CD وجود نداشت. DVD هم که پس از CD عرضه شد هرچند ظرفیت را تا حد باور نکردنی افزایش داد (۴ تا ۱۷ گیگابایت در هر دیسک) ولی باز هم امکان خواندن و نوشتن به تعداد دفعات نامحدود را نداشت. CD-RW (یا CD با قابلیت نوشتن مجدد) به این مشکل پایان داد. این CDها با حفظ ظرفیت بالای سایر CDها امکان بیش از هزار بار ضبط و پاک شدن را دارند و عمر مفید ۳۰ ساله‌شان آنها را به انتخابی ایده‌آل برای پشتیبانگیری از داده‌ها مبدل می‌سازد. درایورهای CD-RW با همهٔ نسخه‌های قبلی CD سازگارند، یعنی می‌توانند همهٔ انواع CD را بخوانند و بر روی CD-Rها بنویسند و CDهای ضبط شده توسط آنها با هر درایو CD یا DVD معمولی قابل خواندن است. ظاهر آنها هم کاملاً شبیه هر درایو CD دیگری است. و دارای یک گذرگاه اسکازی۲ (SCSI-2) می‌باشند. تفاوت آنها با سایر درایوهای CD در استفاده از تکنولوژی تغییر فاز چهارگانه -که در آن دیسکهای مختلف به‌وسیلهٔ تنظیم خودکار مدولاسیون لیزر خواننده/نویسنده کار می‌کنند- است. قیمت ۲۵ دلاری دیسکهای CD-RW (که پنج برابر بیشتر از CD-Rها است) ممکن است درنگاه اول مانعی در راه رشد آن به‌نظر برسد، ولی درواقع اینطور نیست، چرا که CD-Rها قابلیت به روز کردن اطلاعات ذخیره‌شده را ندارند و در نتیجه عمر مفید آنها نسبت به CD-RWها بشدت کاهش می‌یابد. بعلاوه CD-RWها می‌توانند از استاندارد جدید فرمت‌جهانی‌دیسک (UDF) -که روزبروز گسترش بیشتری می‌یابد پشتیبانی کرده عمل نوشتن را به صورت بسته‌ای (Packet Writing) انجام دهد که اینکار CD-RW را شبیه یک درایو هارد دیسک‌می‌سازد. البته CDای که توسط UDF ساخته شده باشد برای شناخته‌شدن توسط PC نیاز به یک درایور دارد.

در نهایت، CD-RW یک تکنولوژی گذرا در راه رسیدن به DVDهای قابل بازنویسی آینده‌است. اما مزایای فراوان آن، آنرا به انتخابی مناسب برای بسیاری کاربردها در دنیای ذخیره‌سازی داده‌ها مبدل می‌کند.

ترجمه: **بهنام دانمی**

جاوا چیست؟

خیلی‌ها معتقدند جاوا زبان برنامه‌نویسی آینده است.
این‌روزها جاوا دارد همه‌گیر می‌شود! درواقع جاوا بابه‌پای توسعهٔ وب، گسترش می‌یابد. جاوا به‌دلیل بهره‌مندی از ماشین‌مجازی (VM) و عدم وابستگی به سیستم‌عامل یا معماری بخصوصی از کامپیوتر، از حد یک زبان برنامه‌نویسی معمولی فراتر رفته و به بستری برای رشد نرم‌افزار تبدیل گردیده‌است.

زبان برنامه‌نویسی جاوا

نوشتن یک برنامهٔ جاوا از جهات بسیار زیادی شبیه زبان C++ است که این موضوع خودش یکی از دلایل محبوبیت جاوا در بین برنامه‌نویسان است. درواقع تفاوت اصلی جاوا با C++ (و سایر زبانهای مبتنی بر مترجم «Compiler») در نحوهٔ اجرای برنامه‌هاست. زبانهای مرسوم فعلی برای اجرای برنامه‌ها از یکی از دو شیوهٔ ترجمه یا تفسیر استفاده می‌کنند. در روش ترجمه، زبان برنامه‌نویسی دارای یک مترجم (کامپایلر) است که کد برنامهٔ نوشته‌شده را به زبانِ‌ماشین دستگاه مورد استفاده تبدیل می‌کند و نتیجه را در یک فایل که فایل اجرایی برنامه (معمولاً دارای پسوند EXE. در نام فایل) نامیده می‌شود قرار می‌دهد. این‌کار برای هر برنامه یکبار انجام می‌شود و از این پس، هر زمان که نیاز به اجرای برنامه باشد، کافی است که فایل اجرایی برنامه اجرا شود و دیگر با کامپایلر کاری نخواهیم داشت. در روش تفسیر، مفسر (اینترپرتر) زبان، برنامهٔ نوشته‌شده را خط به خط می‌خواند، به‌زبان ماشین ترجمه می‌کند و در نهایت آنرا اجرا می‌نماید. بعد به سراغ خط بعدی می‌رود. به‌این ترتیب، هرزمان که نیاز به اجرای برنامه باشد، باید از مفسر استفاده کرد. زبان Basic اولیه چنین زبانی بود. در حالی‌که سایر زبانها از قبیل C و Pascal (و نیز نسخه‌های جدیدتر Basic) از مترجم استفاده‌می‌کنند.

ولی جاوا، کاری متفاوت انجام می‌دهد. جاوا دارای یک مترجم است که مانند سایر مترجم‌ها، متن برنامه (که به java. ختم می‌شود) را خوانده از آن یک فایل اجرایی می‌سازد (که در پایان نامشان class. قرار می‌گیرد و بایت‌کد «byte-code» نامیده‌می‌شوند). تفاوت این بایت‌کدها با فایلهای اجرایی در آن است که بایت‌کدها از دستورات زبان ماشین خاصی استفاده نمی‌کنند، بلکه دستورات یک ماشین فرضی بنام "ماشین مجازی جاوا" (Java Virtual Machine) را بکار می‌گیرند. بنابراین بایت‌کدها (که فایل اجرایی ماشین مجازی هستند) بر روی این ماشین اجرا می‌شوند. این ماشین مجازی برای هر سیستم بطور نرم‌افزاری شبیه‌سازی می‌شود. با این روش دیگر نیازی نیست که هر برنامهٔ جاوا برای هر ماشین خاص جداگانه کامپایل شود. بلکه یکبار با کامپایل شدن، می‌تواند برروی هر دستگاهی که ماشین مجازی‌جاوا بروی آن شبیه‌سازی شده‌باشد اجرا شود. یعنی برنامه‌های جاوا، مستقل از بستر سخت‌افزاری و سیستم‌عامل و... همه‌جا قابل اجرا خواهند بود. به این استقلال از سخت‌افزار،اصطلاحاً بسترتقابلی (Cross-platform) گفته‌می‌شود. سیستم‌عاملهایی مثل Unix و Windows NT هم هستند که ادعا می‌کنند مستقل از سخت‌افزارند، اما آنها صرفاً یک سیستم چندبستری (Multi-Platform) هستند. در سیستم عاملهای فوق، کاربر درصورت تعویض پردازنده باید برنامهٔ خود را مجدداً کامپایل کند. بعلاوه همهٔ نرم‌افزارها در هر سیستم‌عاملی قابل حصول نیستند. مثلاً برنامه‌های زیادی برای Linux (که نسخهٔ خانگی یونیکس است) نوشته شده که هیچکدام بر روی Sco Unix قابل استفاده نیستند. بعلاوه این سیستم‌عاملها شما را مقید به سیستم‌عامل می‌کنند، در صورتیکه جاوا، کاربر را از سیستم‌عامل منفک می‌گرداند.

کدام جاوا؟

جاوا وقتی به بازار رسید که تب وب همه‌جا را فرا گرفته بود. قابلیت اجرای برنامه‌های جاوا بر روی سیستم‌های گوناگون در کنار برخی مزایای دیگر آن، جاوا را مورد توجه توسعه‌دهندگان وب ساخت. پس از آن بود که چیزی بنام آپلت جاوا (Java Aplet) متولد شد.

آپلت جاوا به برنامه‌های جاوایی گفته می‌شود که برای قرارگرفتن در صفحات وب (Web Pages) نوشته‌شده‌اند. نوشتن این آپلت‌ها بسیار شبیه نوشتن هر برنامهٔ جاوای دیگری است؛ با برخی تفاوت‌ها. مثلاً اینکه آپلت‌های جاوا به‌جای تابع ()main، تابعی به‌نام ()index دارند.

پس از آنکه آپلت‌های جاوا با استقبال روبرو شدند، شرکت Netscape تصمیم‌گرفت از آن در مرورگر وب خود استفاده کند. حاصل کار چیزی بنام جاوااسکریپت (Java Script) بود. جاوااسکریپت زبانی‌است که در نوشتن صفحات HTML مورد استفاده قرار می‌گیرد. به‌این ترتیب به HTML -که تنها زبانی است برای شکل‌دادن به صفحات وب- قابلیتهای برنامه‌نویسی اضافه می‌شود. بنابراین، جاوااسکریپت زبانی جدید است که بخاطر شباهتهایش به جاوا و نیز بدلیل محبوبیت جاوا، اسمش شبیه اسم آن انتخاب شده.

پنج مشکل اساسی جاوا

با وجود مزایای آشکار جاوا، هنوز دلایلی وجود دارد که پدیدآورندگان نرم‌افزار را از کنار گذاشتن سیستم‌عاملها و زبانهای محلی مبتنی بر سخت‌افزار خاص باز می‌دارد:

۱) سیستم‌عاملی که برای یک سیستم خاص ساخته شده، از بسیاری تواناییهای خاص آن دستگاه استفاده خواهد کرد. درحالی که‌جاوا سازگاری چندبستری خود را با قربانی‌کردن این تواناییها بدست می‌آورد.

۲) سرعت‌اجرای بایت‌کدهای جاوا نسبت به کدهای محلی سیستم بسیار پایین‌تر است.

۳) برنامه‌نویسانی که به ابزارهای یک زبان عادت کرده‌اند، دوست ندارند آنها را کنار بگذارند.

۴) برنامه‌های بسیاری با زبانهای سابق نوشته‌شده،که تبدیل آنها به جاوا برای شرکتها مقرون‌به‌صرفه نخواهد بود.

۵) وقتی برنامه‌ای باید با داده‌ها و کدهای بجامانده از برنامه‌های دیگر کار کند، بیشتر ابزارهای شناخته‌شده و زبانها، باید بسیار بهتر از یک زبان سطح بالا مانند جاوا عمل کنند. (این موضوع بویژه برای شرکتهایی که نرم‌افزارهای تقابلی و اشتراکی تولید می‌کنند بسیار مهم است)

در یک دیدگاه کلی، جاوا زمانی بطور جدی گسترش خواهد یافت که امکان استفاده از ویژگیها و امکانات اختصاصی یک بستر اصلی را به راحتی فراهم سازد. دسترسی به ویژگیهای اختصاصی سیستم‌عامل محلی به دو روش امکان‌پذیر است. روش نخست (که بهتر است) بسته‌بندی و یا به اصطلاح کپسوله کردن سرویسهای اختصاصی و محلی به شکل کلاس‌فایلهای جاوا است. این روش باعث جداشدن انواع سرویسهای متعلق به بسترهای اصلی خواهدشد. روش دیگر دسترسی به امکانات محلی با فراخوانی و اجرای مستقیم کدهای محلی است. این روش باعث فداشدن سازگاری با انواع بسترها در کنارهم می‌شود.

محدودیت سرعت

برنامه‌نویسانی که سرعت اجرا برایشان خیلی مهم است، ممکن است جاوا را زبان دلخواهشان ندانند. این نوع برنامه‌نویسان با کامپایلرهای C یا C++ حداقل سرعتی ۱۰ برابر بیشتر از جاوا خواهند داشت. اما برای بسیاری سرعت چندان مهم نیست. بعلاوه با به بازار آمدن کامپیوترهای سریعتر، در بسیاری از موارد این مشکل بروز نخواهدکرد. بااین همه، متخصصان بدنبال تقویت عملکرد و رشد سرعت جاوا هستند. از جمله روشهایی که برای این منظور در نظر گرفته‌شده، ساختن ماشینهایی است که دستورات زبان ماشین آنها همان دستورات ماشین مجازی جاوا باشند و به این شکل، امکان اجرای مستقیم بایت‌کدهای جاوا فراهم شود. در حال حاضر شرکتهای سان و متاسیستم در حال طراحی و ساخت پردازنده‌ای هستند که قابلیت فوق را داشته باشد. این محصول به‌زودی تحت نام تراشهٔ جاوا (Java Chip) به بازار عرضه‌خواهدشد. [برای مطالعهٔ بیشتر درمورد

تراشه‌های جاوا مقاله بعدی این شماره تحت همین عنوان را بخوانید].

شاید امیدبخش‌ترین راه‌حل دررابطه با سرعت، روش مفسر کار درزمان (Just_In_Time=JIT) باشد. همانند سایر مفسرهای جاوا، این‌نوع مفسرها بایت‌کدهای جاوا را همزمان به کدهای محلی تبدیل می‌کنند، اما آنرا در یک حافظهٔ نهانی (Cache) ذخیره می‌کنند. با این‌روش می‌توان به ۵۰٪ سرعت‌کدهای محلی دست یافت. این تکنیک درحال بررسی و پیشرفت است.

روش دیگر تفسیر ثابت (Static) است. شرکت سیلکون‌گرافیک با این روش بایت‌کدهای جاوا را به کدهای محلی Mips ترجمه می‌کند و نتایج باینری بدست آمده را به کلاس فایل‌های جاوا ملحق می‌سازد. در واقع برنامه به بلوکهای مجزایی از کدهای ترجمه‌شده و نشده تبدیل می‌شود که‌هر بلوک قبل از آغاز اجرا چک می‌شود، اگر ترجمه‌شده به کدهای Mips باشد در محیط محلی و درغیراینصورت در محیط جاوا اجرا می‌شود. این روش باوجود سرعت خوبش از جهات سازگاری در محیط‌های مختلف ضعف دارد و کدهای ترجمه‌شده صرفاً مخصوص محیط مرجع خود هستند.

آینده

در دراز مدت هیچ‌یک از مشکلات تکنیکی که امروزه مانع رشد جاوا به‌نظر می‌رسند، غیرقابل حل نخواهد بود. چه‌از جنبهٔ امکانات زبان برنامه‌نویسی و چه از نظر بسترنرم‌افزاری، مسیری که جاوا به ما نشان می‌دهد بی‌سابقه است.

مهمترین گرایش در صنعت کامپیوتر توجه به نرم‌افزارهای سطح بالای مستقل از سخت‌افزار است. مشکل و دردسر عمده نیز درهم آمیختگی سخت‌افزار و نرم‌افزار در صنعت کنونی کامپیوتر است. برنامه‌نویس برای رسیدن به یک سیستم مطلوب و مناسب، زحمت و انرژی بسیاری برای پیاده‌سازی و بخصوص کدنویسی متقبل می‌شود و همزمان، سخت‌افزار به سرعت درحال تغییر و تحول است.

در جاوا کلیه کدها برای ماشین مجازی جاوا طراحی شده است، بصورتی که سیستم عامل و پردازنده قابل تعویض و تغییر هستند؛ بدون اینکه نیاز به تغییر و تحول در کد برنامه باشد. جاوا برای دورهٔ تحول و دگرگونی طراحی شده‌است. به این‌ترتیب این امکان فراهم می‌شود که برای رسیدن به بهترین‌بازدهی، کمترین‌هزینه، استفاده‌بهتر از انرژی، کم‌وزن‌ترین و یا هر پارامتر مورد نظر دیگر، پردازنده و سیستم‌عامل و... تغییر یابند. بعلاوه توفیق جاوا در زمینهٔ شبکه و یا کاربردهای اختصاصی و یا در PDAها قابل انکار نیست.

شرط‌بندی روی جاوا

درنهایت انتخاب جاوا بوسیلهٔ پدیدآورندگان نرم‌افزار، منوط به پاسخ‌گویی به دو سؤال است.

اول اینکه آیا جاوا می‌تواند از عهدهٔ وظایف نرم‌افزاری محوله بر آید؟ هنوز جاوا نمی‌تواند هرکاری بکند، اما با عرضهٔ ابزارهای جدید تواناییهایش بسیار بیشتر خواهد شد.

دوم اینکه آیا سازگاری بسترها به‌صورت تقابلی موضوع تعیین‌کننده‌ای است؟ اگر اعتقاد داریم که کامپیوترهای فردا همانند کامپیوترهای امروزی‌ولی با حجم حافظه و سرعت بالاتر هستند، در اینصورت جاوا بهترین انتخاب نیست. دیگر زبانها و ابزارها عملکرد بسیار بهتری در سخت‌افزارهای متداول دارند.

اما اگر دوست دارید برنامه‌ای بنویسید که قابل اجرا روی هر سخت‌افزاری باشد، در اینصورت جاوا بهترین است. مسلماً با HTML و JavaScript و Perl و VBscript و سایرراه‌حل‌های بستر تقابلی می‌توان خیلی کارها کرد. بخصوص اگر مایکروسافت در پروژهٔ ActiveX خود موفق شود. [برای اطلاعات بیشتر درمورد ActiveX به مقالهٔ "ActiveX چیست؟" در همین شماره مراجعه کنید.]

تراشه‌های جاوا

امروز یک برنامه کاربردی جاوا را از اینترنت download کنید، خواهید دید که پردازنده x86 یا RISC شما ایرادی پیدا نمی‌کند. این پردازنده‌های مرکزی (cpu) برای اجرای بهینه برنامه‌های کاربردی C طراحی شده‌اند، ولی با این وجود در شبیه سازی ماشین مجازی جاوا برای برنامه‌های ساده وب (applets) کنونی، نیز موفق عمل می‌کنند.

بنابراین تا زمانیکه جاوا چیزی پیچیده‌تر از برنامه‌های فعلی شبکه، تولید نکرده است، مشکلی پیش نخواهد آمد. امّا جاوا قابلیت آنرا دارد که خیلی بیشتر از این رشد کند. امکان سازگاری طرح زیربنایی و اصلی آن، برخی شرکتهای تولید نرم افزار مانند کورل (Corel) را، واداشته است تا برنامه‌های کاربردی - تجاری توسط جاوا را بطور عمده گسترش دهند.

آیا می‌توان به یک پردازنده همه منظوره اکتفا کرد و امیدوار بود که بتواند برنامه‌های کاربردی جاوا را در آینده به دقت اجرا کند؟

شرکت میکرو سیستم سان (Sun Microsystem) که جاوا را ارائه کرده است، برروی تراشه‌های جاوا، مختص برای انجام کارهای طراحی شده توسط جاوا و یا برنامه‌های جایگزین آن، سرمایه گذاری کرده است.

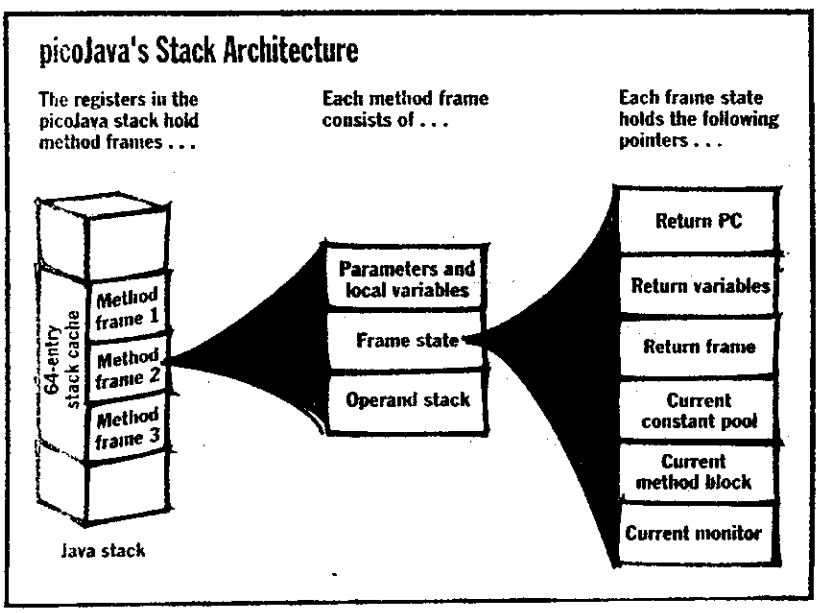
سان در حال کامل کردن نوعی مشخصات مرکزی و اصلی برای تراشه‌های جاوا می‌باشد که picoJava نام دارد. این معماری، تعدادی نو آوری در طراحی را، برای اجرای هر چه بهتر برنامه‌های جاوا، ارائه کرده است.

معماری Pico Java اساس و پایهٔ اولین نسل تراشه‌های جاوا، موسوم به micro Java خواهد بود. سان همچنین در حال کامل کردن تراشهٔ قدرتمندتری است که ultraJava نام دارد و برای استفاده در سیستمهای رومیزی (desktop) به کار خواهد رفت.

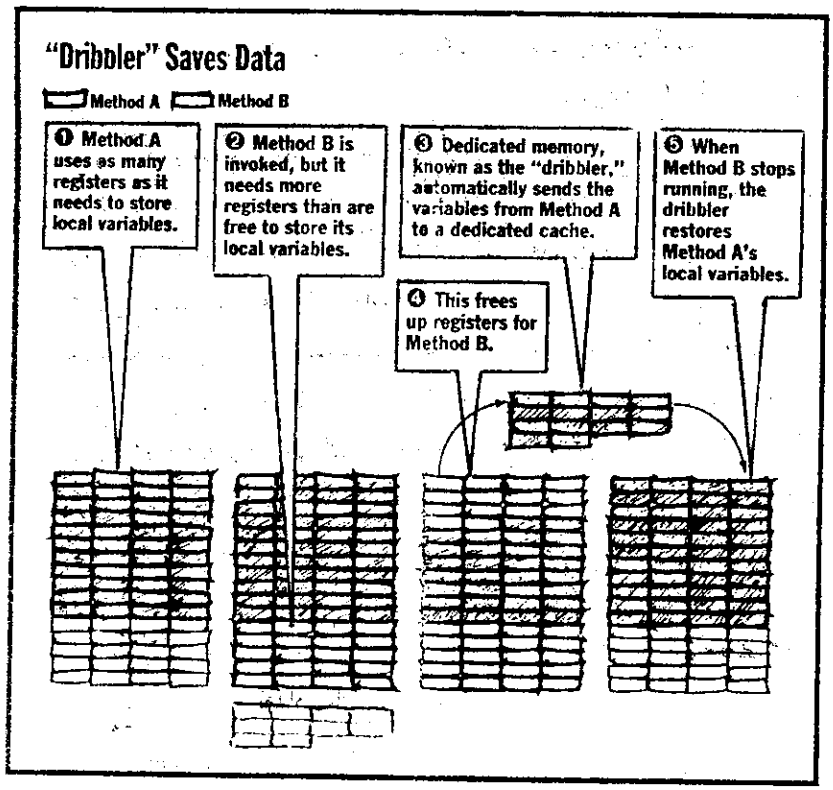
سان ادعا می‌کند که این تراشه‌ها، نسبت به مفسرهای کنونی جاوا، برنامه‌های جاوا در حدود ۱۲ برابر سریعتر اجرا می‌کنند. البته مفسرهای کدهای جاوا نیز بهبود یافته‌اند. برای نمونه اینتل مفسری برای پردازنده‌های سری x86 نوشته است و ادعا می‌کند که این مفسر، برنامه‌های جاوا را سه برابر سریعتر از مفسر سان اجرا می‌کند.

کامپایلرهای Just-in-time (JIT) برنامه‌های جاوا را سریعتر از مفسرها اجرا می‌کنند، اما سان معتقد است که تراشه‌های picoJava، پنج برابر سریعتر از یک پنتیوم به همراه کامپایلر JIT می‌باشند. از آنجا که یک تراشه جاوا، دستور العملهای جاوا را بدون تبدیل آنها به مجموعه دیگری از دستورات cpu اجرا می‌کند، نیازی به حافظهٔ اضافی یا حافظه نهانی -که در هنگام اجرای مفسر و یا کامپایلر جاوا توسط پردازنده‌های همه منظوره نیاز است- ندارد. همچنین کد تولیدی آن نیز عموماً کوچکتر از کد تولید شده توسط یک پردازندهٔ RISC می‌باشد. برای نمونه، بطور متوسط کد تولید شده توسط پردازندهٔ جاوا برای هر دستورالعمل، ۱/۸ بایت طول دارد، در حالیکه کد تولید شده توسط پردازندهٔ RISC برای هر دستورالعمل، بطور متوسط ۴ بایت طول دارد.

پرکردن پشته (STACK)



The picoJava stack uses 64 32-bit registers. picoJava allocates variables on the stack; method calls pass data through the stack.



Sun's "dribbler" is a clever method of caching data and returning it to the stack when registers become full.

چه چیزی سبب تمایز تراشه‌های picoJava از سایر پردازنده‌ها می‌شود؟ نخستین چیز چگونگی بهبود بخشیدن آن به پشته است. در معماری picoJava، تراشه‌های جاوا متغیرها را بطور موضعی در پشته جای میدهند و فراخوانی توابع و دستورالعملهای عملیاتی نیز داده‌ها را درون پشته می‌فرستند.

اغلب کامپایلرهای c کدبرنامهٔ c را به یک زبان ساخته شده برای کار با پشته تبدیل می‌کنند. سپس در یک مرحله اضافی، این زبان میانی (واسطه) را به کدهای معمول RISC تبدیل می‌کنند. این ار کامپایلر را قادر می‌سازد که جریان داده را بررسی کرده و ضروری‌ترین اجزا را در ثباتهای cpu نگهداری کند. یک پردازندهٔ استاندارد RISC یک ساختار پشته‌ای را با ذخیره کردن یا بارکردن داده از پشته، به داخل ثباتها، شبیه سازی می‌کند. سپس یکی از ثباتها را بعنوان اشاره‌گر پشته، استفاده می‌کند. این روند ساده است. ولی تعداد ثباتها، امکان بهینه سازی را محدود می‌کند.

معماری picoJava، از یک پشته با ۶۴ ثبات ۳۲ بیتی به‌همراه یک اشاره‌گر به ثبات بالایی پشته، استفاده می‌کند.(شکل یک را ببینید.) اگر شما ۲۰ ثبات داشته باشید که برای یک قالب پشتهٔ مشخص اختصاص داده شده‌اند (مثلاً متد A)، در اینصورت فراخوانی یک متد دیگر مانند B ثباتهای ۲۱ به بعد را مورد استفادهٔ قرار می‌دهد و ثبات اشاره‌گر به بالای پشته، از ثبات ۲۰ به آخرین ثبات استفاده شده توسط متد B انتقال خواهد یافت.

حافظه نهانی (CACHE) باهوش

معماران (طراحان) Sun روش هوشمندی از نهانسازی داده را در حالتی که تمامی ثباتها پر هستند ابداع کرده‌اند، (شکل دو را ببینید) برای نمونه هنگامیکه متد B را فراخوانی می‌کنید، فایل ثباتهای picoJava، تمامی ثباتهای خالی باقیمانده را تخصیص می‌دهد و اگر حافظه اضافی بیش از ۶۴ نیاز باشد، مابقی را به ثبات ۱ منتقل می‌کند. چه اتفاقی برای داده‌های متد A، موجود در آن ثباتها رخ خواهد داد، اگر اجرای متد B متوقف شود و متد A بخواهدمجدداً اجرا شود؟ در اینجا چیزی که Sun آنرا dribbler می نامد، از پس زمینه وارد عمل شده وداده های متد A را جایگزین می‌کند. dribbler پیوسته داده‌ها را از ۶۴ ثبات خوانده و در یک نمونه که در حافظه نگهداری می‌شود، می‌نویسد. بنابراین هنگامیکه متد B ثباتهای اضافی را اشغال می‌کند، dribbler داده‌ها را نمونه برداری کرده است. (اگر بنا به دلایلی dribbler هنوز نمونه را تهیه نکرده باشد، تراشهٔ جاوا کلیه کارهای پردازش را متوقف می‌سازد تا زمانیکه dribbler عملیاتش را خاتمه دهد). هنگامیکه متد B، اجرا را متوقف کرده و ثباتها را آزاد می‌کند، dribbler داده‌ها را در پشته جایگزین می‌کند و از آن پس متد A در جریان خواهد بود. dribbler از این واقعیت استفاده می‌کند که رفت و برگشت داده بین ثباتها و مدلشان در حافظه، به راحتی قابل پیش بینی است. طراحان سیستم قادرند به راحتی یک حافظه نهانی را به گونه‌ای سازگار کنند که خواسته‌های dirbbler را پیش بینی کرده و اطمینان حاصل کند که در هنگام نیاز، داده‌های لازم در حافظه نهانی موقتی داده، در دسترس است.

دستیابی به ثباتهای انعطاف پذیر آشکارترین فرق picoJava در برابر دستهٔ ثباتهای سادهٔ پردازنده‌های RISC است. dribbler جاوا بطور پویا می‌کوشد که همه متغیرهای موقتی را در درون ثباتهای سریع در دسترس نگاه دارد. از طرف دیگر تراشه‌های RISC برای مطلوب نگهداشتن حرکت اطلاعات به داخل یا خارج تراشه، کامپایلر متکی هستند. تخصیص ثبات بصورت ایستا (static) با کدهای علمی بخوبی کار می‌کند، در حالیکه ممکن است حلقه‌های پیچیده‌ای داشته باشد که هر قسمت

از داده‌ها را در محاسبات مضاعف بکار می‌برد.

یک کامپایلر قوی ممکن است راهی برای باز کردن حلقه‌ها و سامان دادن به جریان ورود و خروج داده به ثبات‌ها، پیدا کند. همان کامپایلر ممکن است قادر باشد داده‌ای را - در مواقعیکه نیاز به استفاده مجدد از آن در ۵۰ سیکل بعدتر است - در یک ثبات باقی گذارد. پشته picoJava برای باقی گذاردن داده‌ها یا برای انباشتن عمیق اطلاعات در پشته - به منظور استفاده مجدد در زمان لازم - مناسب نیست.

هر چند برجستگی picoJava در هنگامی است که یک برنامه تعداد زیادی پردازه، که پیوسته شروع به کار کرده و متوقف می‌شوند، را فراخوانی می‌کند. اینگونه فراخوانی‌های توابع بطور پیوسته ثباتها را پاک کرده و دوباره از داده پر می‌کنند. پشته جاوا این خرده‌کاریها را در پس زمینه به کمک dribbler انجام می‌دهد.

وجود پشته در مرکز ماشین مجازی جاوا،افتخار آمیز است زیرا درست کردن کدبرنامه را آسان می‌کند. این طراحی، ماشین‌های RISC و توانایی آنها، برای سرعت بخشیدن به جریان داده بوسیله استفاده از ثباتها در حالت هوشمند، را به مبارزه می‌طلبد. یک مفسر جاوا نمی‌تواند جریان داده را در پشته پیش بینی کند، به همین دلیل نمی‌تواند از ثباتها به عنوان چیزی بیشتر از یک نمونه موقتی از پشته استفاده کند. کامپایلرهای Just-in-time ممکن است بتوانند تجزیه و تحلیل‌های لازم را برای استفاده کارآمدتر از ثباتها انجام دهند، ولی صرف زمان برای اینگونه تجزیه و تحلیلها باعث از بین رفتن کارآیی خود آنها خواهد شد.

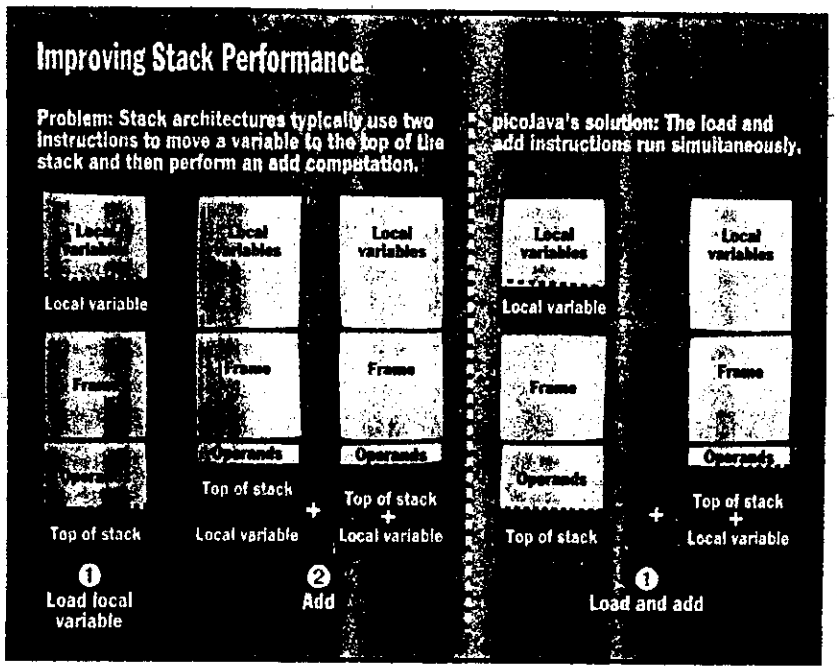
کارآیی پشته

کارآیی مهم دیگر معماری picoJava این است که این ساختار می‌تواند هر زمان که نیاز به انتقال متغیری محلی - برای انجام محاسبات بر روی آن- به بالای پشته است، دستورالعملهایی را بطور همزمان ارسال کند(شکل سه را ببینید). اگر دستورالعملها همزمان ارسال نشوند، داده‌ها درست پس از نوشته شدن در بالای پشته، مورد استفاده قرار می‌گیرند. picoJava عمل انتقال و عملیات حسابی را با هم صادر می‌کند. بنابراین هر دو در یک زمان اجرا می‌شوند. بدون اینکه پشته را مخدوش کند یا از ثباتی استفاده کند و یا از dribbler کاری بخواهد. اینکار باعث کاهش زمان دستیابی به اطلاعات شده و خود به خود زمان اجرا را حذف می‌کند.

گزارش‌های اولیه Sun بیانگر این مطلب بود که تأثیر دستورالعملهای همزمان بسیار جالب خواهد بود. با توجه به تجزیه و تحلیل کدهای Sun، عملیات مربوط به پشته، ۴۳ درصد کل عملیاتی راکه یک تراشه جاوا انجام می‌دهد، شامل می‌شود. اگر دستورالعملها را ترکیب کنید، این میزان به ۲۹ درصد کاهش خواهد یافت.

هدف همیشگی در طراحی همه cpuها، چگونگی اداره جریان داده‌ها، درون سیستم است. یک پردازشگر جدید RISC دارای دو سطح از حافظه نهانی است که داده‌ها را به داخل یا خارج حافظه اصلی، انتقال می‌دهند. حافظه اصلی نیز به نوبت خود، همانند یک حافظه نهانی، برای مقدار زیادی حافظه مجازی برروی دیسک سخت، عمل می‌کند. معمولاً این ترکیب برای هرچه نزدیکتر نگهداشتن اطلاعات مورد نیاز، به cpu به کار می‌رود. با این فرض که آخرین اطلاعات استفاده شده، بیشترین احتمال را برای استفاده مجدد دارند.

جمع آوری آشغالها (garbage collection) تکنیکی که طی آن پردازشگر تمام اشیاء را امتحان کرده و معین می‌کند که کدامیک مورد استفاده نیست، ممکن است باعث تخریب این رویه شود. این جستجوی خسته کننده می‌تواند تمام زحماتی را که حافظه نهانی و کنترلگر حافظه مجازی، برای نزدیک نگهداشتن اطلاعات جاری و مهم به cpu کشیده‌اند، ضایع کند. و در نتیجه تمام اشیاء بعنوان آخرین اطلاعات منظور خواهند

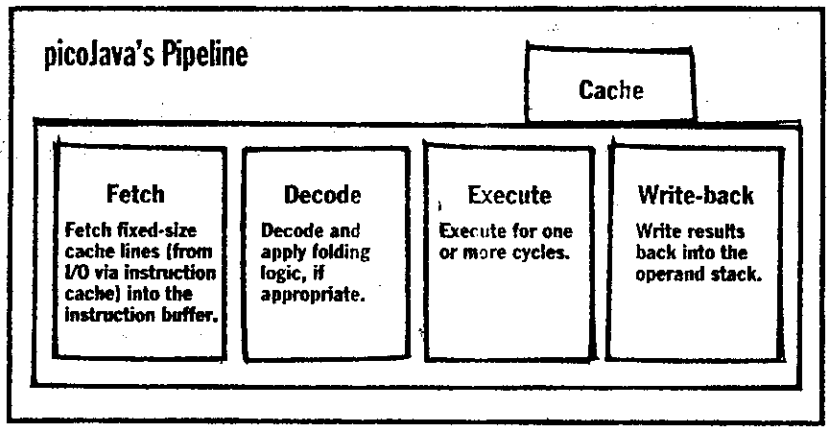


A picoJava chip moves data to the top of the unused registers in the stack and simultaneously dispatches a computation instruction.

شد. این موضوع می‌تواند مشکلی اساسی باشد، اگر گردآورندهٔ آشغالها کار خود را همانند یک کار فرعی بطور همزمان انجام دهد.

ساده ترین راه حل این است که نرم افزار را مجاز کنیم تا بتواند بخشهایی از حافظه نهانی را روشن یا خاموش کند. این کار به مدیریت پشته کمک می‌کند، زیرا احتمال استفاده از اطلاعات بالایی پشته در پردازش بعدی، بسیار بیشتر از اطلاعات انتهایی پشته است بسیاری از تراشه‌های RISC از این روش کنترل حافظه نهانی استفاده می‌کنند.

مشکل بزرگتری که وجود دارد، اینست که حتی ساده‌ترین روشهای گردآوری آشغال، نمی‌توانند توسط کارهای معمول سیستم، دچار وقفه شوند. اگر گردآوری آشغال دچار وقفه شود، در اینصورت ممکن است مجموعه حافظه‌های با مرجع و بدون مرجع، ضایع شده و اطلاعات مفید دور ریخته شود. برای محافظت در برابر چنین مشکلی برای هر شیء از یک بیت برچسب که «مانع چاپ» (write barrier) نامیده می‌شود، استفاده می‌شود. این مانع، امکان آنرا فراهم می‌کند که گردآوری اطلاعات در پس زمینه اجرا شود و عملاً باعث حذف تأثیراتی می‌شود که ممکن است بر اجرای برنامه‌ها- در زمانیکه کل دستگاه برای مشخص کردن حافظه‌های بدون مرجع، مکث کرده است - داشته باشد.



To get data in the right place at exactly the right time, picoJava uses a simple, RISC-like pipeline with only four stages.

Pipeline رشته‌ای(StreamLined)

برای داشتن حداکثر کار آیی، هر طراحی پردازنده مرکزی باید قدرت محاسباتی هر دستورالعمل را بسنجد تا بگونه مفیدی بتواند اطلاعات را Pipeline کند. فرآیند Pipeline کردن، یک دستورالعمل را به قسمتهای مختلفی تقسیم می‌کند، که همه آنها دارای زمان اجرای برابر هستند. این خاصیت سبب می‌شود که پردازنده‌های فوق سنجشی (Superscalar cpu) بتوانند چندین دستورالعمل را بطور همزمان اجرا کنند.

برای اینکه Pipeline بتواند کار کند، لازمست که کلیه اطلاعات لازم برای محاسبه، در زمان مناسب در جای مناسب موجود باشد. RISCها توسط کامپایرهای مکمل (بهینه ساز)، اینکار را بطرز مطلوبی انجام می‌دهند، و Sun روشی بسیار شبیه به پردازنده‌های RISC، در جاوا استفاده می‌کنند. Pipeline تنها ۴ مرحله دارد: واکشی، رمزگشایی، اجرا و باز نویسی(شکل چهار را ببینید).

تراشه در زمان اجرا، به حافظهٔ نهانی رجوع می‌کند که ممکن است باعث انجام عملیات اضافه شود. برای نمونه، در بعضی از دستورات جاوا لازم است که شما با اضافه کردن n بایت به اشاره‌گر آغازین یک شیء،به یک فیلد آن دسترسی پیدا کنید. اینگونه دستورات جاوا در فرآیند Pipeline Pico-Java همانند یک دستورالعمل واحد اجرا می‌شوند.

آیابه تراشه های جاوا نیازمندیم؟

قابلیت عظیم جاوا ،تحرکی در صنعت کامپیوتر برانگیخته است.با این وجود،همه معتقد نیستندکه تراشه های جاوا ضروری‌اند.بعضی از فروشندگان تراشه،ادعا می‌کنند که معماری RISC و CISC فعلی ،میتواند به خوبی از عهده جاوا بر آید.

ماشینهای RISC پیشرفته ،دارای معماری ARM قوی تغییر یافته‌ای، برای برنامه‌های جایگزین و زبانهای ساخته شده برای کار با پشته، همانند جاوا و PostScript می باشند.ساختار قوی ARM می‌تواند یک قالب پشته را تنها با یک دستورالعملِ واحد،به داخل یا خارج ثباتها منتقل کند.

مانع دیگر Sun ،مشکلات پیش بینی نشده‌ای می تواند باشد که از بکارگیری تراشه های picoJava در سیستم ،ممکن است بروز کند. در نهایت، موفقیت تراشه های جاوا تا حد زیادی وابسته به موفقیت خود جاوا خواهد بود.

اگر جنبه های استقلال و امنیت جاوا، باعث شود توسعه دهندگان کامپیوتر جاوا را بپذیرند،انگاه کاربران کاملاً از سیستمهای مختص جاوا ،راضی خواهند شد.اما اگر تسلط برنامه های کاربردیِ نوشته شده با کدهای عادی ،بر بازار ادامه داشته باشد،تنها مورد استفاده تراشه های خاص جاوا،در دستگاههای جایگزینِ کم قدرت خواهد بود.

علیرضا ماندگار

تعریفهای پیچیده اشاره‌گرها در زبان C

محمدرضا خجسته

بعضی مواقع تعاریف اشاره‌گرها (declarations) می‌تواند بسیار پیچیده شود، طوری که به سختی بتوان آن تعریف را معنا کرد. به مثالهای زیر دقت کنید: (مثالها به‌ترتیب درجه پیچیدگی مرتب شده‌اند.)

```
char **argv
    argv : Pointer to pointer to char
int (*daytab)[13]
    daytab : Pointer to array[13] of int
int *daytab[13]
    daytab : array[13] of pointer to int
void *comp()
    comp : function returning pointer to void
void (*comp)()
    comp : pointer to function returning void
char (*(*x())[])()
    x : function returning pointer to array[] of pointer to function returning char
char (*(*x[3])())[5]
    x : array[3] of pointer to function returning pointer to array[5] of char
```

در این نوشتار، برنامه‌ای ارائه می‌دهیم که کار بالا را برای ما انجام دهد! این برنامه (که آنرا dcl می‌نامیم) بر اساس گرامر زیر بنانهاده می‌شود:

```
dcl : Optional  *'s  direct-dcl
direct-dcl :name
                    (dcl)
                    direct-dcl()
                    direct-dcl[optional size]
```

این گرامر می‌تواند برای parse کردن تعریفها بکار رود. قلب برنامهٔ dcl از دو تابع dcl و dirdcl تشکیل شده‌است که در زیر ارائه می‌شوند. از آنجایی که گرامر به صورت بازگشتی تعریف شده‌است، تابعها نیز همدیگر را بصورت بازگشتی call می‌کنند. دنبال کردن طرز کار این برنامه آموزشی، نمونهٔ ساده‌ای برای یادگیری کارکرد recursive-descent parser ها در کامپایلرها می‌باشد.

```
/* dcl : parse a declarator */
void dcl(void)
{
    int ns;
    for (ns=0 ; gettoken()=='*'; )    /* count *'s */
        ns++;
    dirdcl();
    while(ns-->0)
        strcat(out,"pointer to");
}

/* dirdcl: parse a dirdct declarator */
void dirdcl(void)
{
    int type;
    if(tokentype=='('){    /* (dcl) */
        dcl();
        if(tokentype!=')')
            printf("error : missing )\n");
        } else  if (tokentype==NAME)    /variable name */
                strcpy(name,token);
        else
            printf("error:expected name or (dcl)\n");
```

```
        while ((type=gettoken())==PARENS || type==BRACKETS)
            if (type==PARENS)
                strcat (out, "function returning");
            else {
                strcat (out, "array ");
                strcat (out, token);
                strcat (out, "of");
            }
}
```

البته باید توجه داشت، حالتهایی که این برنامه درنظر می‌گیرد کلی نیستند. (مثلاً تنها **int** و **char** را می‌شناسد و ...) بنابراین تعریفهای نادرست، این **parser** را دچار دردسر خواهند کرد!

متغیرهای سراسری (**global**) و تابع **()main** در زیر آورده‌شده‌اند:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define MAXTOKEN 100
enum { NAME, PARENS, BRACKETS};
void dcl(void);
void dirdcl(void);
int gettoken(void);
int tokentype;                  /*type of last token*/
char token[MAXTOKEN];           /*last token string*/
char name[MAXTOKEN];            /*identifier name*/
char datatype[MAXTOKEN];        /*data type=char,int,etc.*/
char out[1000];                 /*output string*/

main()                          /*convert declarations to words*/
{
    while (gettoken()!=EOF) {   /*1st token on line*/
        strcpy(datatype,token);     /*is the data type*/
        out[0]='\0';
        dcl();                      /*parse rest of line*/
        if(tokentype!='\n')
            printf("syntax error \n");
        printf("%s:%s %s\n", name, out, datatype);
    }
    return 0;
}
```

تابع **getoken**، **lablank** و **tab**ها را رد کرده و **token** بعدی را در ورودی پیدا می‌کند که این **token** می‌تواند یک اسم، یک جفت پرانتز، یک جفت کروشه -که ممکن است داخل آنها عددی هم باشد- و یا هر کاراکتر دیگری (بصورت تکی) باشد. (بعنوان تمرین می‌توانید این تابع را هماهنگ با پروسه‌های بالا بنویسید).

تمرین: برنامه‌ای (یا تابعی) بنویسید که عکس کار بالا را انجام دهد، بعنوان مثال:

```
ورودی :  x () * [] * () char
         /* x is a function returning a pointer to an array
         of pointer to function returning char */
خروجی :  char ( *(*x( ) ) [ ] )( )
```

مرجع

1- The C Programming language
Brian W. Kernighan & Dennis M. Fitchie
Second edition

بهبود کارایی ویندوز ۹۵

شما هم می‌توانید با اعمال تغییراتی در سیستم عامل ویندوز۹۵ کارایی آن را بالا ببرید
برای بالا بردن کارایی، دادن تغییرات زیر که در کتابهای راهنمای ویندوز۹۵ نیامده، توصیه می‌شود.

۱) مطابق شکل ۶، typical role of this machine (نوع استفاده از دستگاه) مشخص‌کنندهٔ فضایی است که در حافظه اصلی (RAM)به حافظه نهانی (cache memory) اختصاص داده می‌شود.

دلیل وجود حافظه نهانی چیست؟

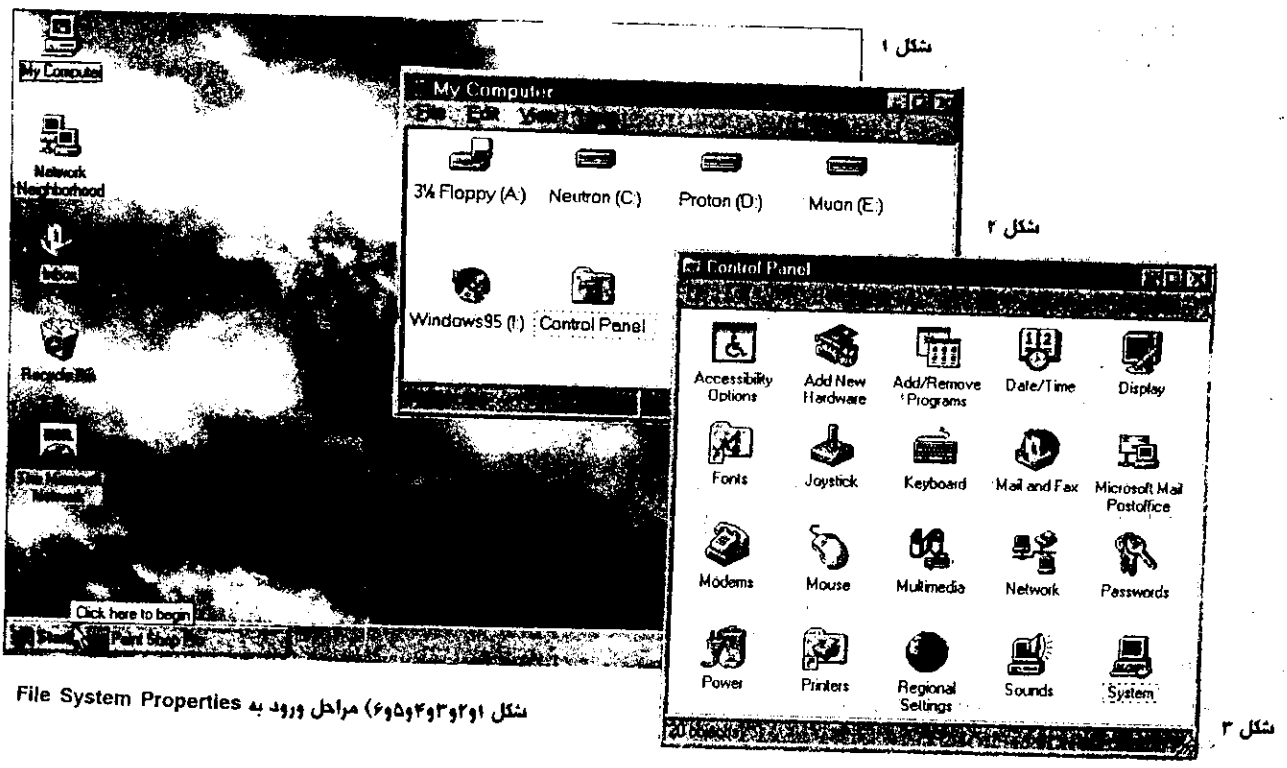
می‌دانیم که دسترسی به اطلاعات روی دیسک (CD،دیسک‌سخت) به کندی صورت می‌گیرد. برای بالا بردن سرعت دسترسی، مقداری از فضای حافظه به حافظه نهانی اختصاص داده شده است.

زمانی که کار با فایلی تمام شد، اگر فایل اجرایی باشد مسیر(path) آن و در غیر این صورت تمام فایل آن درون کاشه (حافظه نهانی) کپی می‌شود.با استفاده از این اصل که: فایلی را که با آن کار کردیم در زمانی نه چندان دور هم، به آن نیاز خواهیم داشت. برای دسترسی مجدد، چون نام فایل یا مسیر آن درون کاشه کپی شده، دستیابی کامپیوتر برای کپی آن در RAM و اجرای آن سریعتر خواهد شد.

حداکثر فضایی که به کاشه اختصاص داده می‌شود متغیر است. با توجه به شکل (۶) اگر typical role of this machine را روی Desktop computer قرار دهیم، ۳۲ مسیر یا ۶۷۷نام فایل آخر و اگر روی Network Server قرار دهیم، ۶۴ مسیر یا ۲۷۲۹ نام فایل‌آخر درون کاشه نگهداری خواهدشد.

برای استفاده از کاشه بیشتر، حتی اگر کامپیوتر ما Server نباشد، حالت Desktop computer را به Network Server تغییر می‌دهیم. نگران کم شدن فضای حافظه اصلی نباشید چون باانجام این کار سرعت دسترسی مجدد به فایل بالا رفته است.

برای اعمال این تغییرات اگر از نسخه جدید ویندوز۹۵ (OEM Server Release 2 = OSR2) استفاده می‌کنید اشکالی پیش نمی‌آید. ولی اگر از نسخه اصلی ویندوز۹۵ استفاده می‌کنید، registry (فهرست) آن را به صورت زیر



شکل ۱

شکل ۲

شکل ۳

شکل ۱و۲و۳و۴و۵و۶) مراحل ورود به File System Properties

تغییردهید :

HKEY-LOCAL-MACHINE\SOFTWARE\Microsoft

\Windows\currentversion\FSTemplates\Server

و نیزنام کاشه را به 00 00 .00 0a a9 و مسیرکاشه را به 00 00 00 40.

۲) مطابق شکل (۶) Read_ahead optimazition (بهینه سازی یا از پیش خواندن اطلاعات روی دیسک سخت) مشخص کننده آن مقدار اطلاعات اضافی است که ویندوز۹۵ باید بعد از هر مراجعه به دیسک، درون بافر کپی کند.

به چه منظور و با چه ترتیبی اطلاعات درون بافر کپی می‌شوند؟

معمولاً اطلاعات مورد نیاز ما به صورت متوالی روی دیسک سخت قرار دارند به همین دلیل زمانی که برای خواندن یک سری از اطلاعات به دیسک سخت مراجعه شود، ویندوز۹۵ یک سری اطلاعات بعد از آن را به صورت اضافی درون بافر که در حافظه اصلی قرار دارد، کپی میکند. با قرار دادن این اطلاعات در بافر تعداد دفعات مراجعه به دیسک سخت کم شده و سرعت دستیابی به اطلاعات بالا میرود. برای رسیدن به این هدف، با توجه به شکل (۶) شاخص Read-ahead Optimization را همیشه روی full قرار دهید. بدین ترتیب ۶۴Kb برای از پیش‌خوانی به بافر اختصاص داده می‌شود.

۱) مقدار مکمل کاشه (Supplemental cache size): مطابق شکل (۷) با تغییر جای شاخص می‌توان مقدار بافر تحت اختیار read-ahead (از پیش خواندن) CD را تغییر داد.

سه مرحله اول شاخص، ۶۴Kb یا ۱۲۸Kb و سه مرحله آخر هر کدام ۲۵۶Kb که جمعاً باعث افزایش ۱۰۸۸Kb فضای حافظه به کاشه می‌شوند.

۲) بهینه سازی الگوی دسترسی (Optimize access patern for):

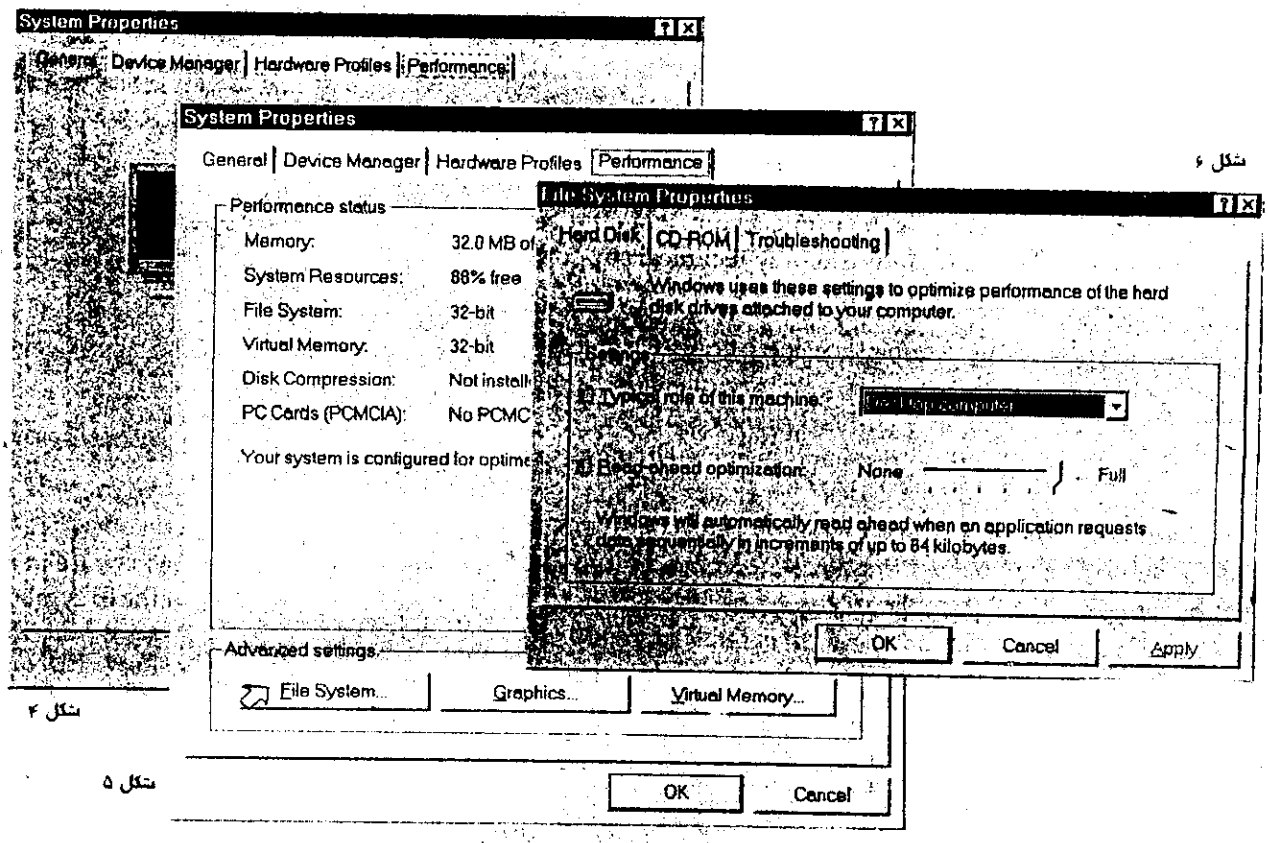
نباید تصور کرد با اعمال تغییرات در این جعبه سرعت دسترسی به اطلاعات روی CD افزایش پیدا می‌کند. بلکه ویندوز۹۵ با اختصاص دادن فضای بیشتری به بافر، حجم اطلاعات ذخیره شده در بافر را افزایش می‌دهد. بدین ترتیب به جای رجوع به CD به بافر مراجعه می‌کند.

اگر حالت No Read-ahead یا single speed را انتخاب کنیم بافر تغییر نمیکند. واگر حالتهای Double-speed یا Triple-speed یا Gaud-speed or higher را انتخاب کرده باشیم، ۵۰Kb، ۱۰۰و۱۵۰ به بافر افزوده می‌شود(جدول ۱).

حتی اگر شما از CD استفاده مستمر نمی‌کنید، شاخص Supplemental cache size را روی Larg و حالت Gaud-speed or Higher را برای Optimize access pattern for انتخاب کنید.

حافظهٔ مجازی (*virtual memory*) چیست؟

فرض کنید برنامه‌ای نیاز به ۸Mb حافظه دارد و کامپیوتر ما



شکل ۴

شکل ۵

شکل ۶

فقط ۴Mb حافظه دارد. چون ما در یک لحظه به کل برنامه نیاز نداریم، آن را روی دیسک نگهداری می‌کنیم و با استفاده از روش Paging، برنامه را به pageهای مختلف تقسیم می‌کنیم. و در صورت نیاز به هر page، آن page به حافظه اصلی که از پیش تقسیم بندی شده، منتقل می‌شود. با پر شدن ۴Mb حافظه، برای انتقال page مورد نظر ما، یک page از حافظهٔ اصلی روی دیسک قرار می‌گیرد و page مورد نظر ما به حافظه اصلی منتقل می‌شود. عمل انتقال page از حافظه اصلی به دیسک از اولین pageکه در حافظهٔ اصلی قرار گرفته، شروع می‌شود.

حسن این کار در این است که حافظه بیشتر از آن چیزی که هست بنظر می‌رسد و اشکال آن در این است که رجوع کردن به دیسک و انتقال page به حافظه اصلی سرعت اجرا را پایین می‌آورد.

با توجه به شکل (۹) با اصلاحاتی در حافظهٔ مجازی، می‌توان سرعت اجرای برنامه های کاربردی که نیاز به حافظهٔ زیاد دارند مثل Adobe photoshop یا Microsoft Office 97 را بالا برد.

اگر کامپیوتر شما به بیش از یک گرداننده دیسک سخت مجهز است، سریعترین گرداننده را برای جعبهٔ دیسک سخت انتخاب کنید، سپس آن را Defragment کنید.

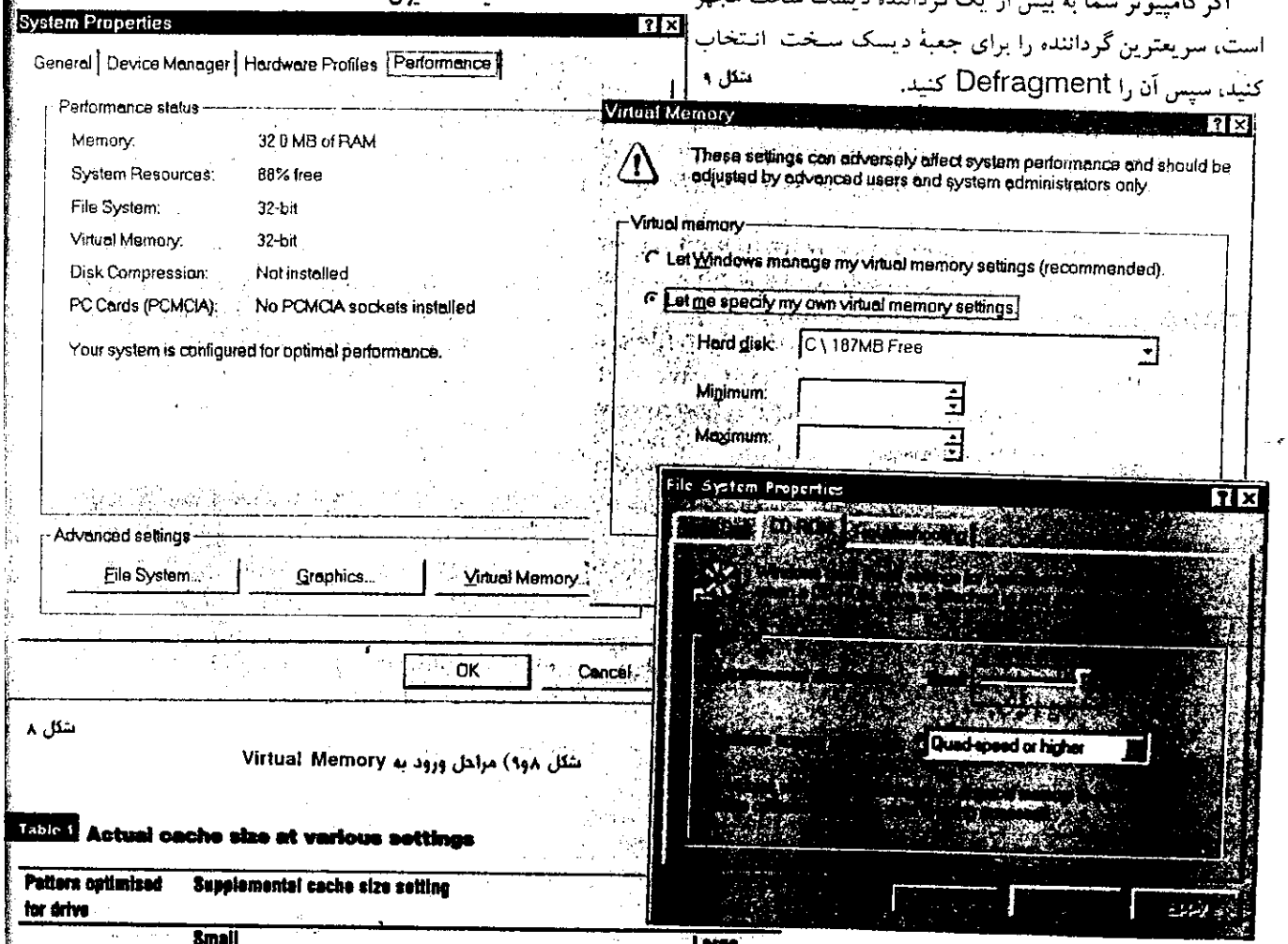
Defragment یعنی چه؟

بین اطلاعات یک فایل روی دیسک سخت فضای خالی وجود دارد. به این حالت اصطلاحاً fragement (تکه‌تکه) می‌گویند که این فضای خالی باعث پائین آمدن سرعت خواندن اطلاعات می‌شود، چون بازوی هد هنگام رسیدن به فضای خالی بلند شده و تا رسیدن به اطلاعات بعدی فضای خالی را رد می‌کند. برای از بین بردن فضای خالی باید دیسک سخت را Defragement کنیم. برای کم کردن زمان جابجایی اطلاعات بین حافظهٔ اصلی و دیسک، عددی که برای مینیمم و ماکزیمم انتخاب می‌کنید مساوی و تقریباً ۲/۵ برابر حافظهٔ RAM باشد. مثلاً برای ۱۶Kb حافظه می‌توان عدد ۴۸ را انتخاب کرد. سپس OKرا انتخاب کرده و ویندوز۹۵ را Restart کنید.

با تشکر از جناب آقای مهندس دست‌پاک

تهیه کنندگان: مهسا صابر

لیلا شمشیری‌خامنه



شکل ۸

شکل ۹و۸) مراحل ورود به Virtual Memory

شکل ۷

Table 1 Actual cache size at various settings

Pattern optimized for drive	Supplemental cache size setting						
	Small						Large
No read-ahead	64K	128K	192K	320K	576K	832K	1,088K
Single-speed	64K	128K	192K	320K	576K	832K	1,088K
Double-speed	114K	178K	242K	370K	626K	882K	1,138K
Triple-speed	164K	228K	292K	420K	676K	932K	1,188K
Quad-speed	214K	278K	342K	470K	726K	982K	1,238K

جدول۱

آینده

ذخیره و بازیابی داده‌ها

داستان ذخیره سازی در کامپیوتر از زمان اولین کامپیوترهای الکترونیکی در نیمه قرن حاضر، نقل محافل علمی بوده و بطور پیوسته ادامه یافته و درنتیجه باعث انقباض ابعاد و کاهش قیمتها گردیده‌است. حول و حوش سال ۱۹۸۴، سخت دیسک بیست مگابایتی کامپیوتر بسیار ابتدایی IBM برای تمام کاربرها ایده‌ال به نظر می‌رسید. امروزه ظرفیت رومیزی معمول نزدیک به ۱۰۰ برابر شده است: ۲ گیگا بایت. با این نرخ رشد، در مدت زمانی حدود ده دوازده سال یک سیستم رومیزی نمونه دویست گیگابایت حافظهٔ ذخیره شده را عرضه خواهد کرد. هر چند که ابزار ذخیره سازی ۲۰۰ گیگابایتی نسلی از دیسک گردانهای مغناطیسی وینچستری (Winchester) خواهد بود که در این صورت برای هر بحثی در این باره، عرصه فراخ است. حتی زمانیکه سخت دیسکها با محدودیتهایشان، در ذخیره سازی برخورد می‌کنند، حقیقتی که احتمالاً در ۱۰ سال آینده اتفاق می‌افتد - زمینه‌های بسیار زیادی جهت پیشرفت در فن آوریهای دیگر وجود خواهد داشت. برخی از این فن‌آوریها از قبیل ذخیره سازی هولوگرافیک (Holographic) هنوز در سطح آزمایشگاههای تحقیقاتی هستند لیکن به نظر می‌رسد که با سرعت بالایی در حال گسترش و توسعه می‌باشند. انواع دیگر مانند ذخیرهٔ داده در یک الگوی منحصر به فرد اتمها (با سیله حرکت اتمها با یک میکروسکوپ Scaning Tunneling) از راه حل‌هایی که ضرورتاً ارائه شده‌اند قابل فهم‌تر است از طرفی بسیاری از طرحهایی که اکنون محتمل و قابل اعتماد به نظر می‌رسند مطمئناً از رده خارج خواهندشد و این در طی گذشت زمان به ما ثابت می‌شود ولی عده‌ای دیگر حتماً نتیجه‌بخش خواهندبود. آنچه که اکنون می‌توان ابراز کرد این است که به نظر می‌رسد آیندهٔ ذخیره سازی بسیار وسیع بوده و سرانجامی خوش در انتظار آن است.

دستیابی به ظرفیت بالاتر بر پایهٔ توانایی جای دادن بیتهای بیشتر و بیشتر روی هر سطح داده شده (تحت مقیاسی شناخته شده همچون چگالی سطحی) استوار است. امّا برای مقدار اطلاعاتی که شما می‌توانید در هر اینچ مربع جاسازی کنید، یک سری محدودیتهای فیزیکی وجود دارد و این بدان معناست که دستیابی به ظرفیتهای بالاتر می‌باید در نقطه‌ای متوقف شود. ولی با این اوصاف ما هنوز به حدود آن نقطه نرسیده‌ایم.

سخت دیسک از چهل سال پیش که IBM مدل RAMAC*۳۰۵ را معرفی کرد، راهی طولانی را طی کرده است RAMAC۳۰۵ که از پلاترهای ۲۴ اینچی (به تعداد ۵۰ عدد) استفاده می‌کرد، تقریباً حجمی به اندازهٔ دو یخچال اشغال می‌کرد و ۵ مگابایت حافظه ذخیره شده را ارائه می‌داد. (۱۰۰۰۰ دلار برای هر مگابایت) در حالیکه چگالی سطحی آن تنها ۲ کیلو بایت یا ۲۵۶ بایت در هر اینچ مربع بود، زمانی که اولین کامپیوتر شخصی IBM در سال ۱۹۸۱ به بازار آمد (در حالیکه فاقد سخت دیسک بود که از الزامات استاندارد است) شما می‌توانستید دیسک گردانهای ۵ مگابایتی را مشاهده کنید که حجمی به ارتفاع ۵/۲۵ اینچ را اشغال کرده بودند، (۱۰۰۰ دلار برای هر مگابایت) بدین ترتیب چگالی سطحی به ۲ مگابایت در هر اینچ افزایش یافته بود. امروزه چگالی سطحی تا ۵۰۰ مگابایت در هر اینچ بهبود یافته است و بعضی از کامپیوترهای شخصی به طور تقریبی تا ۱/۴ گیگابایت در هر اینچ مربع را در اختیارمان قرار می‌دهند (یعنی ۱۷۵ مگابایت حافظه روی یک تمبر پستی) و قیمت آن حدود ۱۰ الی ۱۵ سنت برای هر مگابایت تمام می‌شود. اولین و مهمترین اقدام جهت بهبود بخشیدن چگالی در طول ۵ سال آینده، افزایش حساسیت هِد خواننده (Read Head) می‌باشد. دراین راستا اخیراً بواسطهٔ حرکت به سمت هدهای MR یا هدهای Magneto Resistive پیشرفتهای قابل ملاحظه‌ای در زمینهٔ افزایش چگالی سطحی به دست آمده‌است. هدهای MR از هدهای لایه نازک القایی (Thin Film Inductive) حساسترند و به خاطر نوظهور بودنشان جولانگاه وسیعی جهت پیشرفت و تکامل در اختیار دارند. برای مثال شرکت IBM اعلام کرده که با استفاده از فن‌آوری هد MR به چگالی ۵ گیگابایت در هر اینچ مربع دست یافته است. شرکت مزبور امیدوار است که ظرف مدّت کمی (حدود ۲ سال) ظرفیت درایوها را حدود ۳ تا ۵ برابر مقدار مذکور افزایش دهد و سرانجام در سال ۲۰۰۲ به ظرفیتی حدود ۱۰ برابر دست یابد. همچنین IBM امیدوار است تا سال ۲۰۰۰ قیمت عرضه به مشتری به ۳ سنت (برای هر مگابایت) کاهش یابد. علاوه‌بر این زمانیکه هدهای MR به واسطه

* Random Access Method of Accounting And Control

محدودیتهایشان از پیشرفت باز بمانند، در این حال زمینهٔ پیشرفت و تکمیل هدهای حساستری که به هدهای spin-valve معروفند، فراهم خواهدشد. هدهای spin-valve وابسته به اثر حساسیت مغناطیسی بسیار شدید (giant magnetoresistive) می‌باشند که برای افزایش مقاومت و به تبع آن ایجاد حساسیتی بزرگتر، تعداد زیادی از لایه‌های thin-film را مورد استفاده قرار می‌دهند. به احتمال قوی هدهای spin-valve برای تولید چگالی‌های بزرگتر از ۵ گیگابایت در هر اینچ مربع، و افزایش چگالی سخت دیسکها به حداقل ۱۰ گیگابایت در هر اینچ مربع جایگزین هدهای MR خواهند شد. مزیت دیگر افزایش چگالی سطحی، علاوه بر توانایی افزایش حافظه تعبیه شده در کامپیوتر، افزایش سرعت خواندن داده‌هاست زیرا هنگامی که بیتها فشرده و بهم نزدیک شوند، بیتهای بیشتری در واحد زمان از زیر هد عبور میکنند و هد بسیار سریعتر از گذشته داده‌ها را می‌خوانند. در واقع یک دیسک ۴۵۰۰rpm با یک چگالی سطحی بالا، سریعتر از یک دیسک ۵۴۰۰rpm با چگالی سطحی پائینتر عمل می‌کند با این وجود ما هنوز به یک سرعت چرخشی قابل قبول دست نیافته‌ایم. در حال حاضر اکثر درایوها با سرعتی حدود ۴۵۰۰rpm یا ۵۴۰۰rpm می‌چرخند و سریعترین آنها با سرعت ۷۲۰۰rpm چرخش می‌کند. (البته امسال اولین درایوهای ۱۰۰۰۰rpm را نیز خواهیم دید.) همچنین علاوه بر درایوها، واسطها نیز باید به سرعت بالاتری دست یابند. خط سیر SCSI نشان می‌دهد که فن‌آوری در این زمینه در حال پیشرفت است.

این روزها یکی از موضوعهای داغ و بحث برانگیز در محافل مهندسی سخت افزار، **محدودیت نهایی** ظرفیت سخت دیسکها می‌باشد. هنگامی که بیتها (سطوح مغناطیسی شدهٔ روی دیسک) کوچک و کوچکتر شوند، عاقبت به نقطه‌ای خواهیم‌رسید که انرژی مورد نیاز برای از بین‌بردن خاصیت مغناطیسی با انرژی گرمائی محیط برابر شود. به بیان دیگر دمای یک اتاق برای از بین بردن میدانهای مغناطیسیِ روی دیسک و در نتیجه از بین رفتن اطلاعات، کافی باشد. چنین محدودیتی، محدودیت فوق مغناطیسی (super paramagnetic) نامیده‌می‌شود. محدودیت فوق مغناطیسی زمانی اتفاق می‌افتد که ما به یک چگالی سطحی بین ۵ و ۱۰ گیگا بایت در هر اینچ مربع دست یافته باشیم؛ میزانی که به احتمال قوی ظرف ده سال آینده محقق خواهد شد. دو نوع تفکر در رابطه با مقابله با این محدودیت وجود دارد. تفکر اول معتقد است که محدودیت فوق مغناطیسی خط پایانی است برای دیسکهای مغناطیسی. تفکر دوم قائل به این است که ما قادریم زمان فرا رسیدن این محدودیت را بوسیله مواد جدید و ساختارهای جدید دیسکی، تمدید کنیم. به عنوان مثال ساختاری که تمام حوزه‌های مغناطیسی آن از حوزه‌های همسایه جدا (ایزوله) شده است، حوزه‌های مغناطیسی‌اش از ثبات و قوام بیشتری برخوردارند. بنابراین باید بسیار امیدوار بود که بتوان محدودیت فوق مغناطیسی دیسکها را به تعویق انداخت.

دیسکهای نوری:
مسیرهای چندگانه پیشرفت و تکامل

آینده دیسکهای نوری بسیار با ثبات و امیدوار کننده به نظر می‌رسد. در این رابطه، بزرگترین نکتهٔ قابل ذکر DVD می‌باشد که قطعاً بر تمام فعالیتهای چند سال آینده سایه افکنده است. تا جایی که انواع مختلف و متعدد آن جایگزین درایوهای CD-ROM و CD-R در کامپیوترهای شخصی خواهد شد. (برای کسب اطلاعات بیشتر در مورد DVD می‌توانید مقالهٔ دیسکهای DVD پویش شماره ۲ و یا مقالهٔ 21 st-century storage مجلهٔ PC، ۲۱ ژانویه ۹۷ را مطالعه کنید)

انتظار می‌رود دیسکهای DVD-RAM در سال ۱۹۹۸ ظرفیتی معادل ۲/۶ گیگا بایت را در اختیار ما قرار بدهند. سری اول DVD-RAM از فن آوری تغییر فاز بهره می‌برد، که این فن آوری یک فن آوری صددرصد مطمئن نیست و نمی‌توان در سریهای بعدی هم از آن استفاده کرد، اما برخلاف آن، فن‌آوری نوری-مغناطیسی یک جایگزین مناسب در مسیر تکامل DVDهاست؛ علاوه بر این ممکن است یک مشخصهٔ گمنام به نام MO7 نیز برای مقابله با فن آوری تغییر فاز به پا خیزد. همان گونه که از نام این مشخصه برداشت می‌شود درایوهای MO7 از فن آوری MO با ظرفیتی حدود ۶ تا ۷ گیگابایت در هر پلاتر استفاده می‌کنند. اما بر خلاف دیسکهای MO که از فرمت دیسکهای ۵/۲۵ اینچ استفاده می‌کنند، درایوهای MO7 از دیسکهای ۱۲۰ میلی متری (مانند CDها و DVDها) استفاده می‌کنند. بنابراین MO7 را می‌توان به شیوه‌ای طراحی کرد که قادر به خواندن DVD نیز باشد. علیرغم توضیحات بالا هنوز این سؤال قابل طرح است که آیا سری بعدی تولیدات DVD-RAM از فن‌آوری تغییر فاز استفاده می‌کند یا MO؟ جواب قویاً وابسته به پاسخ این سؤال است که کدام فن آوری، چگالی سطحی را سریعتر افزایش می‌دهد؟ یکی از روشهای رسیدن به چگالی بیشتر، استفاده از لایه‌های چند گانه است. در این روش استفاده از فن‌آوری تغییر فاز سودمندتر است و تها حل یک مسئله باقی می‌ماند و آن اینکه چگونه لیزر را هدایت کنیم تا فقط بر لایهٔ مورد نظرمان تأثیر بگذارد؛ حال آنکه درایوهای MO علاوه بر کنترل لیزر به کنترل میدان مغناطیسی نیز محتاجند زیرا تواناییهای یک درایو MO در خواندن و نوشتن به اتکاء هر دو (لیزر و میدان مغناطیسی) می‌باشد.

اما اخیراً شرکت سونی تکنیکی را ارائه کرده است که در آن برای درایوهای MO این امکان فراهم می‌شود که بدون استفاده از لایه‌های اضافی، در بیتهای کوچکتری بنویسند تا به این ترتیب چگالی سطحی افزایش یابد. بنابراین با توجه به آنچه گفته شد و نیز دلایل عدیدهٔ دیگر به این نتیجه می‌رسیم که فعلاً نمی‌توان درایوهای MO را از رده خارج فرض کرد.

علاوه بر تلاشی که در راستای تکمیل دیسکهای ۱۲ سانتیمتری صورت می‌گیرد، تلاشی موازی نیز برای بکارگیری

دیسکهای ۳/۵ اینچ یا کوچکتر در حال شکل گیری است و به این خاطر که دیسکهای CD جهت حمل و نقل آسان بسیار بزرگ هستند، فن آوری DVD ممکن است در یک اندازهٔ کوچکتر دیسکی به رشد و تکامل ادامه دهد.

با نگاهی اجمالی به ۱۰ تا ۱۵ سال آینده، متوجه می‌شویم که تکامل در زمینهٔ چگالی سطحی تا حد زیادی به پیشرفت ما در علوم لیزری وابسته است. لیزرهای با طول موج کوتاه‌تر جهت تبدیل بیتها به نقاط کوچکتر بر روی دیسکها بسیار توانا هستند. طراحی لیزرهای با طول موج کوتاه گسترهٔ وسیعی است که اکنون زمان پرداختن به آن فرا رسیده است.

نوار: بهترین در راه است

مدت مدیدی است که نوار برای پشتیبان گرفتن از نرم افزارها، مشهورترین انتخاب بوده‌است؛ زیرا در بین ابزار ذخیره سازی، هر مگابایت یک نوار کمترین هزینه را به خوداختصاص می‌دهد. اما طبیعت ترتیبی نوار و در نتیجه قدرت مانور ضعیفش، آنرا برای ذخیره‌سازی پیشرفته، انتخابی ضعیف قرارداده‌است.

از آنجاییکه نوارگردان جهت بارگذاری یک فایل باید نوار را تا ابتدای فایل بچرخاند، بارگذاری فایل از روی نوار چند ثانیه طول می‌کشد در حالیکه بارگذاری از روی سخت دیسک چند هزارم ثانیه بیشتر به طول نمی‌انجامد؛ با این وجود ذخیره سازی فایلها برخلاف بارگذاری، به سرعت انجام می‌پذیرد زیرا ابتدا نرم‌افزار از روی RAM بر سخت دیسک انتقال یافته آنگاه به نوار منتقل میشود. در چند سال آینده اقدامات جالبی دراین‌باره صورت‌خواهد پذیرفت، مثلاً طرحهایی در دست‌اقدام هستند که نوار را به یک وسیلهٔ دسترسی تصادفی تبدیل می‌کنند، با این امید که سرعت نوار را در نوشتن و بازیابی فایلها به اندازهٔ سرعت دیسکها بالا ببرند. مانند طرحهای Tape-It از شرکت PGSoft و نوار دسترسی مستقیم از شرکت Seagat (که دومی را مرور خواهیم کرد). دو طرح مذکور باعث می‌شوند که نوارگردان برای سیستم عامل، مانند یک دیسک گردان عمل‌کند.

هم اکنون طرحهایی تحت عنوان نوار هوشمند در حال شکل‌گیری‌است. برای مثال شرکت اگزابایت در حال کار کردن برروی روشی است که امکان دسترسی آنی به فایلها را از روی یک نوار گردان فراهم می‌سازد. (حداقل برای فایلهای بزرگ مانند فایلهای تصویری و آوا و نماها «Video clips») در این طرح سطوح مرجعی (Reference) بر روی نوار به طور خود به خود بوجود می‌آید. این سطوح مرجع شامل تعداد زیادی فریم از یک فایل تصویری می‌باشند و زمانیکه سیستم اولین فریمها را اجرا می‌کند فرصتی پدید می‌آید تا نوار به فایل مورد نظر دسترسی پیدا کند. علاوه‌براین شرکتهای دیگر مشغول کار بر روی طرحهایی هستند که از نوار جهت ذخیره سازی پیشرفته و از سخت دیسک جهت Caching استفاده کنند.

عمده موفقیت نوار در سالهایی که پیش رو داریم به سرعت اجرا وابسته است و در این صورت است که‌می‌توان داده‌ها را روی یک‌نوار با سرعتی پردازش کرد که قابل رقابت با سرعت دیسکها باشد. در راستای این هدف، جهت دستیابی به نوار گردانهای ارزان با سرعت انتقال بالاتر و اعتماد بیشتر می‌باید راهی به سوی استفاده از هدهای MR و شیارهای Servo (شیارهای مخصوصی که به درایو کمک می‌کنند تا شیارهای داده را پیدا کند) و هدهای هشت کاناله (بجای هدهای یک کانالهٔ فعلی) و کانالهای دیجیتالی (بجای کانالهای آنالوگ) پیداکرد.

در ظرف ۵ تا ۱۰ سال آینده، دستیابی به هدهای ۱۶ شیاری یک پیشرفت قابل پیش بینی است. درعین‌حال فن آوری حال حاضر لیزر در مورد درایوهای نوری نیز ممکن است گوشهٔ چشمی به نوار بیاندازد و منجر به تولید نوارهای نوری شود. در اینباره مدیر کل شرکت اگزابایت بر این باور است که ظرف ۵ تا ۱۰ سال آینده به نوارهای نوری دست خواهیم یافت؛ اما عده‌ای نیز معتقدند مدت زمان ۱۰ سال بسیار خوش بینانه‌است زیرا یک نوار نوری می‌باید هم ظرفیت و هم سرعت را در حد مطلوبی تکامل بخشد تا قابل قیاس با نوار مغناطیسی باشد. مدیراگزابایت همچنین اضافه می‌کند که ممکن است نحوه ذخیره داده روی نوار نوری با ذخیره آن روی دیسک، متمایز و متفاوت باشد (به سبب ایجاد سطوح مرجع مضاعف در طول نوار نوری). درست‌است که چنین نگرشی به نوار زیاده از حد ایده‌ال می‌باشد لیکن بیانگر نگرشی نوین بر پدیده‌ای است که در حال آمدن است.

فن آوریهای پدیداری
Emerging Technologie

مشهورترین این فن‌آوریها، ذخیره‌سازی هولوگرافیک می‌باشد که به نظر می‌رسد در شرف تحقق است. هولوگرافی می‌تواند دنیا را کاملاً متحول سازد. مبانی این نوع جدید ذخیره‌سازی بسیار ساده است: ابتدا توسط ابزاری به نام تنظیم‌کنندهٔ فضایی‌نور، صفحه‌ای می‌سازیم که اطلاعات رقمی را به شکل یک نمودار دو بعدی نمایش بدهد. بعد از آن یک هولوگرام از نمودار را بوجود می‌آوریم. ایجاد هولوگرام دو مرحلهٔ شکاف نوری اشعهٔ لیزر و سپس هدایت آن توسط آینه‌ها را شامل می‌شود. یک اشعه که اشعهٔ شیئی نامیده می‌شود برای تولید الگوی متداخل و ضبط یک هولوگرام، قبل از پیوستن به اشعه دیگری که اشعه مرجع نام دارد صفحه را روشن می‌سازد. برای خواندن داده‌ها عکس عملیات فوق صورت می‌پذیرد. یعنی ابتدا اشعهٔ مرجع را باز آفرینی می‌کنیم و آنرا بر روی هولوگرام می‌تابانیم تا نمودار صفحه باز تولید شود.

ذخیره سازی هولوگرافیک انواع مزایا را در بر دارد. هولوگرافها نه مانند تصاویر بلکه تحت الگوهای متداخل ذخیره می‌شوند و به این خاطر پس از تقسیم یک هولوگرام، دو نیمهٔ یک نمودار را در دست نداریم بلکه دو نمودار کامل در دسترسمان قرا می‌گیرد و این بدان معناست که در صورت بروز اشکال در ابزار ذخیره سازی هیچ داده‌ای پاک نخواهد شد زیرا اطلاعات متکثراً در سرتاسر هولوگرام ذخیره شده است. هنچنین می‌توان با تغییر طول موج یا زاویهٔ لیزر هولوگرامهای متعددی را در مکان یکسان ذخیره‌کرد. از

آنجاییکه هر صفحه می‌تواند بیشتر از یک میلیون بیت را در خود ذخیره کند و می‌توان (در حیطهٔ نظر) هزاران صفحه را در یک حجم چند بعدی مشخص ذخیره کرد، چگالی حجمی نهفته (نزدیکترین معادل چگالی سطحی) چیزی حدود یک تریلیون‌تزا بیت تخمین زده می‌شود و از آنجاکه در ذخیره سازی هولوگرافیک، در زمان واحد کل یک صفحه را می‌خوانیم، بنابراین سرعت اجرا شدیداً افزایش می‌یابد.

علیرغم همهٔ آنچه که گفته شد، راه حل مناسب ذخیره سازی در ۱۰ سال آینده هنوز ناشناخته مانده است. این راه حل هر کدام از امکاناتی که توضیح داده شد می‌تواندباشد یا اینکه می‌تواند چیزی متفاوت با تمام راه‌حلهای مذکور در این نوشتار باشد. مدیر اگزابایت معتقد است که هنوز فن‌آوری خاصی که آیندهٔ حتمی ذخیره سازی درسال ۲۰۱۰ را تشریح کند، بوجود نیامده است. ممکن است حق با او باشد!

ذخیره سازی توده‌ای پدیداری : حرکت سریع با ابعاد کوچک

با توجه به گذشتهٔ ذخیره سازی توده‌ای در می‌یابیم که در آینده نیز انواع متنوعی از فن آوریها ظهور خواهند کرد و به موازات یکدیگر به تکامل خواهند رسید. در این قسمت سعی ما بر این است که به نوار و سخت دیسکهای تحت این فن آوری نگاهی اجمالی بیفکنیم:

DTA:نرم‌افزار پرقدرت پشتیبان

به تازگی نوعی نرم افزار نوار گردان جدید به بازار آمده است بانام دسترسی نواری مستقیم سی‌گیت (نسخهٔ ۲/۰۵) که جدیدترین و دومین برنامه از نوع خودش می‌باشد. مهمترین نکتهٔ این نرم افزار که به DTA موسوم است، همانند PGSoft's Tape-It این است که نوارگردان شما را همچون یک دیسک گردان به ویندوز مایکروسافت مجهز می‌سازد و دیگر مجبور نیستید که از نرم افزار پشتیبانی جهت ذخیره سازی یا بازیابی فایلها استفاده کنید. هم اکنون DTA برای ویندوز ۳/۱ و ویندوز ۹۵ موجود است و نیز با درایوهای شرکتهای Hewlett-Packard، Iomega و Tandberg سازگار می‌باشد.

در حال حاضر ممکن است چند ثانیه و یا حتی دهها ثانیه زمان تلف شود تا DTA یک فایل را جهت بارگذاری آن و یا مکان صحیحی را جهت شروع ذخیره‌سازی یک فایل پیدا کند. با این وجود جای نگرانی وجود ندارد زیرا این اتلاف وقت تنها زمانی اتفاق می‌افتد که ما در صدد بارگذاری یک فایل بزرگ باشیم.

پیش از این در نظر ما استفاده از یک نوار گردان هم سطح یک دیسک گردان تعجب آور بود و اندکی شبیه رقص یک خرس می‌نمود و این تعجب نه بواسطهٔ خوب رقصیدن خرس که به این دلیل بود که خرس اصلاً نمی‌رقصد؛ اما در آزمایشات با نوار گردان تراوان - ۳، ابزار DTA خیلی سریعتر وسلیس‌تر از آنچه که انتظار می‌رفت عمل کرد. بنابراین احتمال می‌رود DTA یا چیزی شبیه آن، در اکثر نوارگردانها جایگزین برنامه‌های اصلاح کننده وپشتیبانی کننده شود و یا آنها را تکمیل کند.

بلند پروازی تصویری

ابزار Enagle Digimax (از شرکت نرم‌افزاری اگزابایت) نوع جدیدی از نوار گردانهای ۸ میلی متری می‌باشد. این نوار گردان روی هر کارتریج نواری ۱۳ گیگابایت حافظهٔ محلی ایجاد میکند. ظرفیت بالا، اجرای سریع و قیمت پایین برای هر مگا بسایت، DigiMax را به یک مدعی جهت کاربردهای بایگانی-تصویری (که هم اکنون توسط ابزارهای ذخیره سازی نوری انجام می‌شود) تبدیل کرده‌است.

همچنین DigiMax یک درایو جابجایی پذیر است که با یک کیت همراه با آشیانه (Nest) ارائه می‌شود. آشیانه را باید روی پی[1] یک درایو ۵/۲۵ اینچ نصب کرد. درایو را می‌توان داخل یا خارج از پی قرار داد تا بتوان آنرا از یک سیستم به سیستم دیگر انتقال داد. شرکت اگزابایت آیندهٔ واقعی این درایو را در ذخیره سازی تصویری متصور می‌داند. این شرکت معتقد است که **ظرف چند سال آینده شاهد پست الکترونیکی تصویری خواهیم بود**؛ اگر این پیش‌بینی اتفاق بیفتد نیازهای ذخیره سازی به حد اعلای خود می‌رسد و در این حال DigiMax بعنوان یک درایو ظرفیت بالای ارزان قیمت (که توانایی ذخیره سازی تصویری روی یک نوار و یا اجرای تصویری از یک نوار را داراست) آمادهٔ ارائه خدمات خواهدبود. محتوای تصویری که توسط ما(مؤلفین) مورد بررسی قرار گرفت تقریباً هیچ نقصی نداشت و کمترین شکست بافتی (نارسایی ساختاری) درآن مشاهده نشد. اگزابایت همچنین امید زیادی به تولید انواع دیگری از DigiMax دارد.

احتمالاً DigiMax بعنوان یک VCR برای تلویزیون یا پدیدهٔ نوظهور PC/TV و یا فایلهای دریافت شده از اینترنت روی یک web TV و حتی فیلمهای بصری دریافت شده، به خدمت گرفته‌خواهدشد.

با توجه به توانایی بالای ذخیره سازی تصویری DigiMax و بازیابی سهل الوصول تصویر و مهمتر از همه اقتصادی بودن آن، مشخص می‌شود گزارشهایی که خبر از مرگ نوار می‌دهند دور از واقعیت هستند.

ترجمه: علیرضا بذرافشان

LOOKING FORWARD:

Mass Storage

PC magazine;

23 march 1997

۱- Bay

قسمت دوم:

زاد و ولد پردازه‌ها

علی حاجی‌زاده

در قسمت قبل، نحوهٔ برخورد یونیکس با فایلها و فراخوانهای سیستم مربوط به کار با دیلفه و نیز شیوهٔ تغییر مسیر ورودی-خروجی را در C تحت یونیکس توضیح دادیم. در این قسمت به روش زادوولد پردازه‌ها و ارتباط آنها با هم می‌پردازیم

پردازه چیست؟

به هر چیز درحال اجرا در یونیکس یک پردازه (Process) گفته می‌شود. این "چیز" می‌تواند یک برنامهٔ C باشد که توسط کاربر نوشته و اجرا شده و یا یکی از فرمانهای خود یونیکس. توجه کنید که یک برنامه یا دستور تنها تا زمانی پردازه نامیده می‌شود که در حال اجرا باشد. مثلاً اگر شما در شاخه‌تان یک فایل a.out دارید، تا زمانی که آنرا اجرا نکرده‌اید به آن پردازه نمی‌گویند. همینطور پس از اتمام اجرای آن.

یونیکس به هرکدام از پردازه‌های جاری یک شماره اختصاص می‌دهد که عددی صحیح و یکتاست و شمارهٔ پردازه یا Process ID (به اختصار PID) نامیده‌می‌شوند. شما با استفاده‌از دستور ps می‌توانید لیست پردازه‌های در حال اجرای مربوط به خودتان را ببینید. اولین ستون خروجی این دستور PID پردازه‌ها است. برای قطع اجرای یک پردازه از فرمان kill بهمراه PID پردازهٔ مربوطه استفاده می‌شود. به این ترتیب، اگر شما برنامه‌ای را اجرا کرده‌اید و نمی‌توانید از آن خارج شوید یا به اجرای آن خاتمه دهید، کافی است که از طریق یکی دیگر از ترمینالها login کرده دستور ps - x را اجرا کنید. این دستور فهرست همه پردازه‌های جاری و PID را به شما می‌دهد. حال با دستور kill-9 <PID> می‌توانید به اجرای پردازهٔ مورد نظرتان خاتمه دهید.

تکثیر پردازه‌ها

زاد و ولد پردازه‌ها شبیه تکثیر باکتری‌هاست. همان طور که یک باکتری تقسیم شده دو باکتری مشابه ایجاد می‌شود، یک پردازه هم بوسیله فراخوان سیستمی به نام fork یک کپی از خودش ایجاد می‌کند. استفاده از فراخوان fork تنها راه ایجاد یک پردازه جدید در یونیکس است. در برنامه‌نویسی به زبان C تحت یونیکس هم تابع fork() این فراخوان سیستم را صدا می‌زند. شکل کلی صدا زدن این فراخوان به صورت مقابل است: pid = fork(); که در آن pid عددی صحیح بوده، محتوی شماره پردازه جدید است.

بعنوان مثال برنامهٔ زیر را کامپایل و اجرا کنید:

```
#include <stdio.h>
main( )
{ printf(" begin \n");
  fork( );
  printf(" End \n");
}
```

پس از اجرای این برنامه ابتدا عبارت begin چاپ می‌شود. سپس فراخوان ( )fork یک کپی از پردازه ایجاد می‌کند. بنابراین از این لحظه به بعد مانند آن است که برنامهٔ فوق دوبار اجرا شده‌باشد. بنابراین دوبار عبارت End چاپ می‌شود. نمودار یک مراحل این کاررا نشان می‌دهد. توجه کنید که هیچ پارامتری به این فراخوان سیستم فرستاده نمی‌شود. گفتیم که فراخوان ( )fork پس از اجرا یک کپی کامل از پردازهٔ در حال اجرا که او را صدازده ایجاد می‌کند بطوری که دو نسخهٔ یکسان از برنامه وجود خواهد داشت که هردو به اجراشدن ادامه‌می‌دهند. پردازهٔ صدازنندهٔ ( )fork از این پس پدر و پردازهٔ ایجاد شده فرزند نامیده می‌شود. تنها تفاوت بین این دوپردازه در مقداری است که ( )fork به آنها برمی‌گرداند. در پردازهٔ فرزند، فراخوان ( )fork مقدار صفر رابر می‌گرداند. در پردازهٔ پدر این مقدار شمارهٔ پردازهٔ فرزند است که همراه مقداری غیر صفر است. (همان شماره پردازه‌ای که در ستون اول از خروجی فرمان ps مشاهده می‌شود). به این ترتیب، پردازه‌های پدر و فرزند می‌توانند تشخیص دهند که کدامیکی هستند (پدر یا فرزند) و بر این اساس اعمال متفاوتی انجام دهند:

```
if( (pid = fork() ) != 0 )
      printf(" I am Parent \n");
else printf(" I am Child \n");
```

گفتگوی پردازه‌های پدر و فرزند

بسیار خوب! حالا دو پردازه داریم که یکی فرزند دیگری است

و هر دو همزمان در حال اجرا هستند. ولی چطور می‌توان داده‌ها را بین آنها رد و بدل کرد؟ به عبارت دیگر اگر بخواهیم هر دو پردازه با هم کاری را انجام دهند و برای اینکار احتیاج باشد (مثلاً) خروجی پردازهٔ پدر، توسط پردازهٔ فرزند دریافت شود و برروی آن محاسبات جدیدی انجام گیرد، چه باید بکنیم؟ پاسخ، استفاده از یک لوله است.

لوله (Pipe) چیست؟

یک لوله، مسیری است که داده‌ها از یک طرف به آن وارد شده از طرف دیگر آن خارج می‌گردند. لوله در واقع یک میانگیر (بافر) از نوع صف (fifo) است. یعنی اولین مورد از داده‌ها که در آن نوشته‌شود، اولین موردی خواهد بود که از طرف دیگر آن خوانده می‌شود. پس باید هر لوله دو نقطهٔ دسترسی داشته باشد: یکی برای نوشتن اطلاعات و دیگری برای خواندن از آن. برای این منظور، به‌هر لوله در هنگام ایجاد آن، دو توصیف‌گرفایل (file descriptor) اختصاص داده می‌شود. (توصیف‌گر فایل بطور کامل در قسمت قبلی این مقاله توضیح داده شده‌است. رجوع کنید به: پویش۲ / صفحهٔ۲۳ / چندکارگی در یونیکس)

برای ایجاد یک لوله از فراخوان سیستم ( )pipe به شکل روبرو استفاده می‌شود:

```
i=pipe(fd);
```

که در آن fd باید بصورت آرایه‌ای دو عضوی از اعداد صحیح تعریف شده باشد:

```
int fd[2];
```

دو توصیف‌گر فایلی که ذکر کردیم، در دو خانهٔ این آرایه قرار می‌گیرند. یعنی پس از اجرای فراخوان ( )pipe، fd[0] محتوی توصیف‌گر فایل مخصوص خواندن از لوله‌است و fd[1] محتوی توصیف‌گر فایل مخصوص نوشتن در آن. خود ( )pipe هم مقداری برمی‌گرداند که در صورت موفقیت عمل صفر و در غیراینصورت 1- است.

برای آنکه هردو پردازهٔ پدر و فرزند به یک لوله دسترسی داشته‌باشند، باید قبل از صدازدن فراخوان ( )fork (یعنی قبل از ایجاد پردازهٔ فرزند) لوله ایجادشده باشد. در اینصورت، چون فرزند تمام خصوصیات پدرش را به ارث می‌برد، به توصیف‌گرهای فایل لوله هم دسترسی خواهد داشت. یعنی اگر پردازهٔ پدر یک لوله درست کند و سپس یک فراخوان ( )fork را صدا بزند، فرزند هم کاملاً به لوله دسترسی دارد. پس تنها کاری که باقی می‌ماند تا دوپردازهٔ پدر و فرزند بتوانند باهم گفتگو کنند، آن است که (مثلاً) پدر اطلاعات را در لوله بنویسد (یا خروجی استانداردش را به توصیف‌گر فایل fd[1] تغییر مسیر دهد) و فرزند از لوله بخواند (یا ورودی استانداردش را به fd[0] تغییر دهد). در قسمت قبل نحوهٔ تغییر مسیر ورودی و خروجی استاندارد را بیان کردیم. تکه برنامه‌های زیر این کار را انجام می‌دهند:

در پردازهٔ پدر:

```
fdtmp=dup(1);
close(1);
dup(fd[1]);
close(fd[1]);
....
close(1);
dup(fdtmp);
close(fdtmp);
```

در پردازهٔ فرزند:

```
fdtmp=dup(0);
close(0);
dup(fd[0]);
close(fd[0]);
....
close(0);
dup(fdtmp);
close(fdtmp);
```

از این شیوه می‌توان برای ارتباط دو پردازهٔ پدر و فرزند (و یا هر دو پردازهٔ دیگری که اجداد مشترکی داشته‌باشند) استفاده کرد. ولی برای ایجاد ارتباط بین پردازه‌هایی که هیچ نسبتی با هم ندارند (و یا جد مشترکشان قبلاً یک لوله ایجاد نکرده که‌هردوی آنها آنرا بشناسند) چه‌باید کرد؟

لولهٔ بانام

یک لولهٔ بانام شکل خاصی از فایل است که خصوصیات صف (fifo) را دارد. بطوری که یک پردازه می‌تواند از آن بخواند و دیگری بر آن بنویسد. در واقع وقتی یک لولهٔ بانام ایجاد می‌کنیم، یک فایل جدید (در دایرکتوری جاری) ایجاد کرده‌ایم که اطلاعات نوشته شده در آن در صف قرار می‌گیرند و اطلاعات خوانده‌شده از آن پاک می‌شوند. فراخوان سیستمی که برای ایجاد یک لولهٔ بانام بکار می‌رود mknod( ) است:

```
mknod(filename,filetype);
```

اولین پارامتر مطابق معمول اشاره‌گری به یک رشته حاوی نام فایل است. پارامتر دوم یک عدد شش رقمی است که سه رقم سمت چپ آن برای ایجاد لوله باید 010 باشند و سه رقم بعدی مجوزهای خواندن و نوشتن در لوله برای صاحب آن، گروه ودیگران است. (برای توضیحات بیشتر درمورد بیتهای مجوز به راهنمای فرمان chmod یونیکس مراجعه کنید.) مثلاً mknod("PIPE",010666); یک فایل لوله بنام PIPE در دایرکتوری جاری ایجاد می‌کند که موقع لیست‌گرفتن با فرمان ls به‌این‌شکل ظاهر می‌شود:

```
ls -l PIPE
prw-rw-rw- 1 pc      pc      0 Apr 8 16:08 PIPE
```

به اولین کاراکتر خروجی دقت کنید. این فیلد برای فایلها "-" و برای دایرکتوری‌ها "d" است و همانطور که می‌بینید برای لوله‌ها این فیلد "p" خواهد بود. اکنون هر دو پردازهٔ مستقلی می‌توانند این‌لوله را مانند یک فایل معمولی (با استفاده از توابع fopen یا open) باز کنند و برروی آن بنویسند یا از آن بخوانند و یا آنکه ورودی/خروجی استانداردشان را به‌طرف آن تغییر مسیر دهند.

اجرای فرمانها

ممکن است گاهی بخواهید پردازه‌ای که با فراخوان ( )fork ایجاد شده یک برنامهٔ دیگر یا یکی از فرمانهای سیستم یونیکس را اجرا کنید. بعنوان مثال شاید بخواهید پردازهٔ فرزند، فهرست فایلهای دایرکتوری جاری را نشان‌دهد (فرمان ls- ls). برای اینکار از فراخوان سیستمی بنام exec استفاده می‌شود. این فراخوان نسخه‌های مختلفی دارد که هرکدام امکانات متفاوتی در اختیار برنامه‌نویس قرار می‌دهند. نسخه‌ای که معمولاً استفاده می‌شود ( )execlp نام دارد. شکل کلی استفاده از این فراخوان به این صورت است:

```
execlp(command,param1,param2,...,NULL);
```

در اینجا همهٔ پارامترها آرایه‌ای کاراکتری (یا اشاره‌گر به یک رشتهٔ کاراکتری) هستند که این رشته‌ها محتوی فرمان مورد نظر و پارامترهای آن است. دقت کنید که فهرست پارامترها باید به یک NULL ختم شود (که NULL=(chat *)0). بعنوان مثال، برای اجرای فرمان ls -la از این دستور استفاده می‌شود:

```
execlp("ls","ls","-la",NULL);
```

فرمان فوق، دستور ls -la را جایگزین پردازهٔ جاری می‌کند.

دقت کنید که می‌گوییم فرمان نام برده‌شده در تابع exec جایگزین پردازهٔ جاری می‌شود، یعنی از این پس، اجرای فرمان فوق (مثلاً ls) جایگزین اجرای برنامهٔ فراخوانندهٔ exec می‌شود. بنابراین هرآنچه بعد از تابع exec بیاید اجرا نخواهدشد. چرا که برنامه پس از رسیدن به exec با برنامهٔ دیگری جایگزین می‌شود. (مگر آنکه فراخوان exec ناموفق عمل کند) همچنین دقت کنید که اولین پارامتر (param1) هم باید نام فرمان باشد. در واقع این پارامترها همان پارامترهای خط فرمان هستند که می‌توان موقع اجرای برنامه‌ها یا فرمانها در جلوی اسم برنامه یا فرمان نوشت و اگر برنامه به‌زبان C نوشته‌شده باشد این پارامترهای خط فرمان بصورت اشاره‌گری به یک آرایه از اشاره‌گرها (که معمولاً بصورت char **argv تعریف می‌شود) به تابع ( )main برنامه فرستاده می‌شود.

یک مثال: شبیه‌سازی تابع ( )system

در اینجا می‌خواهیم تکه‌برنامه‌ای بنویسیم که کار تابع ( )system را انجام دهد. تابع فوق به این شکل به‌کار می‌رود:

```
system(command);
```

که در آن command یک رشته کاراکتر محتوی فرمان مورد نظر است. مثلاً:

```
system("ls -la");
```

برنامه پس از رسیدن به تابع ( )system صبرمی‌کند تا فرمان ذکرشده اجرا شود و سپس به‌کار خودش ادامه‌می‌دهد. برای نوشتن این تکه برنامه باید از فراخوان دیگری به‌نام ( )wait استفاده شود. این فراخوان وقتی پس از فراخوان fork می‌آید باعث می‌شود پردازهٔ پدر تا زمان پایان اجرای پردازهٔ فرزند صبر کند (یا اصطلاحاً به خواب رود). فراخوان wait به این شکل ذاستفاده می‌شود:

```
pid=wait(&status);
```

در اینجا pid شمارهٔ پردازهٔ فرزندی است که اجرایش خاتمه یافته و &status اشاره‌گری به یک متغیر صحیح (int) است که در آن مقدار وضعیت بازگشت (exit status value) فرزند برگردانده می‌شود. (این مقدار در پردازه فرزند بعنوان پارامتر تابع exit مشخص می‌شود. مثلاً exit(9)) به کمک این توابع می‌توان تکه‌برنامه‌ای نوشت که کار فرمان system("ls -la"); را انجام دهد:

```
int pid,status;

pid=fork();
if (pid==0)    {
                    /* I am the child */
                    execlp("ls","ls","-la",(char *)0);
               }
wait(&status);
```

منبع:

Unix & C

Philip Cornes

تصحیح و پوزش

در شماره قبلی پویش در گزارش مربوط به مسابقات برگزار شده در همایش دانشجویی، در مورد مسأله الگوریتم، جواب آقای امید محسنی بعنوان بهترین جواب معرفی شده‌بود که جواب ایشان اشتباه می‌باشد. لذا پویش ضمن تصحیح این مطلب، از این بابت از خوانندگان عزیز پوزش می‌خواهد.

یک سیستم شیءگرا برای ایجاد محیط مناسبات توزیع‌شدهٔ باز

پایان‌نامهٔ کارشناسی ارشد: مازیار چیت‌ساز — استاد راهنما: دکتر ناصر مدیری

امروزه سیستمهای توزیع شده، بخش مهمی از علوم و مهندسی رایانه را به خود اختصاص داده‌اند. رشد شبکه‌ها و قراردادهای ارتباطی، گسترش‌پذیری، اطمینان‌پذیری و اشتراک منابع از مهمترین عوامل رشد این شاخه محسوب می‌شود. توزیع‌شدگی خود به شاخه‌های گوناگونی تقسیم می‌شود که به‌عنوان نمونه می‌توان به هوش‌مصنوعی توزیع‌شده، پایگاه‌داده‌های توزیع‌شده و سیستم‌عملهای توزیع‌شده اشاره کرد.

طبیعی است که برای ایجاد تمامی این‌قبیل سیستمهای توزیع‌شده، وجود خدمات‌اساسی توزیع‌شده لازم‌است. معمولا این‌قبیل خدمات مانند خدمات نام، خدمات زمان، فایل و غیره توسط سیستم‌عامل توزیع‌شده فراهم‌می‌شود. اما کنارگذاردن سیستم‌عاملهای فعلی و طراحی مجدد سیستم‌عاملها، خالی از عیب نیست. صرف زمان و هزینه زیاد، چشم‌پوشی از مهارت فعلی کاربران سیستم‌عاملهای رایج و وجود ابزارهای گوناگون برای سیستم‌عاملهای فعلی، برخی ازاین مشکلات هستند.

به‌جز طراحی و پیاده‌سازی سیستم‌عاملهای توزیع‌شده، گرایش‌دیگری نیز برای ایجاد این‌خدمات‌اساسی وجوددارد و آن استفاده از سیستم‌عاملهای‌موجود برای بوجود آوردن خدمات توزیع‌شده است. دراین گرایش، سعی می‌شود بدون‌تعویض سیستم‌عامل این خدمات فراهم‌آید. یکی‌از اهداف اولیه این‌پروژه نیز ایجادبرخی ازخدمات توزیع‌شدگی بدون تغییر سیستم‌عامل بوده‌است.

بدیهی است که‌برای تولید این‌خدمات، نیازمند مدل‌تحلیل و طراحی هستیم. مزایای زیاد مدل‌شیءگرا از یک‌سو قابلیتهای موجود در این‌مدل برای توزیع‌شدگی از سوی‌دیگر، مارا برآن داشت که‌مدل شیءگرا برای ایجادخدمات‌توزیع‌شده ایجادکنیم. به‌این‌ترتیب هدف‌دوم این‌پروژه، استفاده از مدل شیءگرا و نشان‌دادن قابلیتهای این‌مدل برای کاربردهای توزیع‌شده بوده‌است.

دراین راستا، برای افزایش کیفیت محصول‌نهایی و گسترش‌پذیری آن، از چهارچوب شیءگرا استفاده‌شده‌است. استفاده‌از چهارچوبها برای نرم‌افزارهای‌کاربردی روزبروز افزایش‌یافته و الگوهای طراحی‌آن درنرم‌افزارهای‌سیستمی و به‌خصوص سیستم‌عامل چندان رایج نشده‌است. هدف‌سوم این پروژه نشان‌دادن قابلیت چهارچوبها درطراحی نرم‌افزارهای سیستمی و مخصوصا سیستمهای‌توزیع‌شده بوده‌است.

هدف چهارم این‌پروژه، فراهم‌آوردن طرحی گسترش پذیر است. دراینجا منظور از گسترش‌پذیری، فراهم‌آوردن راه‌کاری برای کاربر است بنحوی که بتواند طراحی موجود را بنا به‌میل خود تغییر داده و یا به امکانات آن بیافزاید. این سیستمها که با صفت باز شناخته‌می‌شوند، براحتی امکان تغییر و گسترش را فراهم می‌آورند، بدون اینکه به طراحی قبلی خدشه‌ای وارد شود.

خدماتی که به عنوان خدمات اساسی انتخاب شده‌اند عبارتند از: خدمات نام، امنیت، زمان و ریسمان. در بخش خدمات نام، بدون اینکه کاربرد خاصی - مانند سیستم فایل - مورد نظر بوده‌باشد، امکانات اصلی شناسایی شده‌اند. منظور از امکانات اصلی، توانایی‌هایی است که در تمامی سیستمهای توزیع‌شده به عنوان توانایی‌های نامگذاری وجوددارد. برای مثال تولید نام، یکی از این نوع امکانات است. ساختار پایگاه داده‌هایی که خدمتکاران نام (Name server) برای نگهداری اطلاعات از آنها استفاده‌می‌کنند، تقسیم و یا تکثیر پایگاه داده‌ها و عملیات‌مجاز بر روی آنها و همچنین تقسیم وظایف بین خدمتکاران نام و نحوه ارتباط آنها نیز ازدیگر امکانات اصلی نامگذاری بشمارمی‌رود.

برای تامین امنیت در بیشتر سیستمهای توزیع‌شده از مدل ماتریس کنترل دستیابی استفاده‌می‌شود. دو روش اصلی برای پیاده‌سازی این مدل وجود دارد: قابلیت و لیست دستیابی. باتوجه به اینکه هریک از این روشها دارای معایب و مزایای مخصوص به خود هستند، در سیستم ما ترکیبی از هر دو روش بکارگرفته شده است تا مزایای آنها با یکدیگر ترکیب شده، تاحد امکان از معایب آنها کاسته‌شود. این روش که قفل/کلید نام گرفته‌است، ترکیبی از قابلیتهای شرطی و لیست کنترل دستیابی به همراه استفاده از رمزبندی است.

نبود ساعت و حافظه مشترک در سیستم‌های توزیع شده، مشکلات فراوانی را بوجود می‌آورند. یکی از این مسایل، همگام‌سازی است. نبود ساعت مشترک، همگام سازی را دشوار می‌کند زیرا ساعت گره‌های مختلف معمولاً زمان یکسانی را نشان نمی‌دهند. به همین دلیل روشی برای تولید ساعت منطقی طراحی شده‌است تا این مشکل بدون استفاده از ساعتهای واقعی حل شود. باوجود این، ساعتهای واقعی که اصطلاحا ساعتهای فیزیکی نامیده می‌شوند، بسیار مفید هستند. نمایش ساعت فیزیکی و تغییر آن، جزء لاینفک تمامی سیستم‌عاملها محسوب می‌شود. بنابراین در سیستم ما اینگونه ساعتها نیز روشهای گوناگونی وجود دارند. برای همگام سازی این ساعتها نیز، وجود دارد که دربین آنها سه روش کلی‌شناسایی شده و رده هایی برای تامین هر سه روش طراحی شده است.

اصولامدیریت پردازه ها جزء سیستم عامل می‌باشد و با توجه به این که یکی از اهداف ما تغییر ندادن سیستم عامل است، نمی‌توانیم در این مورد تغییری ایجاد کنیم. با وجود، این برای یکنواختی چهارچوب خود واسطی شیء گرا طراحی کرده ایم که از دیدکاربر، پردازه ها نیز جزء سیستم تلقی شود، اگرچه تمامی عملیات توسط خود سیستم عامل انجام می‌گیرد. رسیمانها جزء اصلی سیستم عامل نیستند و می‌توان آنها را به سیستم عامل اضافه کرد. بنابر این با ایجاد رده هایی برای فراهم آوردن امکان استفاده از ریسمانها، انواع و اقسام ریسمانها نیز طراحی شده‌اند.

به طور خلاصه اهداف این پایان نامه عبارتند از:

۱. فراهم آوردن خدمات اساسی توزیع شدگی. برای این منظور، خدمات نام، امنیت، زمان و ریسمان انتخاب شده‌اند.

۲. استفاده ازمدل شیءگراو نمایش قابلیتهای آن درکاربردهای توزیع‌شده.

۳. بکاربردن چهارچوب و اثبات مناسب بودن آن برای کاربردهای سیستمی و به خصوص سیستمهای توزیع شده.

۴. باز بودن سیستم طراحی شده به نحوی که کاربر بتواند به راحتی و بدون تغییر طراحی قبلی، تغییرات وگسترشهای لازم را بوجود آورد.

سید جلال

چهل و شش بهار را در آب و خاکمان سپری کرد و در این بازه، اکثر فراز و نشیب‌های اجتماعی را تجربه کرد و به یکی از چهره‌های سترگ ادبی-سیاسی آب و خاکمان مبدل شد. قلم را به خدمت مردم درآورد و کالبدش را روحی و جانی تازه بخشید؛ صبور و بی‌پروا بود؛ هرگز به طفیلی قلم نان نخورد بلکه افسار قلم را پی افکار خویش به هرسویی کشید. به‌دورانی که آب و خاکمان تحت هجوم گردبادهای وحشی و شیطانی، می‌رفت که هیچ شود، او همچون آباء سبزپوش خویش با صلابت و ستیزندگی که در سخن و عملش هویدا بود، همچون سدی بی‌خلل ایستاد و ایستادگی کرد. زندگی‌اش سراسر سعی بود، سعی بین صفای جهل و مروهٔ شناخت و چه هروله‌ای کرد در انجام این سعی. خود می‌گفت: «قلم این‌روزها برای ماشده یک سلاح، بریده‌باد این دست اگر نداند این سلاح را کجا باید به‌کار برد.» و سرانجام نیز این رودخانهٔ پرتلاطم به دریای ابدیت پیوست، مغرور و سربلند.

«بسم‌ا... الرحمن الرحیم. تولد نورچشمی آقا سید جلال ملقب به جلال‌الدین حفظه‌ا... تعالی در لیلهٔ پنج‌شنبه بیست‌ویکم شهر شعبان‌المعظم ۱۳۴۲، تقریباً یک ساعت از شب گذشته مطابق با برج حمل. خداوند قدمش را مبارک نماید. به حق محمد و آله الطاهرین»

در یازدهم آذرماه سال یکهزار و سیصد و دو شمسی در محلهٔ سیدنصرالدین تهران در خانواده‌ای روحانی دیده‌به جهان گشود. پدربزرگ، پدر، برادر بزرگ و برادرزاده و دوتا از شوهرخواهرهای او در کسوت روحانیت بودند.

جلال‌آل‌احمد پس از پایان دوران دبستان به اجبارِ پدر، روانهٔ کار در بازار تهران شد. علت مخالفت پدر وی با ادامهٔ تحصیل او این بود که پدرش می‌خواست برای مسجد و محراب و منبرش جانشینی داشته‌باشد تا پس از وی امورشرعیش را برعهده بگیرد. از این رو با هرگونه تحصیل غیر حوزوی جلال مخالفت کرد.

دکتر امیرحسین آریانپور از همشاگردیهای جلال در دورهٔ دبستان است و از او اینگونه یاد می‌کند: « جلال پیکری باریک و چهره‌ای گیرا داشت، خوش‌بیان و نکته‌گیر و شوخ بود. در خواندن شعر و تلاوت قرآن کسی به گرد او نمی‌رسید... مبصر کلاس بود، از این‌رو تا اندازه‌ای جزو طبقهٔ حاکم مدرسه محسوب می‌شد، با اینهمه با جوش و خروش و شوخی‌ها وشیطنت‌های ظریف خود در دل بچه‌ها رخنه‌می‌کرد... من مانند سایر بچه‌ها او را جلال جنی صدا می‌زدم.» خود جلال مطالب جالبی را دربارهٔ دوران کودکیش و بخصوص تحصیل در دبستان ثریا در داستان **گلبسته‌ها و فلک** آورده است.

جلال در زمانی که در بازار به شاگردی اشتغال داشت، دور از چشم پدر، در دورهٔ شبانهٔ دارالفنون به ادامهٔ تحصیل می‌پرداخت. روزها را به مشاغلی همچون ساعت‌سازی، سیم‌کشی و چرم‌فروشی می‌گذراند و شبها را به ادامهٔ تحصیل؛ تااینکه سرانجام دبیرستان را به پایان رساند و به اخذ دیپلم نایل آمد.

همچنین مدت زمانی دروس طلبگی را نزد سیدهادی طالقانی فراگرفت تااینکه برای ادامهٔ تحصیل دروس حوزوی به نجف‌اشرف سفر نمود. اما پس از گذشت سه‌ماه از راه کرمانشاه و خانقین به ایران بازگشت؛ سرخورده و کلافه از برادر و پدر روگردان شد؛ چراکه به گفتهٔ خودش در آن سفر دامی دیده بود در صورت ردا و عبا. به‌هرحال جلال که با انگشتری عقیق به دست به سفر رفته‌بود، بی‌انگشتری بازگشت و از آن پس دیگر مقید نبود مهر را در هنگام سجده بر پیشانی بگذارد و میلی به شرکت در مجالس روضه و قرائتِ قرآنهای مسجدی نداشت. آری، جلال از همهٔ عقایدش دست شسته‌بود و این در سال ۱۳۲۲ بود.

جلال بیست ساله بود که از مرحلهٔ دین‌اندیشی گذرکرد و دورهٔ جدیدی از زندگیش را آغاز نمود: نواندیشی غیردینی.

مجاهد نستوه مرحوم آیت ا... **طالقانی** درباره علت گرایش جلال به بی‌دینی و سپس مارکسیسم می‌گوید: « یکروز به جلال گفتم این وضعی که برای تو پیش‌آمده که براثر آن به مکاتب دیگر

بخش مهمی از شخصیت جلال و جلالت قدر او همین عبور از گردنه‌ها و فراز و نشیب‌ها و متوقف نماندن او در هیچکدام از آنها بود... آل‌احمد براستی نعمت بزرگی بود. حداقل یک نسل را او آگاهی داده‌است و این برای یک انقلاب، کم‌نیست.
آیت‌الله خامنه‌ای

روی آوردی، نتیجه فشاری است که خانواده بر شما وارد می‌کرد. مثلا اجباراً او را به شاه‌عبدالعظیم می‌بردند تا دعای کمیل بخواند.» بهرحال مجموعه‌ای که فضای تنفسی جلال را شامل می‌شد، وی را به طغیان علیه مربی و حاکم خانه نزدیک کرد که بصورت طغیان علیه باورهای اخلاقی و عقیدتی خانواده بروزکرد. این‌بخش از زندگی جلال سیری است ده ساله از تدین به بی‌دینی آنگاه مارکسیسم، سوسیالیسم و تاحدودی اگزیستانسیالیسم. دوره‌ای که درمسیر تبلور و تکامل اندیشه جلال قرار گرفت و اورا هرچه بهتر به درک حقیقت رهنمون ساخت. نخستین واکنش جلال به این اندیشه، عقاید و اصول جدید، در تأسیس «انجمن اصلاح» متبلور شد. فعالیتهای این انجمن، تدریس زبان فرانسه و عربی، تهیه روزنامه دیواری و وارسی اعمال و اهداف احزاب فراوان و متعدد موجود بود. دراین زمان جلال به چاپ نخستین اثرش (ترجمه کتاب **عزاداریهای نامشروع** تألیف حجه‌الاسلام عاملی) اقدام نمود. کتاب رساله کوچکی درنکوهش زنجیرزنی و شاخ و قفل‌زنی بود که با مخالفت و خشم متعصبان مذهبی روبروشد.

درحقیقت آشنایی جلال باعقاید احمدکسروی و شریعت سنگلجی (که اندیشه‌های نو ولی ضدّ دینی داشتند) و پس از آن انجمن‌اصلاح مقدمه‌ای شد که جلال و همراهانش درسال ۱۳۲۳ به عضویت حزب توده درآیند. این دوران دهساله تا سال ۱۳۳۲ یکی از مهمترین مراحل زندگی جلال است و فعالیتها و آثار زیادی را دراین برهه از زندگانی خود، به‌انجام رسانده‌است. جلال پس از آنکه در سال ۲۳ به عضویت حزب توده درآمد به سرعت مراحل حزبی را طی‌کرد و درکمتر از چهار سال از رهبران حزب توده شد که علت آن به گفته برادرش شمس آل احمد «عربی و فارسی دانستن جلال و جسارت و بی‌پروایی بی‌حدوحصر او بود.» در این زمان جلال مسؤل تبلیغات و انتشارات حزب توده بود و متجاوز از ده حوزهٔ حزبی را اداره می‌کرد و نیز منشی تشکیلات حزب توده بود. در نوروز سال ۱۳۲۴ فعالیتهای مطبوعاتی‌اش را با چاپ نخستین داستان بنام **زیارت** در مجله **سخن** با مدیریت صادق هدایت، آغاز نمود. گرچه آثار وی تهی از خشم و تندی نبود، لیکن اقبال عمومی مردم به آثار وی صاحبنظران را واداشت تا به ارزیابی آثارش بپردازند. سپس در اسفندماه همان سال **دید و بازدید** را منتشر ساخت. در سال ۲۵ مجلهٔ ماهانهٔ **مردم** را راه‌اندازی‌کرد. همچنین مدیر داخلی **روزنامه بشر**، ارگان دانشجویان حزب توده، بود. در سال ۲۹ مدیریت **روزنامهٔ شاهد**، ارگان حزب زحمتکشان، را به عهده گرفت. همچنین سردبیری هفته‌نامه **علم و زندگی** به مدیریت **خلیل ملکی**. در عین حال در گرداندن **روزنامه مهرگان**، ارگان حزب نیروی سوم، نیز نقش مؤثری داشت همچنین است مجله ماهانه **شیر و خورشید سرخ ایران** به مدیریت **ذبیحا... صفا**.

محتوای کلی نوشته‌های جلال در این دوره مرتبط با وقایع سیاسی-اجتماعی و حزبی آن زمان بود. جلال در این مدت، ده عنوان کتاب را ترجمه و تألیف کرد. نخستین تجربه‌اش در کار نویسندگی ترجمه کتاب **عزاداریهای نامشروع** بود و بعد از چاپ **زیارت** اقدام به انتشار مجموعه داستانهای کتاب **دید و بازدید** نمود. مجموعه **دید و بازدید** نوع نگاه جلال به جامعه آنروز است که تصویرهایی سطحی از سنخ‌های مختلف جامعه ترسیم می‌نماید ولی درعین حال درموارد کثیری نیز با خرافه‌هایی که صورت مذهبی به خود گرفته است، به مبارزه برمی‌خیزد. درسال ۲۶ شاهد انتشار دومین مجموعه داستانهای کوتاه جلال با نام **ازرنجی که می‌بریم** هستیم. مجموعه‌ای که درنهایت وفاداری به شعارهای حزبی نوشته شده و مسائل کارگران و دست‌اندرکاران حزبی شمال ایران را بررسی‌می‌کند و سعی برآفریدن رئالیسم سوسیالیستی دارد اما به‌گفته خودش هیچ موفقیتی بدست نمی‌آورد. جلال بعداً بالحنی تحقیرآمیز از این مجموعه یاد می‌کند زیرا به این نتیجه

چهرهٔ دیگر روشنفکر امروز، جلال است و شعارش انفجار بمب تسلیم در متن سنت و فرهنگ اسلامی ما نیست، انفجار بمب عصیان علیه غربزدگی است و اعلام بازگشت به فرهنگ اسلام و تکیه بر خویش.
دکترشریعتی

رسید که عقایدی که این داستانها بر مبنایشان نوشته شده‌اند، نادرست بوده است. در همین سال کتاب حزب توده بر سر دو راه را نیز به چاپ رساند. درسال ۲۷ مجموعه سه تار را به مَلکی تقدیم کرد و کتاب **قمار باز** داستایوسکی را ترجمه نمود. تمام داستانهای **سه تار** درباره زندگی مردمان ساده‌ایست که قربانی شرایط نابهنجار زندگی می‌شوند. **سه تار** حکایت جوان سه‌تار زنی است که با شکسته شدن و پیچیده شدن سیمهای سه تارش، طومار زندگیش نیز درهم پیچیده می‌شود. در سال ۲۸ کتاب **بیگانه** کامو و سال ۲۹ نمایشنامه **سوءتفاهم** کامو را منتشر ساخت.

درسال ۳۱ مجموعه **زن زیادی** و ترجمه **دستهای آلوده** ،اثر سارتر را انتشار داد. جلال این آثار را برای آموختن زبان فرانسه ترجمه کرد، لیکن نوع نگرش نویسندگان این آثار به زندگی، جلال را به اگزیستانسیالیسم متمایل ساخت.

پس از مدتی اختلافات شدید در دیدگاههای اعضای حزب موجب پیدایش دو دسته سیاسی در درون حزب توده شد و به دنبال مشاجرات شدید، انشعابی در حزب توده صورت پذیرفت و باعث شد جلال و عده‌ای دیگر به رهبری «خلیل ملکی» از حزب توده منشعب شوند. جلال و یارانش پس از انشعاب از حزب توده، «حزب سوسیالیسم توده ایران» را تشکیل دادند. اما بواسطه فشار سوء تبلیغاتی رادیومسکو و مطبوعات حزب توده، تنها دوماه دوام آورد و پس از آن منحل شد. بعد از این واقعه جلال خود را از سیاست کنار کشید و ناچار به سکوتی شد که سه سال بطول انجامید. درجریان قضیهٔ ملی شدن نفت مجدداً به سمت سیاست متمایل شد و از نو فعالیتهای سیاسی را آغاز نمود. درسال ۱۳۳۰ همراه با خلیل ملکی «حزب زحمتکشان ملت ایران» را تأسیس کرد که از نهضت ملی و جبهه ملی حمایت می‌کرد و در واقع جانبدار تزملی شدن صنعت نفت و مصدق بود. از طرفی نیز با حزب توده که از مخالفین مصدق و حامی رژیمهای جدایی طلب بود در جدالی سخت قرار داشت. اما در سال ۳۱ همزمان با استعفای مصدق و بروز اختلاف بین سران حزب، جلال و ملکی و عده‌ای دیگر، از حزب زحمتکشان نیز انشعاب کردند و «حزب نیروی سوم» را تشکیل دادند و سرانجام جلال در سال ۱۳۳۲ بعلت برخی اختلاف نظرها با دیگر رهبران نیروی سوم از آنها نیز کناره گرفت. خود جلال در اینباره می‌گوید: «ما بعلت همین حقه بازیها از حزب توده انشعاب کرده‌بودیم و حالا از نو به سرمان می‌آمد.»

آخرین بخش زندگی جلال برهه‌ای است که درآن جلال به خویشتن خویش بازگشت، از نقطه به جمله رسید و از قطره به اقیانوس مبدل گشت. برهه‌ای که در آن روح سرکش و سرشار از غرور آل احمد به درونی آرام لیکن کوبنده و استوار رسید؛ و به این ترتیب تولدی دوباره یافت: «آیا هیچ فکر کرده‌ای که حج یعنی چه؟ حج کلان را خرد می‌کند و خرد را کلان، بالا را پایین می‌آورد و پایین را بالا. شکسته را پیوند می‌زند و نشکسته را مکسور، حج خس را کس می‌کند و کس را خس و این منوط به حاج است که چگونه باشد.» آری جلال، خودآگاه گمشدهٔ خویش را در بستر و مکتب اسلام یافت تا خواب همیشگی‌اش را درآن مکتب به آخر رساند.

آل‌احمد و شریعتی و امثال اینها روشنفکران صادق و پژوهشگر و حقیقت‌طلبی هستند که پوچی مکاتب مادی و دعاوی دیگران را درک کرده‌اند و نیازهای انسان و دعوت فطرت رافهمیده‌اند.
آیت‌ا...طالقانی

این دوره ۱۶ سال بطول انجامید: از سال ۱۳۳۲ تا سال ۱۳۴۸ (زمان مرگ). جلال دراین ۱۶ سال بسیار سختکوش و پرکار ظاهرشد، از تک‌نگاری و ترجمه گرفته تا داستان و مقاله و سفرنامه.

تک‌نگاری: اولین تک‌نگاری جلال، کتابی بود تحت عنوان اورازان که تصویری زیبا و گویا از زادگاهش، آن روستای فراموش شده، را به نمایش می‌گذارد. تک‌نگاری دوم کتابی است بنام **تات‌نشینهای بلوک زهرا** که روستایی به نام بلوک زهرا از توابع قزوین را مورد بررسی و تحقیق اجتماعی قرارمی‌دهد. سومین تک‌نگاری او **جزیرهٔ خارک در یتیم خلیج‌فارس** بود.

ترجمه: اولین ترجمه جلال در سومین دوره زندگیش کتاب بزرگ و جنجالی آندره‌ژید بود که جایزه نوبل را نیز دریافت‌کرده‌است: **بازگشت از شوروی**. سپس کتاب **مائده‌های زمینی** اثر همین نویسنده را ترجمه نمود. درسال ۴۵ نمایشنامه کرگدن اثر یونسکو و درسال ۴۶ عبور از خط نوشته یونکر را ترجمه کرد.

من، بهترین سالهای جوانیم با محبت و ارادت به آن جلال آل قلم گذشته است... در حرفهایی که رد و بدل شد، هوشمندی، حاضرجوابی، صفا و دردمندی مردی که آن روز در قلهٔ «ادبیات مقاومت» قرارداشت، موج می‌زد...
آیت‌الله خامنه‌ای

جلال در راه بود و با عشق می‌رفت، چرتکه نمی‌انداخت و اصالت داشت و اگر به دین روی آورد از روی دانش و بینش بود.

سیمین دانشور

داستان: جلال در سال ۳۷ دو کتاب **سرگذشت کندوها**و**مدیر مدرسه** را تألیف‌کرد که از مهمترین آثار او بشمار می‌آیند و در حقیقت نقطهٔ عطف نمود قلم وی می‌باشند. **سرگذشت کندوها** انعکاس زندگی سیاسی او از سال ۲۹ تا ۳۱ بود، که منجر به شکست شد. درحقیقت دراین داستان مسئله نفت و بردکمپانی‌ها دراین قضیه مورد بررسی قرارمی‌گیرد. مدیر مدرسه انعکاس برداشتهای جلال از حوضهٔ فرهنگ و مدرسه است. سرگذشت انسان باوجدانی که حاضر به سر فرود آوردن دربرابر نابهنجاریهای جامعهٔ آشفتهٔ خود نیست و رنگ جامعه و جماعت نادان آنرا، تیره‌تر از آن می‌بیند که به رنگش درآید. درنهایت نیز داستان این انسانهای غبارگرفته و درمانده، بدون هیچ نقطهٔ روشنی به‌پایان می‌رسد.

نکته‌ای که لازم است دراینجا به آن پرداخته‌شود نگرش جلال به مقولهٔ زن و انعکاس آن در آثار وی می‌باشد. بدون اغراق می‌توان اذعان کرد که شخصیت زن و مظلومیت او در طول تاریخ ایران‌بخصوص تاریخ معاصر، دغدغه دائمی جلال بوده‌است. شاهد این مدّعا نوع پرداخت وی در داستانهایش می‌باشد: مجموعه **زن زیادی** و داستانهای **لاک صورتی**، **بچه مردم**، **سمنوپزان** و... که درآنها به‌بررسی جایگاه زن در جامعه معاصر و معضلات گریبانگیر وی پرداخته است.

نکته دیگر حساسیت وی پیرامون اوضاع و احوال و مسائل جامعه می‌باشد. جلال درپی آن است که اوضاع نابسامان اجتماعی و عقب‌ماندگیهای فکری و فرهنگی را مدام گوشزد نماید، مجموعه‌های **دید و بازدید**، **سه‌تار** و **زن زیادی** و داستانهای **مدیر مدرسه**، **سرگذشت کندوها** و **تجهیز ملت** از این قبیلند.

مقالات: یکی از مهمترین بخشهای آثار جلال که نقش زیادی در روشنگری و هدایتگری جریانات جامعهٔ آنروز داشته‌اند، همین مقالات او می‌باشند. مقالاتی که در زمینه‌های بسیار متنوع و بعضاً عجیب نگارش شده‌اند: سیاست، جامعه، فرهنگ، شعرفارسی، شاعران معاصر، زبان و خط فارسی، ادبیات فارسی، افسانه‌های ایرانی، نقاشی، طراحی، گرافیک، مجسمه سازی، تئاتر و سینما، بازیگران معاصر، داستان و داستان‌نویسی، نویسندگان معاصر، زبان و فرهنگ ترکی و کتابهای درسی. دراینگونه نوشته‌های جلال، از پرداختن به مسائلی همچون افشای اهداف سوء و مقاصد پشت پرده ایجاد مملکتی بنام اسراییل و یا به‌مبارزه طلبیدن دستگاه مقتدر و عظیم انتشارات فرانکلین (غول انتشاراتی آمریکایی که بازار انتشارات ایران را به تسخیر درآورده‌بود)، تا نوشته‌هایی پیرامون درس انشای مدارس را می‌توان مشاهده‌نمود.

علاوه براینها، جلال مقالات متعددی را نیز در مقام دفاع از نوآوری نیما در شعر فارسی به رشته‌تحریر درآورده است. درواقع او از مدافعین سرسخت سبک نیمایی بود، در شرایطی که شعرش و خودش سخت مورد بی‌مهری جامعه ادبی آنروز قرار گرفته‌بود. باجرات می‌توان گفت که بقای شعر نوی ایرانی مرهون مقاومت و سرسختی آل‌احمد و قلیلی از همفکران او بود.این عدّه در این راه مجادلات فراوانی پرداختند و درضمن آن سیل اهانتها و هتک‌حرمتهای ادیبان و صاحب‌نظران را نیز متحمّل شدند. تعدادی از نوشته‌های جلال پیرامون نیما ویا شعر او عبارتند از: افسانهٔ‌نیما، دفاع از نیما، دوست پیرشده‌ام آقای نیما، نیما دیگر شعر نخواهد گفت، گزارش مرگ نیما یوشیج، پیرمرد چشم ما بود، در زندگانی پس از مرگ و یادبود نیما یوشیج. جلال در مکتوباتش علاوه‌بر نیما تقریباً به اکثر ادبای معاصر پرداخته‌است، از آن‌جمله‌اند: شهریار، اخوان‌ثالث، دهخدا، بهار، جمال‌زاده، سپهری، ساعدی، هدایت، گلستان، بهرنگی، هویدا، منصوری، دشتی، حجازی و ... . در سال ۳۱ از او کتابی منتشر شد بنام **هفت‌مقاله** که مجموعه مقالاتی بود در حوزه مسائل هنری و اجتماعی. در سال ۴۱ سه کتاب بسیار ارزشمند منتشر ساخت به‌نامهای **کارنامهٔ سه ساله** (کارنامهٔ سکوت سه ساله‌اش) **ارزیابی شتابزده** و **غربزدگی** که این آخری آنچنان مفاهیم عظیمی را در خود مستتر کرده‌بود که کمتر کسی توانست بدون تأثیرپذیری از آن از کنارش گذر کند. غربزدگی در طول مدت کوتاهی تأثیری ژرف بر اذهان روشنفکران و صاحبان قلم نهاد و در واقع این کتاب صدای غرب‌ستیزی و ایمان جلال بود که کاملا با نیازهای جامعه مطابقت داشت. یکی دیگر از کتابهای جلال که تا امروز هم روشنگر راه حقیقت‌جویان بوده‌است و فی‌الواقع پژواک ستیزه‌جویی جلال با بی‌دینی و روشنفکری غیر دینی بوده‌است، کتاب **درخدمت و خیانت روشنفکران** می‌باشد، که از آن می‌توان به زنگ‌خطری برای خفتگان و و راهگشایی برای آیندگان تعبیرنمود.

سفرنامه‌ها: علاوه بر نوشته‌های یادشده، از آل‌احمد یکی دو سفرنامه نیز بجامانده است. معروف‌ترین آنها سفرنامهٔ **خسی در میقات** است که حاصل سفر حج اوست. همچنین او درنظر داشت کتابی تحت‌عنوان **چهارقبله** منتشر کند که حاصل مشاهدات او از چهار قبلهٔ انسان معاصر یعنی بیت‌الحرام، آمریکا، شوروی و اسرائیل بود. لیکن دست‌اجل مهلت انتشار آنرا به‌او نداد و تاکنون فقط **خسی در میقات** و **سفر به ولایت عزرائیل** او به‌طور مجزا منتشرشده‌اند.

سرآخر حاج سیدجلال آل‌احمد که ساده می‌پوشید و در خانه‌ای که با دستان خود ساخته‌بود ساده زندگی می‌کرد در ساعت ۶ بعد از ظهر روز هیجدهم شهریور سال ۴۸ در منزل ییلاقی‌اش در اسالم به طرز مشکوکی چشم‌از جهان فروبست و در مسجد فیروزآبادی شهرری به‌خاک سپرده‌شد.

علیرضا بذرافشان

نتیجهٔ نظرسنجی شمارهٔ دوم پویش

به‌همراه شمارهٔ دوم پویش، یک‌برگ فرم نظرسنجی در اختیار دوستان قرار گرفت. استقبالی که از این نظرسنجی شد بسیار چشمگیر بود: از هر ده‌برگ فرم، یکی برگشت که البته نمی‌توان گفت "کم‌است". اگر تیراژ پویش مثلاً یک میلیون نسخه می‌بود، ما الان با چیزی درحدود صدهزار فرم پرشده مواجه بودیم! ولی اینطور نیست (حداقل فعلاً) و حتماً شما هم قبول دارید که با سیزده برگهٔ نظرسنجی، نمی‌توان کار آماری کرد. بنابراین به نقل پاسخهای دوستان به بعضی از سؤالات اکتفا می‌کنیم.

■ مقالات این شماره را چگونه ارزیابی می‌کنید؟

○ راستش را بخواهید من هیچکدام از مطالب علمی شما را نخواندم. ولی مطمئن هستم که در سطح پایینی قراردارند و بهتر است حذف شوند!

○ در مصاحبه، گزارشگر (که ظاهراً از احکام اسلام هم تاحدی بی‌خبر است) سعی کرده تا تلقینات و انتظارات خود را به خواننده القا کند و اگر مصاحبه‌ای هم بوده، با طیف خاصی بوده‌است.

○ منابع مورد استفاده را حتماً ذکر کنید، نه اینکه مقالات دیگران را به اسم خودتان تمام کنید.

■ به نظر شما بهتر است پویش به‌چه‌صورت منتشر شود؟

○ هر طور که می‌توانید! هرچقدر عشق می‌گوید!

■ آیا مایلید با پویش همکاری کنید؟

○ بله. البته اگر مرا پیدا کردید!

■ فکر می‌کنید جای چه مطالبی در پویش خالی‌است؟

○ مطالبِ آموزنده!

■ کلا وجود نشریه‌های دانشجویی مانند پویش را تا چه‌اندازه مفید یا ضروری می‌دانید؟

○ اینکه به‌هرحال دوستان محلی برای اظهارفضل پیدا می‌کنند!

○ اگر صرفاً برای اسم‌درکردن نباشد بد نیست.

○ تا هر اندازه که تصور کنید. زیرا "انسان معاصر در قرن اخیر واقعاً تنهاست".

○ خوب.

○ زمینهٔ مناسبی برای رشد و سامان یافتن خلاقیتها و تلاشهای دانشجویی است.

○ کاملاً ضروری به‌نظر می‌رسد، در محیطی که.

نتیجهٔ مسابقهٔ اول پویش

در شمارهٔ قبل، اولین مسابقهٔ پویش را برگزار نمودیم. از بین جوابهای رسیده، جواب خانم فاطمه رحیمیان بعنوان بهترین تشخیص داده‌شد که هدیه‌ای به‌رسم یادبود به ایشان تقدیم‌خواهدشد.

پاسخ سؤال مسابقهٔ اول

دبیر اول دونفر را می‌شناسد و بنابراین به دوطریق می‌تواند انتخابش را انجام دهد. دبیر دوم سه نفر را می‌شناسد که در بدترین حالت دو نفر از آنها همان دونفری هستند که دبیر اول می‌شناخت و چون کی از آنها توسط دبیر اول انتخاب شده‌اند، برای دبیر دوم هم حداقل دو انتخاب وجود دارد. به همین ترتیب برای دبیر nام در بدترین حالت (که تعداد دانش‌آموزان n+۱) باشد حداقل دو انتخاب وجود دارد. بنابراین تعداد کل حالتهای ممکن حداقل برابر است با: $۲^n=۲\times۲\times۲\times..\times۲$

سؤال مسابقهٔ دوم پویش

۱۷ دانشجوی کامپیوتر از نقاط مختلف دنیا به یک سرویس‌دهندهٔ IRC متصل شده‌اند و در حال انجام گفتگوهای دونفره می‌باشند. قرار بر این است که گفتگوها در مورد سه موضوع ریاتیک، شبکه و رمزنگاری باشد. هر دونفر می‌توانند باهم پیرامون یکی از این سه موضوع که مورد توافق طرفین باشد تبادل نظر کنند. توجه‌کنید که هر شخص می‌تواند همزمان با مخاطب‌های مختلف در مورد موضوعات مختلف صحبت کنند ولی موضوع صحبت هردونفر ثابت، تغییر نمی‌کند. نشان دهید که در هرحال:

الف) همواره سه دانشجو وجود دارند که دوبدو با هم درحال تبادل نظر بر روی یک موضوع (مثلاً شبکه) می‌باشند.

ب) بطورکلی‌تر، همواره حداقل دو گروه سه‌تایی از دانشجویان وجود دارند که دوبدو با هم درحال تبادل نظر برروی یک موضوع می‌باشند.

دقت کنید که موضوع صحبت این دو گروه‌سه‌تایی می‌تواند متفاوت باشد. و نیز اگر دونفر از هر گروه سه نفری در این دوگروه مشترک باشند، موضوع صحبت دوگروه یکسان است.

دراین ولایت کار هنر، جهاد است.
جهاد با بی‌سوادی، با فضل‌فروشی، با فرنگی‌مآبی،
با تقلید، با دغلی، با نان به نرخ روز خوردن، بلغمی مزاجی و...
حالا اگر مردی، این گوی و این میدان.

جلال آل احمد