

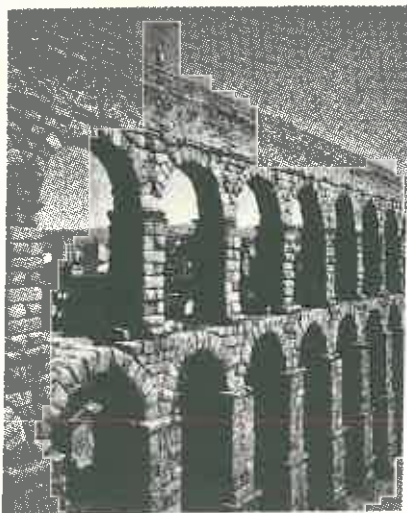
گزارش کابین من

ساخت بیمانه‌ای

روایت تازه‌ای از ویرت
راه سازی موازی
مشاوره کلید و سلامتی



اردیبهشت ۱۳۷۰



تصویر روی جلد: کانال آب‌رسانی متعلق به دوره رم باستان، در شهر سیگویا، اسپانیا.

گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران

سردبیر: علی یارسا

گزارش کامپیوتر فعلاً هر دو ماه یکبار منتشر می‌شود.

مسئولان انجمن

رئیس: آقای ابراهیم نقیب‌زاده‌مشایخ (۲۴۳۸۶۰)

نایب‌رئیس: آقای داریوش جوان‌تبریزی (۵-۶۷۵۱۵۳)

دبیر: خانم میترا کاتوزیان (۶۲۰۶۶۸۹)

خزانه‌دار: آقای علی یارسا (۶۱۴۴۲۵)

سرپرستان کمیته‌ها

انتشارات: آقای علی یارسا (۶۱۴۴۲۵)

آموزش: آقای داریوش جوان‌تبریزی (۵-۶۷۵۱۵۳)

فعالیت‌های علمی و فنی: آقای دکتر کامبیز بدیع (۶۱۳۲۳۲۷۳)

عضویت و روابط عمومی: آقای ابراهیم نقیب‌زاده‌مشایخ (۲۴۳۸۶۰)

دیگراعضای هیئت اجرایی

مسئول هماهنگی کمیته آموزش: آقای حسن کاتوزیان (۳۲-۶۸۷۵۳۰)

مسئول هماهنگی کمیته آموزش: آقای حسین کمالی (۲۸۰۹۵۸)

عضو علی‌البدل: آقای حمید خورشید رحیم‌زاده (۲۹۰۸۸۹)

عضو علی‌البدل: آقای علی زیدی منفرد (۹-۸۲۹۱۸۱)

بازرس انجمن: آقای علی اکبر نهرودی (۶۷۰۷۹۳)

مقاله‌ها و گزارش‌های تالیفی یا ترجمه خود را برای گزارش کامپیوتر بفرستید. مطالب ارسالی را با خط خوانا (یاماشین شده) بر یک روی کاغذ بنویسید. فاصله کافی برای افزودن اصلاحات ویرایشی در بین خط‌ها و حاشیه در نظر بگیرید. در صورت ارسال مطلب ترجمه شده، یک نسخه از اصل مطلب را نیز ارسال دارید.

انجمن انفورماتیک ایران انجمنی است علمی و مستقل که به منظور تبادل اطلاعات علمی و فنی و ارتقا سطح دانش انفورماتیک در ایران تاسیس شده است. این انجمن توسط هیئت اجرایی و زیر نظر مجمع عمومی اداره می‌شود. اعضای هیئت اجرایی با رای اعضای انجمن برای مدت دو سال انتخاب می‌شوند.

مطالب این شماره از گزارش کامپیوتر با استفاده از برنامه $\text{\LaTeX}$ تایپ و صفحه‌بندی شده است. در این شماره از همکاری شرکت‌های داده کاوی ایران، کنترل پروژه‌های صنعتی ایران و پویا برخوردار بوده‌ایم که بدینوسیله از ایشان تشکر می‌کنیم. خانم شیوا دربان و آقایان سعید وحید، حسین بادبانچی و احمد یزدی پور مراحل مختلف صفحه‌بندی و حروفچینی را به عهده داشته‌اند که از ایشان نیز قدردانی می‌شود.

نشانی دفتر انجمن: خیابان دکتر فاطمی، خیابان پروین اعتصامی، کوچه دوم، پلاک ۲۱.

ساعات کار دفتر: هر روز بجز پنج‌شنبه‌ها از ۹/۵ تا ۱۲.

شماره تلفن دفتر: ۶۵۶۶۸۷

شماره حساب بانکی انجمن: ۱۱۱۷ بانک صادرات شعبه ۱۶۵۲

نشانی پستی انجمن: انجمن انفورماتیک ایران، صندوق پستی ۱۹۶-۱۴۱۵۵ تهران.

* * *

GOZARESH-E-COMPUTER

(Computer Report)

Vol.13, No.106, April-May 1991

P.O. Box 1196, Tehran 14155

A publication of the Informatics Society of Iran (ISI). It features original and translated articles, reports, news and reviews on all aspects of computers in Iran and abroad.

ISI is an independent technical and scientific society committed by its constitution to the advancement of computer knowledge in Iran. It is governed by an executive body, members of which are elected every two years by members of the society.

در این شماره

مقاله

تحول مفهوم پیمان‌های بودن نرم افزار ۷

نواذ پاسبان ۱۲.....

مباحثی در برنامه‌سازی موازی ۱۶.....

بخش‌ها

سخن رئیس ۶.....

اخبار انجمن ۳.....

اخبار ایران ۳.....

اخبار جهان ۴.....

نامه‌ها ۵.....

کامپیوتر و بهداشت ۲۱.....

گزارش ۲۳.....

معرفی کتاب ۲۵.....

اعضا ۲۸.....

واژه‌نامه

حرف D ۲۶.....

ترجمه یک کتاب

نفر-ماه افسانه‌ای ۲۹.....

اخبار انجمن

برگزاری سخنرانی اردیبهشت ماه

سخنرانی علمی اردیبهشت ماه انجمن، تحت عنوان "مهندسی نرم افزار به کمک کامپیوتر" توسط آقای ابراهیم نقیب زاده مشایخ در محل گروه ریاضی دانشکده علوم دانشگاه تهران و در حضور عده زیادی از علاقه مندان برگزار شد. کمیته فعالیتهای علمی و فنی انجمن بدین وسیله از مسئولان محترم گروه ریاضی دانشکده علوم دانشگاه تهران به خاطر در اختیار گذاشتن محل سخنرانی و نیز از دانشجویان عزیز این دانشکده که در برگزاری هر چه بهتر این سخنرانی به انجمن یاری رساندند، صمیمانه سپاسگزاری می کند.

اخبار ایران

ثبت انجمنهای علمی

با تلاش مسئولان اداره کل همکاریهای علمی و بین المللی وزارت فرهنگ و آموزش عالی و به پیشنهاد وزیر مربوطه، شورای عالی انقلاب فرهنگی با کلیات انتقال وظائف ثبت و تجدید پروانه فعالیت انجمنهای علمی کشور از وزارت کشور به وزارتخانه های "فرهنگ و آموزش عالی" و "بهداشت، درمان و آموزش پزشکی" موافقت کرده و وزارت فرهنگ و آموزش عالی مسئول تهیه آئین نامه اجرایی آن شده است. بدین منظور، در جلسه ای که در تاریخ ۷۰/۲/۸ در محل اداره کل همکاریهای علمی و بین المللی وزارت فرهنگ و آموزش عالی تشکیل شد، آئین نامه اجرایی پیشنهادی مورد بررسی و اصلاح قرار گرفت. در این جلسه به دعوت مسئولان آن اداره کل، رئیس انجمن نیز حضور داشت. قرار است آئین نامه پیشنهادی از طرف وزیر فرهنگ و آموزش عالی به شورای عالی انقلاب فرهنگی ارائه شود. چنانچه این آئین نامه مورد موافقت شورای مزبور

قرار گیرد یکی از مهمترین موانع موجود بر سر راه فعالیت انجمنهای علمی بر طرف خواهد شد. در حال حاضر، صدور پروانه فعالیت انجمنهای علمی و تجدید آن به عهده وزارت کشور است و مراحل انجام آن به قدری طولانی است که معمولاً پس از پایان دوره فعالیت یک هیأت اجرایی، صلاحیت یا عدم صلاحیت آنان اعلام می شود! این امر باعث شده است که تنها انجمنهای بسیار معدودی آنهم طی مراحل متعدد و دور از شئون فعالیتهایشان، موفق به اخذ پروانه مثبت از وزارت کشور شوند. در آئین نامه پیشنهادی چنین آمده است که وزارتخانه های "فرهنگ و آموزش عالی" و "بهداشت، درمان و آموزش پزشکی" پس از تأیید صلاحیت علمی اعضای هیأت اجرایی انجمنهای علمی، آنها را جهت تأیید صلاحیت عمومی به وزارت کشور معرفی می کنند و در صورتی که ظرف یکماه از آن وزارت جوابی مبنی بر قبول یا رد صلاحیت آنان داده نشود وزارتخانه های مربوطه نسبت به صدور پروانه ثبت و یا تجدید آن اقدام خواهند کرد.

امید است با تصویب نهایی این طرح در شورای عالی انقلاب فرهنگی برای همیشه این مشکل بزرگ از سر راه فعالیت انجمنهای علمی برداشته شود. انجمن انفورماتیک ایران از مسئولان علاقه مند و دلسوز اداره کل همکاریهای علمی و بین المللی وزارت فرهنگ و آموزش عالی که با پشتکار و احساس مسئولیت در رفع مشکلات انجمنهای علمی و فراهم کردن زمینه مساعد برای فعالیت آنان می کوشند، صمیمانه قدردانی می کند.

میزگرد وضعیت کامپیوتر

هیأت تحریریة مجله تدبیر که از طرف سازمان مدیریت صنعتی انتشار می یابد در خردادماه جاری اقدام به برگزاری میزگردی تحت عنوان "وضعیت کامپیوتر در ایران در گذشته، حال و آینده" با شرکت دست اندرکاران انفورماتیک کشور نمود. در این میزگرد آقایان دکتر صنعتی از شرکت نرم افزاری سینا، قائمیان به نمایندگی از شورای عالی انفورماتیک کشور، طالبی رئیس بخش کامپیوتر وزارت نفت، نقیب زاده مشایخ رئیس انجمن انفورماتیک ایران، میرزا عبداللهی

مشاور سازمان مدیریت صنعتی، بهنامی و پوررستمی از طرف مجله تدبیر شرکت داشتند. در این میزگرد که در دو نشست و جمعاً به مدت شش ساعت به طول انجامید، شرکت کنندگان به ارائه نظرات خود در مورد موضوع مطروحه پرداختند و قرار است مشروح بحثها پس از پیاده شدن از روی نوار، در شماره های آینده مجله تدبیر انتشار یابد. خلاصه ای از موضوعات مطرح شده در این میزگرد در شماره بعدی گزارش کامپیوتر درج خواهد شد.

تجدید فعالیت IEEE

بخش ایران انجمن IEEE فعالیت مجدد خود را تحت نام "انجمن مهندسين برق و الکترونیک - بخش ایران" آغاز کرده است. این بخش پس از برگزاری انتخابات و تعیین مسئولین آن تاکنون یک جلسه سخنرانی عمومی برگزار کرده است. سخنران این جلسه آقای دکتر طاهانی از دانشگاه صنعتی اصفهان بودند که در زمینه منطق غیر دقیق (Fuzzy Logic) صحبت کردند. جلسه بعدی این انجمن در ۲۳ خرداد ماه برگزار می شود و در آن آقای دکتر هاشمی گلپایگانی در مورد مهندسی پزشکی صحبت خواهند کرد. انجمن انفورماتیک ایران آغاز فعالیت مجدد IEEE بخش ایران را تبریک می گوید و امیدوار است همکاری علمی و فنی IEEE و انجمن انفورماتیک ایران همانند گذشته به ارتقا دانش انفورماتیک و انتقال تکنولوژی در ایران یاری رساند.

نقشه برداری با کامپیوتر

در روز چهارشنبه ۲۵ اردیبهشت ماه از طرف سازمان نقشه برداری کشوری کنفرانسی تحت عنوان "تهیه نقشه های رقمی و کاربرد کامپیوتر در تهیه نقشه" (Digital Mapping) در محل سالن شهدای هفتم تیر آن سازمان برگزار شد. در این کنفرانس که به همت آقای دکتر محمدعلی شریفی از اعضای هیئت علمی "انستیتو بین المللی نقشه برداری هوایی و علوم زمینی هلند" ترتیب یافته بود، علاوه بر آقای

شریفی، دو تن از اعضای هلندی این انستیتو نیز سخنرانیهایی را ایراد کردند.

کامپیوتر در گفتاردرمانی

از طرف دانشکده توان بخشی دانشگاه علوم پزشکی تهران سمیناری تحت عنوان "کاربرد کامپیوتر در گفتاردرمانی" برگزار خواهد شد. این سمینار در روز سی ام خرداد و در محل آمفی تئاتر دانشکده توان بخشی برگزار می شود و سخنرانان در مورد کاربرد کامپیوتر در گفتاردرمانی و نیز مزیت سیستمهای کامپیوتری نسبت به سیستمهای معمول، به ایراد سخنرانی خواهند پرداخت.

اخبار جهان

قانون کپی رایب

انتظار می رود تصویب قوانین جدیدی از سوی وزرای بازار مشترک اروپا مبنی بر تعمیم قانون حمایت از حقوق مؤلفین به مؤلفین نرم افزار، منافع سرشاری را نصیب تولیدکنندگان نرم افزار آمریکا کند. تخمین زده می شود که فروشندگان آمریکایی سالانه تا پنج میلیارد دلار به دلیل نسخه برداریهای غیرمجاز از نرم افزارهایشان متحمل خسارت می شوند و از این رو تلاش برای جلوگیری از این نسخه برداریها در صدر فهرست الویتهای شرکتهاست. این قانون که از تصویب شورای وزرای اروپا گذشته، اکنون در پارلمان اروپا تحت بررسی است و تصمیم آنها نیز به زودی اعلام خواهد شد. بر طبق این قانون شرکتها از اطلاعات بدست آمده از راه تولید مجدد متن برنامه از روی کد ترجمه شده (Decompiling) تنها برای حصول اطمینان از سازگار بودن محصول خودشان با نرم افزارهای سایر رقیبان، می توانند استفاده کنند و استفاده دیگر از آن اطلاعات مجاز نیست.

زبان ژاپنی در آمریکا

دانشکده مهندسی دانشگاه واشنگتن برای برآورده ساختن نیاز روزافزون به مهندسان و دانشمندانی که به زبان ژاپنی تسلط داشته باشند، یک دوره فوق لیسانس "زبان ژاپنی فنی" را ترتیب داده است که علاوه بر دروس علمی و مهندسی، دوره های آموزش خواندن و نوشتن زبان ژاپنی نیز در آن گنجانده شده است. این برنامه برای پائیز امسال زمانبندی شده و دانشگاه فعلاً مشغول دریافت درخواست ثبت نام داوطلبان است.

سیستم عامل جدید

شرکت نرم افزاری دیجیتال ریسرچ (Digital Research) قابلیت چند کاربری به سیستم عامل دی آر داس خود افزوده و ادعا می کند که این سیستم عامل جدید می تواند جایگزینی برای شبکه های محلی باشد. با استفاده از امکاناتی که این سیستم عامل عرضه می کند، ۶۴ کاربر می توانند هر کدام حداکثر تا هشت کار (Task) که هر یک بیشتر از ۶۲۰ کیلو بایت نباشد را روی یک کامپیوتر شخصی اجرا کنند. سخت افزار مورد نیاز، پردازنده 386SX یا بالاتر و پایانه های سازگار با کامپیوترهای شخصی است. همچنین می توان کامپیوترهای شخصی را نیز به سیستم اصلی متصل کرد. به گفته یکی از مسئولین شرکت "استفاده از شبکه های محلی برای کاربرانی که قصد مرتبط کردن چند کامپیوتر شخصی به هم را دارند پر هزینه است. علاوه بر این با به کارگیری این سیستم عامل خیلی بهتر می توان از توان محاسباتی کامپیوترهای شخصی که پردازنده های ۳۸۶ یا ۴۸۶ دارند سود جست و این محصول مطمئناً در کشورهای اروپای شرقی و شوروی که در آن کمبود کامپیوترهای شخصی چشمگیر است، کارایی بسیار خواهد داشت."

مسابقه برنامه نویسی

پانزدهمین دوره مسابقه برنامه نویسی دانشگاهی که هر سال از سوی انجمن ماشین های محاسب (ACM) برگزار می شود با پیروزی تیم دانشگاه استنفورد پایان یافت. در این مسابقه که در ماه مارس در شهر سن آنتونیو تگزاس انجام گرفت بیست و پنج تیم از دانشگاه های آمریکا، کانادا، اروپا و کشورهای منطقه اقیانوس آرام شرکت داشتند. تیم های دانشگاه فرایه از هلند و موسسه تکنولوژی ویرجینیا مقام های دوم و سوم را کسب کردند. در این مسابقه تعداد زیادی معماهای کامپیوتری که از مسائل واقعی الهام گرفته شده به شرکت کنندگان عرضه می شود و تیمی که تعداد بیشتری از مسائل را در مدت پنج ساعت روی کامپیوتر حل کند برنده می شود. محیط برنامه نویسی مسابقه یک شبکه بر اساس سیستم عامل یونیکس V و زبان برنامه نویسی آن، بسته به انتخاب تیم ها، C یا پاسکال بود.

جایزه تورینگ

جایزه تورینگ که معتبرترین جایزه علم کامپیوتر، و به تعبیری "نوبل" این رشته، به شمار می رود امسال به فرناندو کورباتو استاد موسسه تکنولوژی ماساچوست (MIT) داده شد. این جایزه را انجمن ماشین های محاسب از ۱۹۶۶ تاکنون هر ساله به کسانی که تاثیر مهمی در تحول علم کامپیوتر داشته اند داده است. شرکت ای تی اندتی مبلغ بیست و پنج هزار دلار به برنده جایزه اهدا می کند. کورباتو این جایزه را به خاطر کارهایش در زمینه سیستم های عامل با اشتراک وقت دریافت کرد. او از سازندگان سیستم عامل مالتیکس (MULTICS) است و بسیاری از الگوریتم ها و اندیشه های او در سیستم های عامل امروزی بکار می رود.

نامه‌ها

تکانی شدیدتر

در اخبار جهان شماره ۱۰۵ خبری از خانه تکانی در آی بی ام داشتید که در آن گفته شده بود که این شرکت امسال چهار هزار نفر از کارکنانش را "کنار خواهد گذاشت". برپایه اطلاعاتی که در گزارش مالی سه ماهه اول ۱۹۹۱ این شرکت خوانده‌ام تعداد "کنار گذاشته شدگان" چهارده هزار نفر خواهد بود. این کار از طریق بازنشستگی و بازخرید داوطلبانه صورت می‌گیرد. عده‌ای از این افراد نیز به خاطر فروش قسمت تولید ماشین تحریر و چاپگرهای کوچک آی بی ام از این شرکت بیرون می‌روند. جمشید خوش‌آهوز

پر معنی و نامفهوم

مطالب شماره‌های سال گذشته گزارش کامپیوتر را از اول تا آخر خواندم. باید به کسانی که دست‌اندرکار انجمن و این مجله هستند واقعا تبریک بگویم که کار مشکل و باارزشی می‌کنند. مجله به نسبت آنچه که چند سال قبل دیده بودم خیلی تغییر کرده و پر معنی‌تر شده است. یکی از جالب‌ترین نکته‌های آن استفاده مجله شما از TEX - MARK بود که یکی از موضوعات مورد علاقه من بوده است و چند سال قبل مدتی در این زمینه کار کرده‌ام.

نکته اسف‌انگیزی را که متوجه شده‌ام این است که قسمت زیادی از مطالب مجله برای من قابل فهم نیست! بعضی از مطالب را بعد از ترجمه به انگلیسی می‌فهم ولی بعضی‌هایش "اصلا بی خیال". مثلا در گزارش سمینار هوش مصنوعی یکی از سخنرانی‌ها را هرچه کردم نفهمیدم، نه تیتراش را و نه محتوای صحبت را.

فرهاد ارباب

مرکز ریاضیات و علم کامپیوتر، آمستردام

انتقاد و پیشنهاد سه نفری

- با اینکه بسیاری از اعضای انجمن در تهران هستند و می‌توانند در جلسات انجمن شرکت کنند ولی به دلیل این که گزارش کامپیوتر همیشه یک یا دو روز بعد از سخنرانی‌ها به دستشان می‌رسد موفق به حضور در جلسات نمی‌شوند.

- فرم‌های درخواست عضویت را جداگانه به همراه مجله بفرستید تا مجله را ناقص نکنیم.

- از تصاویر رنگی و جلد مناسب‌تری استفاده کنید.

- با متخصصین ایرانی که در خارج مشغول به کار هستند و در زمینه کامپیوتر فعالیت دارند ارتباط برقرار کنید.

- دانشجویان، خصوصاً دانشجویان در شرف فارغ‌التحصیلی را در مورد بازار کار و فعالیت مفید و سودمند راهنمایی کنید تا با نگرشی درست پا به عرصه فعالیت بگذارند.

- در مورد مطالب مطرح و مهم روز کامپیوتر از جمله ویروس‌ها و سیستم عامل یونیکس بیشتر بنویسید.

از این که بتوانیم فعالیتی برای انجمن انجام دهیم خوشحال خواهیم شد و امیدواریم به نکاتی که مطرح شد ترتیب اثر دهید.

مرضیه پنبه‌ئیان، سبیده مفیدی، سمیرا نمازی

سیستم‌های باز؟

در مقاله شبکه‌های مهم کامپیوتری (گزارش کامپیوتر، بهمن و اسفند ۱۳۶۹) عبارت Open Systems Interconnection به "ارتباط درونی سیستم‌های باز" ترجمه شده بود. باید توجه داشت که کلمه interconnection در آن عبارت به معنای ارتباط سیستم‌ها با یکدیگر است و مفهوم داخل یا درونی در اینجا مورد نظر نیست. کلمه باز هم مربوط به سیستم‌ها نبوده بلکه مربوط به ارتباط سیستم‌ها با یکدیگر است، بدین مفهوم که پیروی از ساختار OSI امکان ارتباط سیستم‌های کامپیوتری مختلف و متفاوت را با یکدیگر فراهم می‌آورد. با توجه به این توضیحات، با حفظ همان کلمات ترجمه شده، می‌توان عبارت را به صورت "ارتباط باز سیستم‌ها" بیان کرد.

کامران قانع

نامه‌های خود را در مورد هر مطلب مربوط به کامپیوتر و انفورماتیک، و نیز نظرات و انتقادهای مربوط به مطالب گزارش کامپیوتر و فعالیت‌های انجمن را، خطاب به سردبیر، به آدرس صندوق پستی انجمن بفرستید. نامه‌هایی که دارای مطالب مورد توجه عموم اعضا باشد، و طولانی هم نباشد (حداکثر یک ستون و نیم) برای چاپ اولویت خواهد داشت.

مسئول بخش اخبار

گزارش کامپیوتر برای اداره بخش اخبار نیازمند همکاری یک دبیر اخبار است. اگر دستی به قلم و دسترسی به نشریات کامپیوتری دارید با آقای پارسا با تلفن ۶۱۴۴۲۵ تماس بگیرید.

سخن رئیس

مشارکت همه جانبه

چند بخشی از نارسائیه‌ها به عدم حمایت عملی و معقول سازمانهای زیربست دولتی مربوط می‌شود ولی قسمت عمده‌ای از مشکلات نیز معلول بی‌تفاوتی و عدم مشارکت وسیع و همه جانبه اعضا در فعالیتهای انجمن است.

هم اکنون انجمن نزدیک به ۲۵۰ نفر عضو عادی، یعنی افرادی با درجه لیسانس یا بالاتر در رشته علوم کامپیوتر و رشته‌های وابسته به آن دارد و این به خودی خود، نیروی بالقوه عظیمی را تشکیل می‌دهد. اگر همه این افراد و همین‌طور سایر اعضای انجمن، که اکنون نزدیک به ۷۰۰ نفر می‌باشند، فقط به پرداخت حق عضویت سالانه اکتفا کنند، که عده‌ای این تنها کار را هم نمی‌کنند و هر سال انجمن تعدادی از اعضایش را به خاطر عدم تمدید عضویت از دست می‌دهد، و در پیشبرد اهداف والای انجمن که وضع استانداردهای علمی و فنی، بالا بردن سطح دانش انفورماتیک از طریق کمک به بهبود کیفیت آموزش، ایجاد همکاری بین مراکز علمی، صنعتی و اجرایی، و توسعه و تشویق فعالیتهای پژوهشی از آن جمله‌اند، مشارکت فعال و همه جانبه نداشته باشند، به طور طبیعی چند نفر انگشت شمار بیش از این نمی‌توانند منشأ اثری باشند.

با توجه به آنچه گفته شد، اکنون که به برگزاری انتخابات هفتمین هیأت اجرایی انجمن نزدیک می‌شویم از تمامی اعضای عادی انجمن درخواست می‌کنم فعالانه در این انتخابات شرکت کنند. تمامی کسانی که دواطلب عضویت در هیأت اجرایی می‌باشند می‌توانند تا پایان مرداد ماه، با ذکر مختصری از سوابق علمی و کاری، آمادگی خود را به اطلاع کمیته روابط عمومی انجمن برسانند. به امید آن که انجمن بتواند با دریافت حمایت‌های مادی و معنوی مسئولان زیربست دولتی از یکسو و داشتن هیأت اجرایی منسجم و فعالی از سوی دیگر، گامهای مؤثرتری در تحقق بخشیدن به اهداف خود بردارد، برای این منظور، مشارکت وسیع همه شما ضرورت دارد.

رئیس انجمن

دوره فعالیت ششمین هیأت اجرایی انجمن در شهریورماه امسال به پایان می‌رسد و در مجمع عمومی مهرماه سال جاری باید افراد هفتمین هیأت اجرایی انجمن انتخاب شوند. تاکنون رسم بر این بوده است که هرگاه موضوع انتخابات هیأت اجرایی مطرح می‌شده، مسئولان انجمن برای آن که تعداد داوطلبان، حداقل به تعداد مورد نیاز برسد، با اصرار و پافشاری زیاد، افرادی را تشویق به قبول کاندیداتوری می‌کردند. تقریباً در هیچ دوره‌ای نبوده است که تعداد داوطلبان "واقعی" بیش از تعداد مورد نیاز بوده باشد تا دیگر نیازی به این گونه تمهیدات نباشد. و طبیعی است افرادی که با این ترتیب به عضویت هیأت اجرایی انجمن درآیند، منشأ خدمات چندانی نخواهند بود. و بدین خاطر است که از بدو تأسیس انجمن تاکنون همواره بار تمامی فعالیتهای و زحمات به دوش چند نفر محدود بوده است و اگر پشتکار و همت بلند آنها نمی‌بود، اکنون جز اسمی از انجمن، چیز دیگری باقی نمانده بود. طبیعتاً این وضع برای مدتی طولانی قابل دوام نخواهد بود: یا آن چند نفر نیز بالاخره طاقتشان به سر خواهد رسید و با توجه به دشواریهای زندگی، عطای این فعالیت جانبی و پر زحمت و افتخاری را به لقایش خواهند بخشید و یا آن که حداکثر اگر هم دوام بیاورند، دامنه فعالیتهای انجمن، گسترده‌تر از این نخواهد شد.

در شرایط امروز که بازار کامپیوتر در کشور بسیار پر رونق است، و کاربردهای کامپیوتر در تمامی زمینه‌ها فراگیر شده، و سازمانها و شرکتهای بزرگ و کوچک در خریداری و نصب و پیاده‌سازی سیستمهای کامپیوتری با یکدیگر مسابقه گذاشته‌اند، و همه در حرفه‌های گوناگون مشتاقانه در صدد آموزش کامپیوتر و کار با آن هستند، و ویدئو کلوبهای سابق به کامپیوترفروشی و شرکتهای کامپیوتری با کلاسهای آموزشی تبدیل شده‌اند، و جای خالی دوره‌های آموزش صحیح و بازآموزی در کلیه سطوح به شدت به چشم می‌خورد، و ... انجمن انفورماتیک ایران به عنوان انجمن علمی و تخصصی رشته کامپیوتر در ایران باید سهم بسیار فعالانه‌تری از چاپ یک نشریه، برگزاری سخنرانیهای علمی ماهانه و برگزاری یکی دو دوره آموزشی و یا سمینار تخصصی در سال داشته باشد.

البته آنچه اکنون می‌شود نیز با توجه به تعداد محدود افرادی که در انجمن فعالیت می‌کنند و نیز با در نظر گرفتن بودجه بسیار ناچیز و اندک آن، واقعاً شایان توجه و قابل تقدیر است و هیأت اجرایی فعلی انجمن را می‌توان به جرأت، پرکارترین و پرثمرترین هیأت‌های اجرایی طول حیات انجمن دانست. ولی بحث بر سر آن است که شرایط امروز کشور در زمینه انفورماتیک، دامنه بسیار وسیعتری از فعالیتهای از سوی ارگانی علمی و مستقل مانند انجمن انفورماتیک ایران می‌طلبد. هر



مقاله

تحول مفهوم پیمانه‌ای بودن نرم‌افزار

نوشته: دیوید کوپکی*

ترجمه: ابراهیم نقیب‌زاده مشایخ

فرضیاتی در مورد عملیات پیمانه‌های دیگر بجز آنچه در فاصل
تعریف شده گنجانده شده است به عمل نیاورد. ^۱ (لی)

به عبارت دیگر، پیمانه یک قطعه فراخوانی‌پذیر از برنامه است که یک
سپر حفاظتی مفهومی دارد و تنها از طریق پارامترهایش می‌توان به آن
دستیابی داشت.

استقلال پیمانه‌ها ممکن است به معنی استقلال در یک سطح فیزیکی‌تر
نیز باشد. یعنی پیمانه‌ها می‌توانند به صورت زیر برنامه‌هایی با قابلیت
ترجمه^۲ جداگانه نیز تشریح شوند. پیمانه‌ای بودن به این معنی هم
یافت می‌شود:

باید بتوانیم مستقل از زمینه استفاده یک پیمانه در ساخت
واحدهای نرم‌افزاری بزرگتر، از صحت و درستی عملکرد آن
اطمینان یابیم. دنیس^۳

تاریخچه

ابتدای این امر از ابداع زیرروال آغاز شده است. در سال ۱۹۴۵، کنراد
زوسه^۴ علائمی را برای زیرروال‌ها تشریح کرد ولی هنوز سخت‌افزار
مناسبی برای پشتیبانی از ایده‌های وی وجود نداشت و کارهای او تا سال
۱۹۷۲ انتشار نیافت. با این وجود، نوشته‌های زوسه نشان می‌دهد
که مفهوم زیرروال در آن سال (یعنی قبل از پیدایش کامپیوترهای با
برنامه ذخیره شده^۵) به وجود آمده است.

در سال ۱۹۴۹، ویلر^۶ مستقل از زوسه، زیرروال‌های بسته^۷ را ابداع
کرد. زیرروال بسته برای نخستین بار اجازه تولید واحدهای جداگانه
برنامه را که قابلیت بکارگیری مجدد را داشتند، می‌داد. با انتقال
پارامترها، محاسبات متفاوتی امکان‌پذیر بود. پیش از این، تنها گونه
زیرروال‌های موجود، زیرروال باز بود که در آن، دنباله‌ای از فرمانها در
داخل متن برنامه اصلی درج می‌شد.

این مقاله ترجمه بخشی از رساله فوق‌لیسانس نویسنده است که به بررسی
چگونگی پیدایش و تکامل یکی از مفاهیم بنیادی در طراحی نرم‌افزار، که
امروزه نیز در قلب مهندسی نرم‌افزار مدرن جای دارد، می‌پردازد. برنامه‌سازی
ساخت‌یافته، تجرید، همزمان‌سازی و جریان داده‌ها از دیگر عناوین این رساله
هستند.

معنی دقیق پیمانه‌ای بودن^۱ تا حدودی مبهم و نامعلوم است زیرا در
متون این رشته، تعاریف نسبتاً متفاوتی برای آن وجود دارد. تعدادی از
تعاریف، صرفاً به منظور باز کردن مفهوم و معنی لفظ پیمانه^۲ به کار برده
شده‌اند. در ابتدا از این لفظ برای تشریح یک دنباله فراخوانی‌پذیر^۳
از احکام (یعنی زیرروال^۴) استفاده می‌شد. پیمانه‌ای بودن به معنی
این بود که برنامه‌ها از پیمانه‌ها تشکیل یا ایجاد شده باشند. سپس
مشاهده شد که راههای خوب و بدی برای تقسیم یک برنامه به پیمانه‌ها
وجود دارد. علت آن است که چگونگی تعریف پیمانه‌ها، تعیین‌کننده
وابستگی درونی آن‌ها به یکدیگر نیز هست و این وابستگی درونی در
صورتی که کم باشد به پیمانه‌های خوب و چنانچه زیاد باشد به پیمانه‌های
ضعیف و بد (از لحاظ اصلاح‌پذیری^۵) می‌انجامد.

در نتیجه، معنی و مفهوم پیمانه‌ای بودن مستلزم تشریح رابطه درونی
پیمانه‌ها با یکدیگر نیز می‌باشد. این رابطه غالباً برحسب اهداف بیان
می‌شود:

... هدف اولیه باید تجزیه برنامه به نحوی باشد که پیمانه‌ها
بسیار مستقل از یکدیگر باشند. هیر^۶

استقلال پیمانه‌ها یعنی:

... فاصل‌های^۷ به خوبی تعریف شده به نحوی که هیچ پیمانه،

David J. Koepke, "The Evolution of Software Design Ideas", *Annals of the History of Computing*, vol.12, no.4, 1990.

Earley	^۸
compile	^۹
Dennis	^{۱۰}
Konrad Zuse	^{۱۱}
stored program computers	^{۱۲}
Wheeler	^{۱۳}
closed subroutine	^{۱۴}

modularity	^۱
module	^۲
callable	^۳
subroutine	^۴
modifiability	^۵
Myers	^۶
interface	^۷

توضیحات ویر هر چند برای آن سال‌ها پیشرفته محسوب می‌شود ولی برای تعیین دقیق منظور او دارای ابهام است. "برنامه تشکیل یافته از مجموعه‌ای از زیرروال‌ها" خیلی پیمانه‌ای به نظر می‌آید ولی روشن نیست که آیا منظور او این بوده است که برنامه کلاً از ترکیب زیرروال‌ها تشکیل شده باشد یا نه. توضیحات تکمیلی بعدی وی در مورد این که "تفکیک برنامه کامل‌تر خواهد شد" نیز بسیار مبهم است و معنی و مفهوم بیشتری را القاء نمی‌کند.

در طول دهه ۱۹۵۰، زیرروال‌ها به عنوان توابع استاندارد برای استفاده باقی ماندند. در اوایل دهه ۱۹۶۰ با ظهور فرتن (۱۹۵۴-۱۹۵۸) و الگول ۶۰ (۱۹۶۰-۱۹۵۷)، مفهوم پیمانه‌ای بودن پدیدار شد. این زبانها سطح بالا بودند و ساختار آن‌ها (رویه‌ها^{۱۷} و توابع) انعکاس افکار جاری در مورد برنامه‌سازی بود.

دایکسترا^{۱۸} چنین می‌گوید:

با ظهور الگول ۶۰، صحنه عوض شد و دیگر ما در مورد زیرروال‌های بسته فکر نکردیم: در عوض آن‌ها را رویه می‌نامیدیم. آن‌ها به عنوان ابزارهای خیلی راحتی برای کوتاه کردن متن برنامه مورد استقبال برنامه‌سازان قرار گرفتند و برنامه‌سازان بیشتر و بیشتری به منظور ساخت دهی برنامه‌ها شروع به استفاده از آن‌ها کردند. بدین ترتیب، پیش‌بینی‌های لازم برای تطبیق برنامه با تغییرات آتی در مشخصات^{۱۹} مسأله، به جایگزین ساختن یک یا چند بدنه رویه با برخی تغییرات در پارامترها محدود شد.

در ۱۹۵۷، بیکر^{۲۰} در بخشی از مرور یک کتاب، بعضی از ویژگیهای پیمانه را تشریح کرد:

یکی از روشهایی که بسیار مورد استفاده قرار می‌گیرد، شکستن مسأله به زیرروال‌های کوچک و خودشمول و سعی در مجزّا کردن بخشهای مختلف برنامه از یکدیگر تا حد امکان و هر چه بیشتر است.

عبارتهای خودشمول و مجزّا، پیمانه‌ای بودن را دقیقتر از پیش تشریح می‌کرد ولی نکته مهم‌تری که در گفته بیکر وجود داشت، "تا حد امکان و هر چه بیشتر" بود. بنابراین، سال ۱۹۵۷ آغاز فکر کردن به برنامه‌ها به صورت این که کلاً از پیمانه‌ها تشکیل یافته باشند، بود. پیش از آن هر برنامه برای سهولت کار، چند زیرروال را فراخوانی می‌کرد.

جنبه دیگری که غالباً مربوط به پیمانه‌ای بودن می‌شود، قابلیت جداگانه ترجمه شدن هر پیمانه است. زیرروال‌هایی که بتوانند جداگانه ترجمه شوند برای اولین بار در سال ۱۹۵۸ در فرتن ۲ پیاده‌سازی شدند. این خاصیت بدین معنی بود که چنانچه تغییر یا اصلاحی منحصراً در محدوده یک زیرروال تأثیر می‌گذاشت، به ترجمه دوباره کل برنامه نیازی نبود.

procedures^{۱۷}
Dijkstra^{۱۸}
specification^{۱۹}
Bakre^{۲۰}

زیرروال‌های بسته تشکیل برنامه‌ها از پیمانه‌ها را ممکن ساختند ولی در سال ۱۹۵۱ نه چنین استفاده‌ای از آن‌ها می‌شد و نه حتی فکرش را می‌کردند، بلکه آن‌ها را ابزاری برای صرفه‌جویی در امر برنامه‌نویسی به حساب می‌آوردند. برای استفاده از یک زیرروال موجود در برنامه‌ای دیگر، به بازنویسی آن نیازی نبود. ویلکس^{۱۵} چنین توضیح می‌دهد:

چنانچه برنامه‌های آماده و کوتاهی برای انجام عملیات محاسباتی متداول در دسترس باشند از حجم کار انسانی لازم برای تولید برنامه‌ای برای یک مسأله خاص کاسته می‌شود. این برنامه‌های کوچک معمولاً زیرروال خوانده می‌شوند و می‌توان آن‌ها را در هر جای لازم در برنامه قرار داد و بدین ترتیب از حجم کاری که باید صورت گیرد کاست.

زیرروال‌ها از طریق قرار گرفتن در کتابخانه زیرروال‌ها در اختیار برنامه‌سازان دیگر قرار گرفتند. امتیاز مهم کتابخانه زیرروال‌ها این بود که از احتمال اشتباهی که در دوباره‌نویسی برنامه‌ها وجود داشت می‌کاست و بدین ترتیب، زمان اشکال‌زدائی را کاهش می‌داد.

هنگامی که یک برنامه از مجموعه‌ای از زیرروال‌ها تشکیل شده باشد، تفکیک برنامه کامل‌تر از وقتی است که غیر از این باشد.

در سال ۱۹۵۰، کاری^{۱۶} اجزاء برنامه را به صورت مستقل از یکدیگر تشریح کرد و بنابراین در زمره اولین کسانی قرار گرفت که به یکی از مهمترین ویژگیهای پیمانه‌ای بودن پرداختند. او می‌گوید:

اولین مرحله در برنامه‌ریزی و تدارک یک برنامه، تجزیه کردن محاسبات در قالب چند بخش عمده ... است به نحوی که کل برنامه را بتوان از ترکیب آن‌ها تشکیل داد. این بخشهای عمده باید به گونه‌ای باشند که همگی یا حداقل بعضی از آن‌ها به تنهایی، محاسبات مستقلی باشند ...

مقاله کاری تا سال ۱۹۷۲ انتشار نیافت ولی نشان می‌دهد که این مفهوم که برنامه در بخشهای مستقلی سازماندهی شود، به سال ۱۹۵۰ باز می‌گردد.

ویلر در سال ۱۹۵۲ چنین گفته است:

هنگامی که یک برنامه از مجموعه‌ای از زیرروال‌ها تشکیل شده باشد، تفکیک برنامه کامل‌تر از وقتی است که غیر از این باشد.

Wilkes^{۱۵}
Curry^{۱۶}

می‌کنند باید به عنوان زیرپیمانه‌های ۲۳ یک پیمانه سطح بالاتر وجود داشته باشند. این ایده را می‌توان به عنوان پیش درآمدی برای ایده داده‌گونه‌های تجربی ۲۴ در نظر گرفت.

مفهومی که امری از پیمانه‌ای بودن در نظر داشت با معنی امروزی آن کمی متفاوت است. او پیمانه‌ها را بلوکهای ساختی استاندارد تصوّر می‌کرد که به تناسب یک کار خاص باید بکار گرفته و احتمالاً کمی اصلاح می‌شدند. او این ایده را از صنعت ساختمان‌سازی وام گرفته بود. در آنجا پیمانه‌های استاندارد (نظیر قطعه‌های بتونی استاندارد) را می‌توان برای ساختن بناها بکار گرفت (امری خود این مطلب را در نامه‌ای که به نویسنده مقاله نوشته خاطر نشان ساخته است).

استفاده از زیرروال‌های کتابخانه به عنوان بلوکهای ساختی برنامه، فقط تا اندازه‌ای موفقیت‌آمیز بود زیرا غالباً مستندات، ناکافی بود و پارامترها همیشه اجازه استفاده‌های متنوع و مختلف را نمی‌داد.

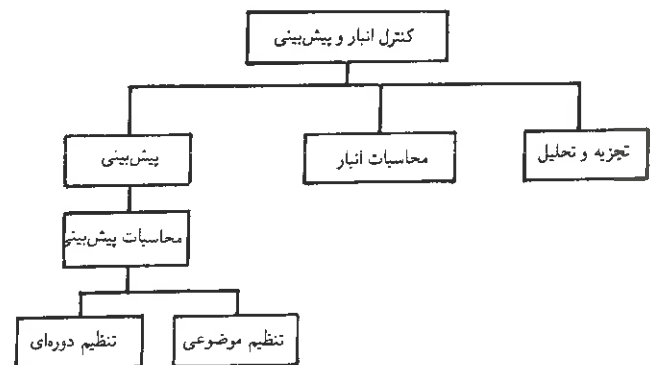
در ۱۹۶۵، کنستانتین ۲۵ پیمانه‌ای بودن را به عنوان یک جعبه سیاه با حداقل وابستگی درونی و ارتباطات محدود بین پیمانه‌ها تعریف کرد. عبارت جعبه سیاه (یعنی شیئی با نحوه عملکرد خارجی معلوم و نحوه عملکرد داخلی نامعلوم) برای اولین بار پوسته‌ای به دور پیمانه‌ای بودن می‌کشید.

اتصال، معیاری است برای شدت وابستگی بین دو پیمانه و چسبندگی معیار شدت ارتباط بین عناصر یک پیمانه است.

سه سال بعد یعنی در ۱۹۶۸، کنستانتین عبارات اتصال ۲۶ و چسبندگی ۲۷ را ارائه کرد که به توضیح استقلال پیمانه‌ها کمک می‌کرد. اتصال، معیاری است برای شدت وابستگی بین دو پیمانه و چسبندگی معیار شدت ارتباط بین عناصر یک پیمانه است. برای مثال، کنستانتین در تشریح درجه اتصال، عوامل زیر را تعیین کننده می‌دانست: پیچیدگی فاصل پیمانه، نوع ارتباط (چگونه فراخوانی می‌شود) و نوع اطلاعاتی که می‌فرستد (داده، کنترل، یا ترکیبی از هر دو). معیار چسبندگی بر روی عناصر پیمانه تمرکز داشت. یعنی پیمانه‌هایی که تنها یک کار منفرد را انجام می‌دادند، با بیشترین میزان چسبندگی در نظر گرفته می‌شدند. این ایده‌ها پیش درآمد ایده پنهان‌سازی اطلاعات ۲۸ بودند زیرا معنی استقلال پیمانه‌ها را روشن می‌ساختند.

دایکسترا در سال ۱۹۶۹ به تشریح مفهوم پیمانه با تشبیه آن به یک

submodule	۲۳
abstract data type	۲۴
Constantine	۲۵
coupling	۲۶
cohesion	۲۷
information hiding	۲۸



شکل ۱- نمودار پیشنهادی امری برای سلسله‌مراتب پیمانه‌ها

عبارت "پیمانه‌ای" به سال ۱۹۶۰ و مقاله منتشر نشده‌ای با عنوان "برنامه‌سازی پیمانه‌ای" از رابرت بیزل ۲۹ باز می‌گردد.

نخستین باری که عبارت "پیمانه‌ای" در یک مقاله منتشر شده بکار رفت در سال ۱۹۶۲ و در مقاله امری ۳۰ با عنوان "سیستمهای داده‌پردازی پیمانه‌ای در زبان کوپول" بود. امری در این مقاله، پیمانه را به عنوان برنامه‌های قطعه قطعه شده یعنی برنامه‌هایی که به خوبی به چند قطعه تقسیم شده‌اند، تعریف کرده است. این قطعه‌ها به صورت زیرروال‌ها پیاده‌سازی شده و معمولاً از یک کتابخانه زیرروال‌ها فراخوانی می‌شدند. امتیاز جدید این روش پیمانه‌ای، افزودن انعطاف‌پذیری در تغییر برنامه‌ها بود. به توضیحات امری توجه کنید:

به منظور به دست آوردن انعطاف‌پذیری بیشتر در طراحی سیستمها، پیمانه‌هایی که در کتابخانه پیمانه‌ها در دسترس طراح سیستم قرار دارند باید تا حد زیادی از یکدیگر مستقل باشند. بدین ترتیب، وجود یا عدم یک پیمانه، تأثیری بر سایر پیمانه‌ها ندارد. به دلیل این استقلال، اصلاح یا توسعه پیمانه‌ها در کتابخانه نسبتاً ساده‌تر صورت می‌گیرد.

هر چند امری اظهار داشت که پیمانه‌ها باید مستقل باشند ولی هیچگونه توضیحی در مورد چگونگی دستیابی به این استقلال نداد. مقاله امری شامل نمودارهایی از سلسله مراتب پیمانه‌ها (نظیر شکل ۱) بود که نشان می‌دهد در سال ۱۹۶۲ برنامه‌ها کلاً متشکل از پیمانه‌ها در نظر گرفته می‌شدند.

به این ترتیب، مفهوم پیمانه از دنباله مفیدی از احکام به مؤلفه‌هایی در طراحی سیستم تغییر یافت. امری توصیه کرد که پیمانه‌ها برای آن بخشهایی از کار که احتمال تغییر در آن‌ها بیشتر است به وجود آیند. وی همچنین پیشنهاد نمود که پیمانه‌هایی که با داده‌های مشابهی کار

۲۹ Robert Bizzell

۳۰ Emery

مروارید پرداخت:

از آنجا که هر مروارید نشانگر تصمیم‌گیری خاصی از لحاظ طراحی (یا بسته به مورد، نشانگر جنبه‌ای از مسئله اصلی) است، باید آن را به عنوان واحد طبیعی معاوضه در تبدیل برنامه یا بسته به مورد، در تطبیق برنامه با تغییرات مسئله در نظر گرفت.

توضیحات دایکسترا دلیل به وجود آوردن پیمانه‌ها را روشن ساخت: تعبیه یک تصمیم‌گیری خاص طراحی، یا جنبه‌ای از مسئله، در هر پیمانه. تشبیه پیمانه به مروارید نیز کیفیت محفوظ‌تری برای آن قائل گردید و این امر با پنهان‌سازی اطلاعات تقویت شد.

یک پیمانه باید تماماً شامل کلیه اطلاعات و آگاهیهای مربوط به یک تصمیم‌گیری خاص طراحی باشد و آن‌ها را از سایر پیمانه‌ها پنهان کند.

پنهان‌سازی اطلاعات

این ایده چنین است که یک پیمانه تماماً اطلاعات راجع به کارهای داخلش را حتی‌الامکان از سایر پیمانه‌ها پنهان کند. تنها اطلاعات آشکار، در فاصل پیمانه قرار دارند. (یعنی پارامترهایی که فرستاده می‌شود). پنهان‌سازی اطلاعات همچنین حکم می‌کند که این پارامترها نیز حتی‌الامکان چگونگی انجام عملیات پیمانه را آشکار نکنند. شاید اولین علائم پنهان‌سازی اطلاعات، ایده برنامه‌سازی بدون داده^{۲۹} بود که در سال ۱۹۶۷ توسط بالزر^{۳۰} بیان شد. وی برنامه‌سازی بدون داده را چنین توضیح می‌دهد:

یک برنامه به صورت مشخصات مجموعه‌ای از عملیاتی که باید بر روی مجموعه‌ای از مقادیر داده‌ها انجام گیرد، در نظر گرفته می‌شود و این مشخصات نیز باید مستقل از شکل نمایش مقادیر داده‌ها باشند. برای حصول چنین استقلال، باید مجموعه‌ای از اعلانها^{۳۱} وجود داشته باشد که به سیستم برنامه‌سازی بگوید چگونه مقادیر داده‌ها را از روی یک نحوه نمایش بخصوص ذخیره و بازیابی نماید.

ایده بالزر، مجزاً ساختن داده‌ها و عملیات روی آن‌ها در یک مجموعه داده‌ای جدا از برنامه مبداء^{۳۲} بود. برنامه می‌توانست از طریق مراجعه

^{۲۹} dataless programming
^{۳۰} Balzer
^{۳۱} declaration
^{۳۲} source program

به نام مجموعه، همراه با یک عمل استاندارد (حذف، اضافه، جایگزینی و ...) به داده‌ها دستیابی داشته باشد. استفاده از این شیوه بدین معنی است که تغییر یک ساختمان داده‌ها از آرایه به لیست پیوندی، فقط اعلان داده‌ها را تغییر می‌دهد و تأثیری بر برنامه مبداء نمی‌گذارد. استفاده از یک مجموعه استاندارد از عملگرها برای کلیه داده‌گونه‌ها کاملاً عملی نبود ولی ایده بالزر در مورد پنهان ساختن جزئیات پیاده‌سازی داده‌ها، پیش درآمد پنهان‌سازی اطلاعات بود.

دید کلیتری نسبت به پنهان‌سازی اطلاعات در سال ۱۹۷۱ توسط پارناس^{۳۳} ارائه شد. او توضیح داد که هر چه اتصال بین پیمانه‌ها حاوی اطلاعات کمتری باشد بهتر است. او اتصالات بین پیمانه‌ها را به عنوان فرضیاتی که هر پیمانه نسبت به سایر پیمانه‌ها به عمل می‌آورد، تعریف کرد. طبق توضیحات پارناس، دلیل محدود ساختن مقدار اطلاعات در اتصالات پیمانه‌ها، تسهیل اثبات صحت پیمانه‌ها و اصلاح آن‌ها است.

نکته دومی که پارناس روی آن تأکید نمود این بود که باید به برنامه‌سازانی که سرگرم تولید یک پیمانه خاص هستند، تنها اطلاعات لازم برای پیاده‌سازی همان پیمانه را داد (یعنی اطلاعات محدود به جای تمام اطلاعات موجود درباره سیستم در حال ایجاد). بدین ترتیب، برنامه‌ساز نمی‌تواند هیچ فرض غیر لازمی در مورد سایر پیمانه‌ها به عمل آورد. تذکر این نکته جالب است که ایده پنهان‌سازی اطلاعات، که در سال ۱۹۷۱ توسط پارناس ارائه شد، با دیدگاه فعلی آن کمی متفاوت است. پارناس می‌گفت که تولید کننده یک پیمانه (و در نتیجه خود پیمانه) تنها می‌تواند اطلاعات محدودی از سایر پیمانه‌ها داشته باشد در حالی که امروزه گفته می‌شود که یک پیمانه باید اطلاعاتش را از کلیه پیمانه‌های دیگر پنهان سازد. به هر حال چه پیمانه اطلاعات محدودی از سایر پیمانه‌ها داشته باشد و چه هر پیمانه کلیه اطلاعات را از سایر پیمانه‌ها پنهان سازد، نتیجه یکی است و آن این که اطلاعات در داخل پیمانه پنهان است.

یکسال بعد در سال ۱۹۷۲، پارناس مقاله‌ای تحت عنوان "ضوابطی که باید در تجزیه سیستمها به پیمانه‌ها بکار رود" منتشر ساخت. او در این مقاله می‌گوید:

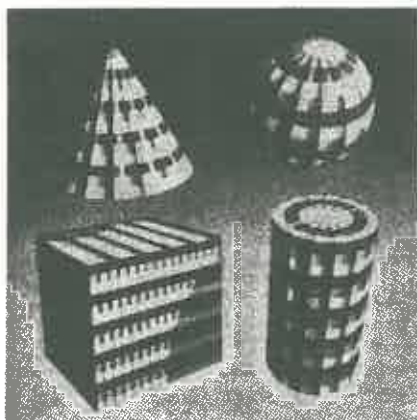
موثر بودن پیمانه‌ای کردن به ضوابط مورد استفاده در تقسیم یک سیستم به پیمانه‌ها بستگی دارد.

او می‌گوید ضابطه‌ای که باید بکار رود، پنهان‌سازی اطلاعات است. منظور او از پنهان‌سازی اطلاعات این است که یک پیمانه باید تماماً شامل کلیه اطلاعات و آگاهیهای مربوط به یک تصمیم‌گیری خاص طراحی باشد و آن‌ها را از سایر پیمانه‌ها پنهان کند. به عبارت دیگر، سایر پیمانه‌ها نمی‌توانند هیچ فرضی در مورد تصمیم‌گیریهای پنهان شده طراحی بجز از طریق پارامترها به عمل آورند.

پارناس پیشنهاد کرد که پیمانه‌ها بر اساس آن تصمیم‌گیریهای طراحی

اگر برنامه می نویسید این کتاب برای شما است

آزمونی برای طراحان نرم افزار



نوشته: رابرت ل. بابر

ترجمه: ابراهیم تقیپ زاده مشایخ



بهای این کتاب برای اعضا ۵۰۰ ریال و برای افراد غیرعضو ۹۰۰ ریال است. برای تهیه کتاب با دفتر انجمن انفورماتیک ایران تلفن ۶۵۶۶۸۷ تماس بگیرید.

که احتمال تغییر در آن‌ها وجود دارد، ساخته شوند. لیستی از چنین تصمیماتی تهیه شود و پیمانه‌ها برای پنهان ساختن آن تصمیمها به وجود آیند. بدین شکل، تغییرات را می‌توان در داخل یک پیمانه به سادگی اعمال کرد. پاراناس همچنین خاطر نشان ساخت که فاصل یک پیمانه باید به گونه‌ای انتخاب شود که تا حد امکان اطلاعات کمتری در مورد چگونگی کارکرد آن آشکار سازد. یعنی استفاده از پارامترهای عمومی‌تر و به تعداد کمتر.

روش سنتی تجزیه سیستمها، یعنی قرار دادن هر مرحله عمده پردازشی در یک پیمانه، نادرست است.

پاراناس علاوه بر ضوابط عمومی پنهان سازی اطلاعات برای پیمانه‌ها، نکات زیر را نیز در مورد تجزیه یک سیستم توصیه نمود:

- یک ساختمان داده، سازماندهی داخلی و رویه‌های دستیابی و اصلاح آن، همگی باید بخشی از یک پیمانه منفرد باشند (ایده‌ای که در پشت داده‌گونه‌های تجربیدی قرار دارد).
- دنباله دستورالعمل‌های لازم برای فراخوانی یک روال مفروض و خود آن روال باید بخشی از یک پیمانه منفرد باشند.
- قالبهای ۳۴ بلوکهای کنترلی صفها در سیستمهای عامل باید در یک پیمانه منفرد پنهان باشد.
- رمزهای نویسه‌ای^{۳۵}، ترتیبهای الفبائی و داده‌های مشابه باید در یک پیمانه منفرد گنجانده شوند.
- ترتیب اجرای پیمانه باید در یک پیمانه منفرد پنهان باشد.

پاراناس اظهار داشت که روش سنتی تجزیه سیستمها، یعنی قرار دادن هر مرحله عمده پردازشی در یک پیمانه، نادرست است زیرا در اغلب موارد، تصمیمات طراحی بر زمان اجرا ارجحیت دارد. هدف از پنهان سازی اطلاعات، محدود ساختن اصلاحات برنامه به یک پیمانه منفرد، در صورت نیاز به تغییر در تصمیمات طراحی است. به طور خلاصه، ایده پنهان سازی اطلاعات پاراناس، کامل کننده معنی استقلال پیمانه‌ها بود. یعنی نه تنها یک پیمانه باید شامل جنبه خاصی از یک مسأله باشد بلکه هیچ پیمانه دیگری نیز نباید به هیچوجه بجز از طریق پارامترها از آن جنبه اطلاع داشته باشد.

مقاله

نواده پاسکال

نوشته: دیک باونتین*
ترجمه آزاد: سعید وحید

پیاده سازی نرم افزارهای سیستم طراحی کرده ولی ابرن علاوه بر زبان برنامه نویسی، یک سیستم عامل و محیط اجرایی است. همانگونه که ماجولا-۲ محیط اجرایی برای ایستگاه کار لیلیت^۶ بود، ابرن نیز همین نقش را برای یک ایستگاه کار ۳۲ بیتی متصل به شبکه به نام سرز^۷ بازی می کند. سرز تماماً در دانشگاه ETH طراحی شده و دانشجویان دانشکده ای که ویرت در آن تدریس می کند، به طور گسترده از آن استفاده می کنند. هدف از طراحی سرز، ایجاد یک ایستگاه کار ساده، ارزان و قابل اطمینان بوده است و برای نیل به آن تنها بخش های ضروری سخت افزار در سیستم گنجانده شده ولی به هنگام نیاز می توان قسمتهای دلخواه دیگر را نیز به آن افزود.

در واقع منظور از ابداع ابرن تعمیم این دید ساده سازی سخت افزار به بخش نرم افزار است. ویرت اعتقاد راسخ دارد که سیستم عامل باید به صورت مجموعه ای از واحدها^۸ طراحی شود که هر کدام جداگانه کامپایل می شوند و فاصل ارتباطی^۹ آنها نیز دقیقاً مشخص است. آنگاه می توان سیستم عامل را با افزودن واحدهای جدید، یعنی برنامه های کاربردی که برایش نوشته می شود، توسعه داد. ماجولا-۲ تمام امکانات لازم برای تأمین واحد مندی^{۱۰} چنین سیستمی را فراهم می کند ولی به عقیده ویرت امکانات کافی برای گسترش توسط کاربر، مخصوصاً برای تعمیم گونه های^{۱۱} موجود و تعریف گونه های جدید را ندارد.

تعمیم گونه ها

تعمیم گونه ها امکان ساخت رکوردی از گونه جدید را بر مبنای یک گونه موجود، فراهم می آورد. مثلاً اگر گونه ای به نام Circle را به ترتیب زیر تعریف کرده باشیم:

TYPE Circle =
RECORD

x, y, radius : REAL

END;

Lilith ۶
Ceres ۸
module ۹
interface ۱۰
modularity ۱۱
type ۱۱

زمانی که دهها زبان برنامه نویسی با قابلیت های گوناگون طراحی شده و در اختیار برنامه نویسان قرار گرفته و در حالی که حتی برخی از آنها گوی سبقت را از دیگر رقیبان ربوده و پیروزمندانه بیکه تازی می کنند، چشم انداز موفقیت برای زبانی جدید چندان امیدوار کننده نیست. ولی چنانچه پیداست نورسیده ای را باید جدی گرفت: ابرن^۱.

این نامی است که نیکلاوس ویرت^۲ استاد علوم کامپیوتر در دانشگاه ETH^۳ زوریخ، به زبان برنامه نویسی جدید خود داده است. او این نام را هنگامی که فضایی می وجر^۴، در سال ۱۹۸۸، از کنار یکی از اقمار اورانوس به همین نام گذشت، برای تحسین دقت فوق العاده در مسیریابی فضاییما، برگزید.

ابرن شباهت زیادی به سایر اعضای

خانواده زبان های ابداعی ویرت دارد ولی

عملاً از پیشینیانش بسیار کوچکتر و

ساده تر است.

بدون شک ویرت یکی از تأثیر گذارترین طراحان زبان در تاریخ کوتاه این رشته به شمار می آید. او در ابتدا پاسکال را به عنوان زبانی برای آموزش برنامه نویسی ابداع کرد. سپس با توسعه آن و افزودن امکاناتی برای برنامه سازی همزمان^۵، زبان ماجولا-۲ را آفرید، و اکنون نیز با ابرن حرف های دیگری برای گفتن دارد. ابرن شباهت زیادی به سایر اعضای خانواده زبان های ابداعی ویرت دارد ولی عملاً از پیشینیانش بسیار کوچکتر و ساده تر است. کامپایلر این زبان برنامه ای به حجم ۱۳۰ کیلوبایت است که بعد از کامپایل شدن ۳۹ کیلوبایت از حافظه را اشغال می کند و برای کامپایل کردن خودش به ۴۱ ثانیه وقت نیاز دارد. هر چند ویرت ابرن را با الگو قرار دادن ماجولا-۲ و به عنوان زبانی برای

Pountain, Dick. "Oberon", Byte, vol.16(3), March 1991.

Oberon ۱

Niklaus Wirth ۲

Eidgenössische Technische Hochschule ۳

Voyager ۴

concurrent programming ۵

```
Circle= POINTER TO CircleObject;
CircleObject= RECORD (Object)
    radius: REAL
END;
```

قواعد سازگاری گونه‌ها در ابرن، انتساب اشاره‌گری از نوع Rectangle یا Circle را به‌متغیری از نوع Node امکان‌پذیر می‌کند و به‌این ترتیب می‌توان درختی ساخت که گره‌هایش از گونه‌های مختلف است. علاوه بر تعمیم گونه‌ها، یکی دیگر از برتری‌های ابرن نسبت به مایجولا-۲ تسامحی عمده در اعمال قواعد سازگاری گونه‌هاست. یعنی اگر گونه T تعمیم یافته و در برگرفته گونه T' باشد، مقادیری از نوع T'، از نوع T نیز هستند و می‌توان آنها را به متغیرهایی از نوع T نسبت داد. به‌عنوان مثال زبان ابرن پنج گونه عددی دارد که به‌طور سلسله مراتبی همدیگر را در بر می‌گیرند (شکل ۱).

به‌این ترتیب همیشه می‌توان متغیری از نوع INTEGER را به‌متغیری دیگر از نوع REAL یا یک SHORTINT را به INTEGER نسبت داد. این روش نیاز به‌توابعی برای تبدیل گونه‌ها را رفع می‌کند و برخی از جنبه‌های محدود کننده مایجولا-۲ مانند ناسازگاری بین گونه‌های INTEGER و CARDINAL و REAL را رفع می‌کند.

LONGREAL



REAL



LONGINT



INTEGER



SHORTINT

شکل ۱ - شمول گونه‌های عددی در ابرن

ساده‌سازی مایجولا-۲

بعضی از ویژگی‌ها و خصوصیات مایجولا-۲ از ابرن حذف شده و در عوض توانایی‌های دیگری به‌آن افزوده شده است. ولی کلاً مواردی که حذف شده‌اند بسیار بیشتر از ویژگی‌های جدیدی است که به‌زبان اضافه شده‌است. این خصوصیات از این جهت در ابرن گنجانده نشده‌اند که یا زبان را آسیب‌پذیر می‌ساختند و یا زائد بودند و کامپایلر را بیش از حد پیچیده می‌کردند.

گونه‌های اسکالر مثل Colours=(red, blue, green) - که در اصل برای بالا بردن وضوح برنامه در پاسکال گنجانده شده بودند، در

می‌توان آن را چنین تعمیم داد:

```
FilledCircle =
RECORD (Circle)
    fillcolour : INTEGER
END;
```

رکوردهای از گونه جدید، حوزه‌های x, y و radius را از "گونه پایه" به‌ارث می‌برد و حوزه جدید خود fillcolour را نیز به‌آن اضافه می‌کند. به‌این ترتیب می‌توان به‌طریقی مطمئن و امن به‌همان هدفی که پاسکال و مایجولا-۲ توسط رکوردهای با طول متغیر^{۱۳} قصد رسیدن به‌آن را دارند، ناقل آمد.

رکوردهای با طول متغیر از نقاط ضعف پاسکال و مایجولا-۲ محسوب می‌شوند و بسیار مورد انتقاد قرار گرفته‌اند. چرا که با این ویژگی برنامه‌ساز می‌تواند با دادن چند نام به‌یک محل از حافظه^{۱۴} قواعد بکارگیری گونه‌ها را که کامپایلر در سایر موارد آن را شدیداً کنترل می‌کند، به‌راحتی زیر پا گذارد. علاوه بر این استفاده از رکوردهای با طول متغیر انجام بعضی از وظایف سیستم عامل، مثلاً پاکسازی و مرتب کردن حافظه^{۱۵} را نیز دشوار می‌کند. از این رو رکوردهای با طول متغیر در زبان ابرن اصلاً وجود ندارند.

تعمیم گونه‌ها برای اشاره‌گرها^{۱۶} نیز قابل اعمال است. ویرت مثال زیر را در مقاله‌ای که تحت عنوان "از مایجولا تا ابرن" نوشته، آورده‌است و هدف آن تعریف ساختمان داده‌های لازم برای ساختن یک درخت است:

```
TYPE Node = POINTER TO Object;
Object = RECORD
    key, x, y : INTEGER;
    left, right : Node
END;
```

یکی از ویژگی‌های اساسی در برنامه‌سازی به‌زبان ابرن این است که می‌توان گونه‌ای را از یک واحد به‌واحد دیگر منتقل^{۱۷} کرد و درون واحد جدید تعمیمش داد. پس مثلاً می‌توان در واحدی دیگر، گونه Object را به‌گونه‌هایی مفیدتر تعمیم داد:

```
TYPE Rectangle = POINTER TO RectObject;
RectObject= RECORD (Object)
    width, height : REAL
END;
```

- field ^{۱۱}
- variant record ^{۱۲}
- aliasing ^{۱۳}
- garbage collection ^{۱۴}
- pointer ^{۱۵}
- import ^{۱۶}

ابرن

ویرت نام زبان برنامه‌نویسی خود را از نام یکی از اقامات اورانوس گرفته، اما ابرن در افسانه‌های اساطیری، شاه پریان و همسر تیتانیا است. در یک تصنیف آلمانی در نقش نگهبان گنج کوتوله‌های نیلونگ ظاهر می‌شود و در فولکلور فرانسه، در یکی از منظومه‌هایی که فتوحات شوالیه‌ها را می‌سراید از او به عنوان یاور عشاق یاد می‌شود. اسینسر و شکسپیر او را موضوع برخی از آثار خود قرار داده‌اند و کارل ماریا فون وبر نیز اپرایی بر این اساس ساخته است.

بر طبق حماسه‌ای به همین نام که مارتین ویلاند، شاعر آلمانی قرن هیجدهم، سروده ابرن از همسرش، تیتانیا، به خاطر این که او از زنی جفاکار حمایت می‌کرده، جدا می‌شود و پیمان می‌بندد تا زمانی که زوج وفاداری را نیابد، نزد او باز نخواهد گشت. در همین حین، شوالیه‌ای به جرم قتل فرزند امپراتور محکوم می‌شود که به بغداد رفته و چهار دندان و دسته‌ای از ریش خلیفه را پس از بوسیدن و کشتن دختر خلیفه، برای امپراتور ببرد. ابرن شوالیه را انتخاب کرده و به یاری او می‌شتابد.

به کمک شیپور جادویی ابرن، که صدای آن دشمنان را به رقص وامیدارد، سرانجام شوالیه موفق می‌شود. اما تسلیم و سوسه‌ای شده و به فرمان ابرن که از او خواسته بود قبل از زفاف با دختر خلیفه بهرم نزد پاپ برود، گردن نمی‌نهد. ابرن آنها را به صحرا می‌افکند. در تونس آن دو به واسطه تیتانیا به اسارت در می‌آیند اما اسیر هوس نشده و به هم وفادار می‌مانند. به این ترتیب چون در امتحان موفق می‌شوند، به اتفاق هم به پاریس برگشته و با امپراتور آشتی می‌کنند. ابرن نیز نزد تیتانیا بازمی‌گردد.

منبع: "Oberon", Encyclopedia Americana, vol 20, p. 591, 1975.

موضوع‌گرایی در ابرن

اگرچه ویرت و همکارانش مخالف کارهایی بودند که در شرکت‌های دیجیتال و الیوتی صورت می‌گرفت و در نهایت به اعمال تغییر در مایجولا-۲ و عرضه مایجولا-۳ به عنوان زبانی با قابلیت‌های موضوع‌گرایانه^{۱۸} انجامید، ولی با این وجود رگه‌هایی از موضوع‌گرایی در زبان ابرن نیز مشاهده می‌شود: قابلیت تعمیم گونه‌ها که در ابرن وجود دارد یک روی سکه از روش برنامه‌سازی موضوع‌گراست. روی دیگر که امکان تعمیم رویه‌ها و روش‌هاست در ابرن دیده نمی‌شود، زیرا ویرت هنوز کاملاً متقاعد نشده که این شیوه بهترین الگو برای برنامه‌سازی سیستم‌های بزرگ است. در روشهای موضوع‌گرا تأکید بر این است که

زبان ابرن وجود ندارند. ویرت معتقد است که بی‌دقتی در استفاده از آنها باعث زیاد شدن دستورات اعلانی و در نتیجه تولید برنامه‌هایی می‌شود که بی‌جهت طولانی است. گونه‌های اسکالر برای کامپایلر نیز مشکل آفرین هستند چون مقداره‌ای آنها وضعیتی استثنایی و غیر عادی دارد: نه مانند متغیرهای عددی مقدار خاصی به خود می‌گیرند و نه این که می‌توان با آنها مثل نویسه‌ها رفتار کرد. علاوه بر این قابل خواندن و نوشتن هم نیستند.

گونه‌هایی نیز که به صورت محدوده کوچکتری^{۱۹} از یک گونه دیگر تعریف می‌شوند. مثل `MonthRange=1..31` - از زبان ابرن حذف شده‌اند. هدف اولیه از وارد کردن آنها به زبان استفاده اقتصادی‌تر از حافظه و امن‌تر کردن دستورات انتسابی بود. ولی اکنون به عقیده ویرت مزایای آنها ارزش پیچیده کردن کامپایلر را ندارد. همچنین در زبان ابرن دیگر تغییری از نوع مجموعه‌هایی که توسط برنامه‌نویس تعریف می‌شد - مثل `'Z'..'A' = LetterSet` - وجود ندارد و به جای آنها فقط یک گونه SET در نظر گرفته شده که مقادیر آن مجموعه‌ای از اعداد صحیح است.

رکوردهای با طول متغیر از نقاط ضعف

پاسکال و مایجولا-۲ محسوب می‌شوند و بسیار مورد انتقاد قرار گرفته‌اند.

اشاره‌گرها فقط برای متغیرهایی از نوع رکوردها و آرایه‌ها قابل استفاده هستند. زیرنویس آرایه‌ها دیگر قابل تعریف نیست و حتماً باید از نوع اعداد صحیح باشد. کران پائین آرایه‌ها صفر است و دستور اعلانی یک آرایه مثلاً به این شکل است: `ARRAY 10 OF INTEGER`. مزیت این روش ساده‌تر کردن کنترل کران آرایه‌ها، مخصوصاً برای آرایه‌های پویاست.

حلقه‌های FOR کاملاً حذف شده‌اند و به جای آنها باید از REPEAT یا WHILE استفاده کرد. دستور WITH هم که در پاسکال و مایجولا-۲ برای خلاصه‌نویسی اسامی حوزه‌های یک رکورد به کار می‌رفت، در ابرن به منظور دیگری استفاده می‌شود و نام حوزه‌ها و رکوردها باید به طور کامل ذکر شود. و سرانجام اینکه برخلاف مایجولا-۲ در ابرن نمی‌توان از برنامه‌سازی همزمان سود جست. ویرت در توجیه این مطلب با تأکید می‌گوید که حذف این ویژگی به معنی رد کردن برنامه‌سازی همزمان نیست بلکه در پروژه ابرن-سرز از ابتدا قصد بکارگیری این روش مطرح نبوده است.

به طور کلی می‌توان گفت که هدف از این تغییرات، ساده‌تر و متقارن‌تر کردن قواعد ابرن برای کار با واحدها بوده است.

چه کسی انجمن را اداره می‌کند؟

عضویت در هیئت اجرایی انجمن موروئی و مادام‌العمر نیست. این هیئت را هر دو سال یک‌بار اعضای انجمن انتخاب می‌کنند. انتخابات دوره بعدی هیئت اجرایی در مهر ماه امسال برگزار خواهد شد. اگر در خود انگیزه‌ای برای هدایت انجمن و طرح و نظری برای بهبود وضع انفورماتیک در ایران و توانی برای فعالیت در راه پیشرفت فرهنگ کامپیوتری سراغ دارید، نامزد عضویت در هیئت اجرایی شوید. انجمن را باید اعضایش اداره کنند، یعنی

شما!

برای اطلاع از شرایط انتخاب و اعلام نامزدی با آقای ابراهیم تقیب‌زاده مشایخ یا تلفن ۲۴۳۸۶۰ تماس بگیرید، یا به آدرس پستی انجمن نامه بنویسید.

تمام رویه‌ها باید در کنار ساختمان داده‌هایی که روی آنها کار می‌کنند، تعریف شوند و این به عقیده ویرت بسیار محدود کننده است. ابرن در نبود ویژگی وراثت رویه‌ها، برنامه‌نویس را ملزم می‌کند اگر قصد استفاده از رویه‌ای از واحدی دیگر را دارد باید آن را به‌طور صریح به‌واحد فعلی انتقال دهد. با برگشت به‌مثالی که قبلاً ذکر شد، برنامه‌ساز می‌تواند Object را از واحدی دیگر گرفته و آن را در واحدی جدید به RectObject یا CircleObject تعمیم دهد و در این واحد رویه‌های جدیدی را برای کار روی این گونه‌های تعمیم یافته، تعریف کند. ولی اگر بخواهد از رویه‌هایی که در واحد اصلی برای کار روی Object وجود داشت استفاده کند در این صورت باید آن‌ها را صریحاً وارد واحد جدید کند.

قابلیت تعمیم گونه‌ها که در ابرن وجود دارد یک روی سکه از روش برنامه‌سازی موضوع‌گراست. روی دیگر که امکان تعمیم رویه‌ها و روش‌هاست در ابرن دیده نمی‌شود.

محیط‌های اجرایی

دانشگاه ETH اخیراً نسخه‌هایی از ابرن را که روی کامپیوترهای مکینتاش، ایستگاه‌کار شرکت دیجیتال^{۲۰} و ایستگاه‌کار شرکت سان^{۲۱} قابل اجرا هستند، در اختیار عموم قرار داده است. از آنجا که در ابرن تمام واحدها هم‌رتبه و برابرند و می‌توان آن‌ها را جداگانه کامپایل و اجرا کرد، بنابراین هر رویه بدون پارامتر متعلق به یک واحد را می‌توان به‌منزله یک "فرمان" با تایپ کردن اسمش اجرا کرد (مثل MyModule.Start). روش فراخوانی برنامه‌ها نیز به‌همین ترتیب است. حال اگر قرار باشد تحت سیستم عامل ام‌اس-داس نیز برای ابرن کامپایلرهایی تهیه شود، این ویژگی مشکل‌آفرین خواهد بود. زیرا که در سیستم عامل داس اجرای بخشهایی از یک فایل EXE، به‌ترتیبی که اشاره رفت، امکان‌پذیر نیست.

^{۲۰} DECstation

^{۲۱} Sparcstation

مقاله

مباحثی در برنامه‌سازی موازی

نوشته: تی.جی. لوئیس*
ترجمه: سعید داعی حق

مشکلات پردازش موازی

برنامه‌سازان علاوه بر تعاریف و اصول معمول در برنامه‌سازی ترتیبی، باید به موارد عدم قطعیت، وضعیت مسابقه، همزمانسازی و زمانبندی آگاهی داشته باشند.

به دلیل تمایل انسان به تفکر ترتیبی به جای موازی، اغلب برنامه‌ها تاکنون به طور طبیعی با زبان‌های ترتیبی تولید شده‌اند. متأسفانه، برنامه‌سازی ترتیبی قادر به استفاده مؤثر از کامپیوترهای موازی نیست. "گامپایله‌های موازی‌کننده" عملاً وجود ندارند، یا اگر هم موجود باشند تنها توسط الگوریتمهای موازی که منجر به برنامه‌ای موازی می‌شوند، بیشترین بهره را بدست می‌دهند. ابزارهای ارائه شده جهت "تفکر موازی" دورنمای امید بخشی را از پیشرفت در آینده به ما نشان می‌دهند و به ضرورت وجود نوعی زبان برنامه‌سازی موازی اشاره می‌کنند که هنوز در عمل بوجود نیامده است.

این مفاهیم مستلزم تعمق بسیار و مهارت است. دهها سال تولید نرم‌افزار را باید به کناری گذاشته و راهی طولانی را در بازنویسی نرم‌افزارها به شکل موازی طی نمود و یا ابزاری به منظور تبدیل آنها به نحوی که در کامپیوترهای موازی آینده قابل استفاده باشند، تولید کرد.

قابلیت انتقال و زمانبندی

حتی اگر زحمت تولید مجدد نرم‌افزارهای گذشته را برای استفاده در کامپیوترهای موازی بپذیریم، مسئله قابلیت انتقال، مشکلات دیگری را ایجاد می‌کند. برنامه‌سازی موازی بدون استفاده از ابزاری خاص از بعضی جوانب مشابه نوشتن برنامه‌ی ترتیبی توسط یک زبان سطح پایین است چون نوشتن برنامه‌های موازی بصورت مستقیم و بدون ابزار، آنها را وابسته به معماری ماشین می‌کند. برای مثال همزمان‌سازی در سطح پایین و نزدیک به ماشین در یک سیستم با حافظه از نوع معماری مشارکتی، بوسیله مسدود کردن و در معماری توزیعی به وسیله انتقال پیام انجام می‌گیرد.

تعجبی ندارد که طراحان برنامه برای اینکه بتوانند الگوریتمهای خود را در یک ساختار سطح بالاتر و بدون نگرانی از جزئیاتی مثل همزمان‌سازی بیان کنند به امکانات تجرید بیشتری در رابطه با استقلال از معماری ماشین نیازمند هستند. محققین، بسیاری از این نوع روشهای تجرید را پیشنهاد کرده‌اند، اما فقط تعداد کمی از آنها را مجریان فن بکار گرفته‌اند.

امروزه توافق عمده‌ای در دنیای کامپیوتر وجود دارد مبنی بر اینکه بیشترین بهره از نقطه نظر هزینه و کارایی از بکارگیری سیستمهای موازی چند پردازنده‌ای بدست می‌آید. این سیستمها از دو یا چند پردازنده و حافظه تشکیل شده‌اند که توسط شبکه‌ای با یکدیگر در ارتباط هستند. بر خلاف تک‌پردازنده‌های برداری یا لوله‌ای^۱، کارایی نسل جدید کامپیوترهای موازی ناشی از تعدد پردازنده‌هایی است که لزوماً چندان سریع هم نیستند.

سیستمهای پردازش موازی با اینکه در ابتدای مراحل توسعه هستند، با بسیاری از سریعترین ابر کامپیوترهای حاضر رقابت می‌کنند. برای مثال، اخیراً پژوهشگران شرکت نفتی موبیل^۲ در یک سیستم کاربردی مربوط به ارتعاشات زمین به سرعتی معادل شش گیگافلاپ روی یک "ماشین اتصال"^۳ دست یافته‌اند.

سرمايه‌گذاری سخت‌افزار را نیز میتوان در رده‌های پائین‌تر مشاهده کرد. برخی از واحدهای تسريع کننده که دارای چندین پردازنده هستند، میتوانند کارایی شبکه‌ای از ماشین‌های رومیزی را تا سطحی نزدیک به ابر کامپیوترها افزایش دهند. مشخصات این سیستمها از لحاظ قیمت و کارایی بدانگونه است که می‌توان آنها را با معیارهای یک دهه قبل در رده ابر کامپیوترها قرار داد.

تکنولوژی سخت‌افزار همراه با جنبه‌های اقتصادی مربوطه، آماده ایجاد تحولی عظیم در کلیه سطوح در بازار است. اما اگر پردازش موازی تا بدین حد دارای مزایا و محاسن است، چرا از آن استفاده نمی‌کنیم؟ جواب این سوال را در نبود نرم‌افزار مناسب برای این‌گونه سیستمها می‌توان جستجو کرد.

ویژگیهای برنامه‌سازی موازی

تولید نرم‌افزار، چه در مورد کاربردهای با پردازش ترتیبی و چه موازی، از نظر ماهیت فعالیتی دشوار و زمانگیر است. ولی طراحی و ساخت نرم‌افزار برای کامپیوترهای موازی کاری دشوارتر است چون این

* T.G. Lewis, "Issues in Parallel Programming", *Super-computing Review*, July 1990

۱ pipeline
۲ Mobil Oil
۳ connection machine

```

PAR 10 ij=1,N*K ( N*K گذر را به طور موازی انجام بده )
j=(ij-1) DIV N+1
i=(ij-1) MOD N+1
S(i,j) ( هر S با دیگر S ها موازی است )
10 CONTINUE

```

شکل ۱

کلمه کلیدی PAR اجرای موازی حلقه‌های تکرار را نشان می‌دهد. حال فرض می‌کنیم عبارتهای داخل بدنه حلقه شدیداً بیکدیگر وابسته باشند. برای مثال $S(i,j+1)$ برای $j=1, \dots, K-1$ به $S(i,j)$ وابسته باشد. از آنجائیکه بررسی دو حلقه تکرار تودرتو در این رابطه مشکل است، لذا بهتر است یک حلقه تکرار را که حاوی K عبارت و هر عبارت وابسته به شاخص "i" است، پیگیری کنیم.

```

DO 20 i=1, N
  S(i,1); (اول باید دستور S(i,1) اجرا شود)
  S(i,2); (دستور S(i,2) باید بعد از S(i,1) اجرا شود)
  ...
  S(i,K); (دستور S(i,K) به S(i,K-1) وابسته است)
20 CONTINUE

```

با فرض اینکه p پردازنده تحت کنترل دستورهای SIGNAL, WAIT بطور همزمان با هم مشغول بکار باشند، میتوان حلقه تکرار بالا را بصورت موازی در آورد. (شکل ۲)

در این مثال، دستور $SIGNAL(i,j)$ خاتمه اجرای عبارت $S(i,j)$ را اطلاع می‌دهد و دستور $WAIT(i,j)$ عبارتی که به $S(i,j)$ وابسته است را تا هنگام دریافت $SIGNAL(i,j)$ به تعویق می‌اندازد. شکل ۲ موازی سازی یک حلقه تکرار را با $N=8$ (دفعات تکرار حلقه) $K=3$ (تعداد عبارات در بدنه حلقه) و $p=5$ (تعداد پردازنده‌ها) نشان می‌دهد.

این راه حل تنها در مورد سیستمهای چند پردازنده‌ای با حافظه مشترک معتبر است و با در نظر گرفتن مشکلات عمومی، که در موازی سازی الگوریتمهای ترتیبی مطرح است، بسیار محدود کننده است. با توجه به مشکلات مربوط به حالات کلی الگوریتمها، امکان یافتن روشی کلی برای تبدیل الگوریتمهای ترتیبی به موازی مورد تردید است.

دومین مشکل اساسی، زمان بندی یا نحوه تقسیم کار میان پردازنده‌ها و تعیین ترتیب اجرای آنها است. برنامه ساز به چه ترتیبی باید اجرای اعمال آماده اجرا را در یک کامپیوتر موازی برنامه ریزی کند تا به کوتاه ترین زمان اجرا دست یابد. از دیدگاه ریاضیات، حل این مسئله کاری دشوار است و غالباً برای اینکه بهترین زمان بندی را پیدا کنند، زمان زیادی صرف میشود. در یک سیستم کاربردی چنانچه زمان بندی بیشتر از اجرای خود آن سیستم طول بکشد، به اهداف پردازش موازی لطمه وارد شده است.

بنابراین، در برنامه سازی موازی علاوه بر کلیه مشکلات برنامه سازی سنتی، برنامه ساز باید با جزئیات معماری جدید و برنامه ریزی اجرای کارهای موازی در محیطی با چند پردازنده آشنا باشد. همچنین در برنامه سازی موازی نیاز به ابزاری برای کمک در کلیه سطوح مثل عیب یابی، تجزیه و تحلیل کارایی و اجرا و قابلیت استفاده مجدد از اجزاء احساس می‌شود. این ابزارها ناسازگاری موجود بین طراحان برنامه های ترتیبی و کامپیوترهای موازی را که ناشی از عدم شناخت برنامه نویسان از سیستمهای موازی است، کاهش می‌دهند.

در زبانهای برنامه سازی برای کامپیوترهای موازی دو نگرش وجود دارد که هر کدام دارای معایب و مزایائی هستند:

- موازی سازی ضمنی

- موازی سازی صریح

در ادامه مقاله به تشریح این دو نگرش خواهیم پرداخت.

موازی سازی ضمنی

در روش موازی سازی ضمنی از زبانهای موجود استفاده می‌شود که خصوصیات کامپیوترهای موازی را از دید برنامه ساز پنهان می‌کنند. برای ترجمه برنامه های کاربردی نوشته شده به زبانهای سطح بالا به شکلی موازی، باید کامپایلرهای هوشمندی برای این زبانهای سطح بالا طراحی شوند. این روش با وجودیکه در سیستمهای برداری موفق بوده است، اما بنظر میرسد خارج از توان طراحان کامپایلر برای کامپیوترهای واقعاً موازی است. برنامه زیر یکی از مشکلات موازی سازی یک برنامه ترتیبی را نشان می‌دهد. برای مثال اگر عبارت $S(i,j)$ به چرخش حلقه های تکرار وابستگی نداشته باشد حلقه های تکرار تودرتوی زیر را می‌توان به $K * N$ عملیات موازی تبدیل کرد. (وابستگی به چرخش حلقه تکرار بدان معنی است که چرخش نام یک حلقه تکرار به چرخش پیش از آن، یعنی چرخش نام بستگی دارد. این مثالها به زبان فورتن ۷۷ نوشته شده اند.)

```

DO 20 i=1, N
  DO 10 j=1, K
    S(i,j) (دستور S به i و j وابسته است)
  10 CONTINUE
20 CONTINUE

```

در زبان استرند ۸۸ به این شکل است:

DoFunction (A,N):-

N>0|

AI is N

putarg (AI,N,A),

DoFunction(A,M)@next,

M is N-1.

شکل ۳

فراخوانی مجدد به جای تکرار، مقدار دادن به یک متغیر با هر دستور و پردازش لیست به جای آرایه از نکات کلیدی برای درک برنامه فوق است. اول اینکه DoFunction یک روال خود فراخوان است و هر بار فراخوانی آن یک تکرار از حلقه را انجام می دهد. برای پیاده سازی N بار چرخش حلقه تکرار، DoFunction باید N بار فراخوانده شود و از این رو، این روال خودش را صدا می زند.

دوم اینکه شرط خاتمه حلقه هنگامی برقرار می شود که N=0 شود در نتیجه حلقه به ازای مقادیر N, N-1, N-2, ... تکرار می شود. این امر توسط شرط N>0 که در ابتدای بدنه روال DoFunction قرار گرفته، تحقق می یابد. زمانی که N=0 شود فراخوانی مجدد متوقف شده و محاسبات پایان می پذیرد.

سوم اینکه عبارت "A(i)=i" با مقدار دادن به AI که بصورت "AI is N" نشان داده شده و سپس بوسیله دستور "Putarg" که عنصری از A را در محل N ام لیست اضافه می کند، عملی می شود. توجه داشته باشید که N در هر فراخوانی توسط عبارت "M is N-1" کاهش می یابد. بنابراین هر بار که DoFunction فراخوانده می شود N با N-1 جایگزین شده و بنابراین مقادیر بعدی AI عبارتند از (N-1), (N-2), ...، الی آخر.

فراخوانی مجدد روال "DoFunction(A,m)@next" موجب می شود که پردازشگر استرند ۸۸ برای هر حلقه تکرار، یک فرآیند جدید تولید کند. عبارت "@next" سبب می شود که فرآیند روی عنصر بعدی اجرا شود. همچنین هر جزء یک تابع استرند ۸۸ بصورت موازی اجرا می شود، اما چون عبارت "@next" برای آنها وجود ندارد اغلب عبارتها در سطحی همسان با فرآیند پدر اجرا می شوند. در نتیجه استرند ۸۸ موازی سازی را در سطح دستورات تامین می کند. شکل ۳ موازی سازی و چگونگی دستیابی به آن را نشان می دهد.

در شکل ۳، عبارت های "AI is N" و "Putarg" و "M is N-1" موازی اجرا می شوند و همزمان به مقادیری یکسان دست می یابند. برای مثال، AI در عبارت نخستین تعریف شده و در عبارت "Putarg" استفاده می شود. همینطور M که در آخرین عبارت تعریف شده، اما توسط عبارتی که قبل از تعریف M قرار گرفته به DoFunction

PAR 20 new=1,P

if N≤P then M=0 else M=N MOD P+P

U=N-M

DO 10 i=new,U,P

S(i,1);S(i,2); ... S(i,K)

10 CONTINUE

DO 15 ij=new,M*K,P

i=(ij-1) MOD M+1

j=(ij-1) DIV M+1

CASE j OF

1: S(i+U,j);SIGNAL(1,i)

2: WAIT(1,i);S(i+U,2)

SIGNAL(2,j)

...

K: WAIT(K-1,i);S(i+U,K)

ENDCASE

15 CONTINUE

20 CONTINUE

شکل ۲

زبان استرند ۸۸

روش موازی سازی ضمنی دیگری که از بلندپروازی کمتری برخوردار است طراحی یک زبان جدید می باشد که موازی سازی را از دید برنامه نویسنده پنهان ساخته و در عین حال، کار نویسنده کامپایلر را آسان تر می سازد. استرند ۸۸^۴ زبان برنامه سازی جدیدی است که در آن، برنامه به شکل ترتیبی بیان شده و به صورت موازی اجرا می شود. یک برنامه به زبان استرند ۸۸ مانند برنامه پرولوگ از مجموعه ای عبارات تشکیل شده است که عملکرد آن تنها بر اساس یک دستور انتساب مقادیر به متغیرها، برای ذخیره سازی اطلاعات و نیز مدلی جهت همزمان سازی و تقسیم عملیات بین پردازنده ها بنا شده است.

استرند ۸۸ به جای حلقه از فراخوانی مجدد و بجای آرایه از لیست استفاده می کند. در نتیجه برنامه زیر از دید برنامه سازان فرتن شکلی عجیب دارد. برای مثال معادل حلقه تکرار زیر در زبان فرتن،

DO 20 i=1,N

A(i)=i

20 CONTINUE

CHAN OF INT north, south:

PAR (Do T1 and T2 in parallel)
 INT x: (T1 ...)
 SEQ (Do T1.1 and T1.2 sequentially)
 north ! 2 (T1.1 sends to T2.1)
 south ? x (T1.2 receives from T2.2)
 INT y: (T2 ...)
 SEQ (Do T2.1 and T2.2 sequentially)
 north ? y (T2.1 receives from T1.1)
 south ! 3 (T2.2 sends to T1.2)

شکل ۴

روش موازی سازی صریح به برنامه ساز "باهوشتری" نیاز دارد. نظریه اینکه تولید ابزار و تکنیک هایی که به برنامه ساز کمک کند تا هوشمندانه تر عمل کند از تولید کامپایلری هوشمند ساده تر است، از اینرو پیشرفت های بیشتری در روش های صریح موازی سازی حاصل شده است. اغلب اینگونه روشها از راه توسعه زبانهای ترتیبی موجود، تحقق یافته اند. برای مثال، شکل ۵ چگونگی تبدیل زبان سی به یک زبان برنامه سازی موازی را نشان میدهد.

برنامه نویسی برای کامپیوترهای موازی مشکل تر از برنامه نویسی برای ماشینهای تک پردازنده ایست و نیاز به مهارت بیشتری دارد.

در این مقاله سعی شده تا با مثالهایی ساده نشان داده شود که برنامه نویسی برای کامپیوترهای موازی مشکل تر از برنامه نویسی برای ماشینهای تک پردازنده ایست و نیاز به مهارت بیشتری دارد. بطور کلی طبیعت روند تولید نرم افزار هنوز ناآشناست و میزان رشد تولید نرم افزار طی چند سال گذشته بسیار پائین بوده است. با وجود پیشرفتهایی که در زبانهای برنامه سازی صورت گرفته، تولید نرم افزار برای ماشینهای موازی هنوز کار دشواری است.

پیشنهادهای

برنامه سازی ترتیبی با زبانهای سطح پائین وابسته به ماشین شروع شد. سپس زبانهای سطح بالاتری بمنظور ساده سازی کار برنامه سازانی که نمیخواستند نگران جزئیات سخت افزار باشند بوجود آمد و بالاخره

انتقال داده می شود. آیا مسابقه ای که بین عبارات موازی واقع می شود موجب نامشخص شدن نتایج نمی گردد؟ پاسخ منفی است.

در استرنند ۸۸ با استفاده از ناحیه ای برای نگهداری داده ها، که در ابتدا خالی است، از وقوع وضعیت مسابقه اجتناب می شود. از زمان شروع یک فرآیند، تا زمانیکه تمام اطلاعات اولیه ورودی آن در آن ناحیه فراهم نشده است، اجرای آن متوقف میماند. بنابراین، اجرای عبارت DoFunction, Putarg تا زمانیکه AI برای Putarg و M برای DoFunction (A,M) آماده نشده باشد، متوقف میماند. برای مثال، زمانیکه AI تعریف می شود در این ناحیه داده ای قرار گرفته و سپس توسط Putarg برداشته می شود. تا زمانیکه عبارت Putarg مقدار AI را به آن ناحیه برنگرداند هیچ فرآیند دیگری نمی تواند به مقدار آن دسترسی یابد.

به طریقی مشابه، M پس از اینکه مقدار (N-1) را به خود گرفت، در آن ناحیه قرار می گیرد. سپس روال DoFunction, "M" را از آن ناحیه برداشته و از آن برای محاسبات تکرار بعدی حلقه استفاده می کند.

استفاده از یک ناحیه داده ای، به این شکل، موجب می شود که یک برنامه استرنند ۸۸ همزمانی را در خود ایجاد کند، اما جریان داده ها را در برنامه محدود می کند. برای مثال، روشی جهت تعریف یک متغیر مشترک که یک یا چند فرآیند بتوانند مقدار آن را فقط بخوانند، وجود ندارد. این امر در اجرا و کارایی برنامه تاثیر می گذارد.

موازی سازی صریح

در روش موازی سازی صریح، زبان برنامه سازی مربوطه دارای دستورات کنترلی برای موازی سازی است و برنامه ساز باید در مورد موازی سازی آگاهی داشته باشد. به عنوان مثال، زبان اکام^۶ برای برنامه سازی موازی صریح طراحی شده است. یک نمونه از برنامه اکام در شکل ۴ چگونگی انتقال داده ها را بین چهار فعالیت موازی T2.2, T2.1, T1.2, T1.1 نشان می دهد.

مجموعه های دستورات در اکام توسط تورفتگی عبارت نشان داده می شوند. کلمه SEQ نشانگر عملیات ترتیبی و PAR نشانه عملیات موازی است. در برنامه نمونه فوق، دو گروه دستورات که با کلمه INT مشخص شده اند موازی اجرا می شوند و مجموعه دستورات ترتیبی دیگر (عملیات مجرای^۷ شمال و جنوب) در داخل هر دو مجموعه عملیات موازی تعریف شده اند. پیامهایی بین واحدهای عملیاتی برنامه انتقال پیدا می کنند: عملگر "؟" به معنی دریافت و عملگر "!" به معنی ارسال است.

data pool ۵
 OCCAM ۶
 CHANnel ۷

سرگرمی

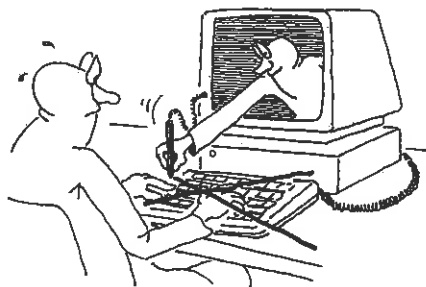
شماره خانه

گیتی در خیابانی زندگی می‌کند که خانه‌های آن از ۳ تا ۹۹ شماره‌گذاری شده‌اند و دوست او فریده می‌خواهد نشانش را به دست آورد. احمد، بیژن، پرویز، تیمور، و ثریا هر یک اطلاعی محدود درباره شماره خانه گیتی دارند:

- احمد می‌داند که این شماره مربع کامل است یا خیر.
- بیژن می‌داند که این شماره بر ۳ بخش‌پذیر است یا خیر.
- پرویز می‌داند که این شماره تک رقمی است یا دو رقمی.
- تیمور می‌داند که این شماره بر ۵ بخش‌پذیر است یا خیر.
- ثریا می‌داند که این شماره فرد است یا زوج.

فریده اطلاعات احمد، بیژن، و پرویز را از آنها می‌گیرد و با خود چنین فکر می‌کند که اگر تیمور هم کمکش کند، ممکن است بتواند شماره خانه گیتی را دقیقاً مشخص سازد. تیمور که در ابتدا سکوت اختیار کرده بود، بالاخره به حرف می‌آید و اطلاع خود را به فریده می‌دهد. اما فریده مجبور می‌شود که با ثریا نیز مشورت کند. عصر همان روز، فریده تصمیم می‌گیرد که سری به گیتی بزند و با اطمینان کامل به سراغ خانه‌ای که شماره‌اش را به دست آورده بود می‌رود. از ظاهر خانه چنین بر می‌آید که سالها است کسی در آن زندگی نکرده است و فریده دست از پا درازتر به خانه خود باز می‌گردد. روز بعد، پرویز، تیمور، و ثریا به فریده اطمینان می‌دهند که اطلاعات درستی در اختیارش گذاشته‌اند، اما احمد و بیژن که دروغ گفته بودند هیچ نوع واکنشی از خود نشان نمی‌دهند. شماره خانه گیتی چند است؟

گردآوری از بهروز پرهامی



امروز، تغییرات و تکامل زبانهای سطح بالا در جهت آن است که هر چه بیشتر ساخت یافته و انتزاعی شوند و برنامه‌های تولید شده از نظر آزمایش و نگهداری ساده‌تر باشند.

برنامه‌سازی موازی باید چنین روند تکاملی را طی کند و بسیاری از مشکلات موجود بر سر راه آن قابل رفع است، اما برای این کار باید شناخت دقیقی از زبانهای برنامه‌سازی و همچنین سخت‌افزار این ماشینها وجود داشته باشد. در نتیجه هنوز کار برنامه‌سازی موازی کاری پر زحمت است و نیاز زیادی به کمک‌های خودکار و نیمه‌خودکار در این امر مشاهده می‌شود.

اما این ابزارهای کمکی به چه شکلی باید موجود باشند؟ در برنامه‌سازی موازی نیاز به ابزاری مستقل از معماری ماشین است تا بتوان

```
shared int x;          /* to be shared */
shared struct yStruct{
    int state;
    struct lock lockp;
}y
main()
{
    s_init_lock(&y.lockp); /* init. primitive lock */
    m_set_procs(nprocs); /* initialize processes */
    m_fork(worker);       /* create proc. & start */
    m_kill_procs();       /* kill workers */
}

Worker()
{
    /* parallel tasks */
    m_lock();             /* begin critical section ... */
    x++;                  /* update shared data */
    m_unlock();           /* ... end of critical section */
}
```

شکل ۵

قسمتهای مختلف برنامه‌های ترتیبی را به برنامه‌هایی کاملاً موازی، کاراً و همزمان تبدیل کرد. چنین نحوه‌نگرشی این مزیت را نیز دارد که قابلیت انتقال در حوزه معماریهای مختلف کامپیوترهای موازی را امکان‌پذیر می‌کند.

کامپیوتر و بهداشت

استفاده درست از صفحه کلید

نوشته: مارک شی هن*

ترجمه: مهسا تاجبخش

می شوند. وقتی وترها متورم شدند به عصب میانی فشار وارد کرده و موجب سوزش، بی حسی و سرانجام دردی مختل کننده می شوند. کارکنان بسیاری از حرفه ها مثل نقشه کشی، منشی گری، نوازندگی، کار در خط موتاز، تعمیر اتومبیل و کار با کامپیوتر در معرض ابتلا به این بیماری قرار دارند. درمان آن با داروهای استروئید و ضدالتهاب، فیزیوتراپی و یا با عمل جراحی برای شل کردن رباط های عرضی میچ امکان پذیر است و اگر چه سلامتی همیشه حاصل نمی شود اما اغلب اوقات عملکرد میچ و دست بطور کامل بهبود پیدا می کند.

مانند بسیاری از بیماری های ماهیچه-استخوانی این بیماری نیز دلایل متفاوتی دارد و بیشتر اوقات در اثر ترکیب عوامل مختلفی بوجود می آید. مهمترین عامل حرکت یکنواخت و تکراری است. هنگامیکه دستها و انگشتان خود را خم می کنید وترهای تاکنده به دیواره های مجرای میچ دست سائیده می شوند. اگر دستهایتان را مدتی آزاد بگذارید این سائیدگی موجب تحریک وترها نخواهد شد. زمان استراحت ممکن است از کسری از ثانیه تا چند دقیقه تغییر کند و به عوامل مختلفی از جمله شدت و میزان خمیدگی میچ بستگی دارد. ضربه و فشاری که بر میچ دست یا ساعد وارد می شود نیز می تواند موجب تورم وترها و

عارضه سائیدگی مجرای میچ دست^۱ در میان استفاده کنندگان صفحه کلید کامپیوتر شایع است. این بیماری ممکن است به هر کسی صدمه برساند و عواقب آن جدی است. آگاهی و اطلاع از بیماری و دلایل ابتلا به آن نقش بسیار مهمی در پیشگیری آن دارد. تشخیص اولیه و معالجه به هنگام، مانع بروز ضعف و درد ناشی از بیماری خواهد شد. با رعایت احتیاط، تلاش در کاهش میزان اضطراب حین کار، ایجاد محیط کاری مناسب و شناسایی نشانه های هشدار دهنده بیماری، می توان از بروز آن در مراحل اولیه جلوگیری کرد.

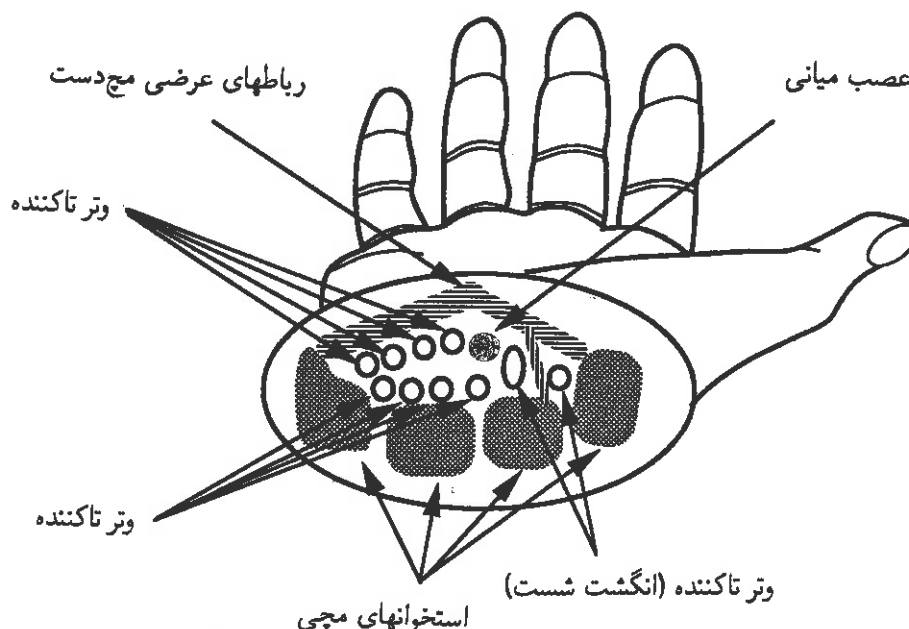
این بیماری، عصب میانی و وترهای تاکنده را در بر گرفته و از طریق مجرای میچ از جنس استخوانهای میچی و رباط های عرضی میچ دست از بازو تا انتهای دست امتداد پیدا می کند (شکل ۱).

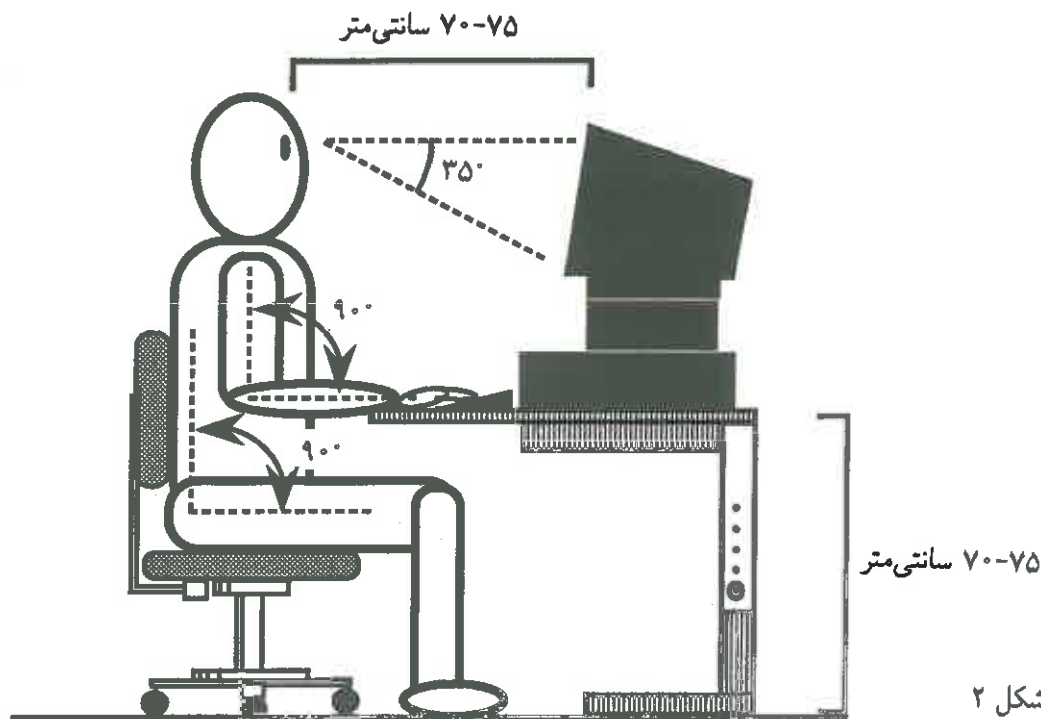
هنگامیکه دست و انگشتان خود را حرکت می دهید وترهای تاکنده به کناره های مجرا برخورد کرده و موجب تحریک وترها و تورم آنها

Mark Sheehan, "Avoiding Carpal Tunnel Syndrome: A Guide for Computer Keyboard Users", *Berkeley Computing Quarterly*, Spring 1991

Carpal Tunnel Syndrome^۱

شکل ۱





شکل ۲

بروز این بیماری شود.

از جمله عوامل دیگر، میزان انعطاف طبیعی وترهای تانکنده، شکل مقطع صلیبی میچ و شکل هندسی مجرای میچ دست که بسته به آمادگی ژنتیک اشخاص، صورتهای مختلف دارد، می تواند باعث بروز این عارضه شود. چون شکل هندسی برخی مجراها زمینه مساعدتری را برای تحریک وترها فراهم می کند. اضطرابهای شغلی نیز موجب افزایش ابتلا به این بیماری می شوند و به نظر می رسد که بیماری در میان الکلی ها رواج بیشتری دارد.

استفاده کنندگان کامپیوتر به چندین روش می توانند احتمال ابتلا به بیماری را کاهش دهند. بعضی از این روشها با شرایط محیط کار و برخی دیگر با عوامل انسانی در ارتباط است. وجود محل مناسبی برای نشستن نقش حساسی در محیط کاری ایفا می کند. ارتفاع صندلی و طرز قرار گرفتن تکیه گاه باید قابل تنظیم باشد. صندلی باید چرخ دار باشد تا بتوان به آسانی حرکت کرد. داشتن صندلی بازو دار اگر چه الزامی نیست اما می تواند مفید باشد. اکنون از روی شکل ۲ به بررسی دقیق این جزئیات می پردازیم.

به طور معمول ارتفاع میز باید بین ۷۰ تا ۷۵ سانتی متر باشد و برای افراد بلندقد می تواند کمی بلندتر از این میزان باشد. اگر ارتفاع میز قابل تنظیم است ابتدا زاویه کمر را بطور قائم قرار دهید آنگاه ارتفاع میز را طوری تنظیم کنید که وقتی دستهایتان روی صفحه کلید است زاویه آرنج ۹۰ درجه باشد. صفحه کلید باید در جایی باشد که میچ به راحتی بتواند روی میز قرار گیرد.

ارتفاع بعضی از صفحه کلیدها آنقدر زیاد است که موجب می شود دستها برای رسیدن به کلیدها با ناراحتی به سمت بالا خم شود که این رایج ترین سبب ابتلا به این بیماری در میان استفاده کنندگان کامپیوتر

است. در این صورت برای بالا نگه داشتن میچ می توان تکیه گاهی روی میز، مقابل صفحه کلید قرار دارد. با بالا و پائین بردن صندلی می توان زاویه آرنج و کمر را تنظیم کرد. آرنجها باید نزدیک بدن باشند تا جابجایی استخوان زند زیرین هنگامی که میچ به طرفین خم می شود (مثلاً وقتی کلید Z را می زنید) به حداقل برسد. پاها باید به راحتی روی سطح زمین قرار گیرد. اگر ارتفاع میز قابل تنظیم نیست می توان از یک زیرپایی استفاده کرد.

در خلال کار با صفحه کلید زمان بسیار کوتاهی برای استراحت وترهای تانکنده لازم است. اگر موقع کار میچ دست به بالا و طرفین خم می شود مدت این استراحت باید طولانی تر باشد. عارضه سائیدگی مجرای میچ اغلب در افرادی بروز پیدا میکند که نه تنها حرکات یکنواخت و تکراری دارند بلکه آنها را به مدت طولانی انجام میدهند. اگر با صفحه کلید کار می کنید بهتر است برنامه کاری خود را طوری تنظیم کنید که در هر ساعت ترکیبی از کارهای مختلف را انجام دهید. مثلاً به جای آنکه تمام صبح را به تایپ و بعد از ظهر را به بایگانی اختصاص دهید بهتر است در طول روز ترکیبی از دو کار را انجام دهید.

از علائم اولیه این بیماری، احساس سوزش در نوک انگشتان است که اغلب اوقات چندین ساعت پس از پایان فعالیت کاری آغاز می شود. دلیل زیاد بودن این فاصله زمانی مبتلایان میان درد و فعالیتهای کاری شان ارتباطی نمی یابند، تا زمانی که دیگر بسیار دیر شده است. سوزش انگشتان به سنگینی و بی حسی انگشتان و دست، و سرانجام به درد شدید میچ و دست منجر می شود. با ظهور اولین علائم بیماری، شخص باید توسط پزشک متخصص بهداشت شغلی مورد معاینه قرار گیرد. پزشک چند آزمایش ساده برای تشخیص بیماری انجام میدهد و مراحل مشخصی را برای اجتناب از آن تجویز می کند.

گزارش

دیسکهای نوری*

را داشتند اما حقیقت این است که آنها از همتهای مغناطیسی شان کنتند. دو دلیل برای این کندی نسبی وجود دارد. اول اینکه نوک ضبطکننده در یک دیسکگردان نوری خیلی بزرگتر و سنگینتر از مشابههای مغناطیسی است که روی ریزتراشه نصب می شوند. دلیل دوم طولانی بودن زمان صرف شده برای نوشتن روی دیسک نوری است. برای گرداندن^۴ دیسکهای ثابت نوشتن بسیار آسان است. زیرا ماده مغناطیسی سطح دیسک مستقیماً به تغییرات نوک ضبطکننده پاسخ میدهد و کلیه عملیات نوشتن فقط با یک بار عبور نوک ضبطکننده روی دیسک انجام می شود. در حالی که گرداندن دیسک نوری به سه مرحله نیازمند است: یک مرحله برای پاک کردن، بعدی برای نوشتن و سرانجام تأیید^۵ اطلاعات نوشته شده. مهندسين و طراحان مشغول طراحی و اصلاح سیستمی هستند که این عملیات را با یک بار عبور نوک ضبط روی دیسک انجام دهد و برای دست یافتن به این هدف ایده ای را از ثبت مغناطیسی اطلاعات به عاریت گرفته اند که به عنوان تکنولوژی راهگزینی میدانهای مغناطیسی^۵ شناخته می شود و طی آن پلاریته سیم پیچهای ضبط کننده با سرعتی بیش از چندین میلیون بار در ثانیه تغییر می کند.

نسل بعدی دیسکهای نوری نه تنها سریعتر خواهد بود بلکه قابلیت بیشتری نیز در ضبط اطلاعات خواهد داشت. دیسک نوری نصب شده روی کامپیوتر NeXT فعلاً توانایی ذخیره سازی ۲۵۶ مگابایت اطلاعات را داراست. این مقدار حافظه برای ذخیره سازی کارهای شکسپر، دایره المعارفها و مراجع دیگر کاملاً کفایت می کند.

دیسکهای نوری-مغناطیسی^۶ که از آخرین محصولات کانن هستند، قابلیت ذخیره سازی ۵۱۲ مگابایت را دارند. اساساً دو راه برای افزایش حجم اطلاعات ضبط شده روی دیسکهای نوری وجود دارد. اول کاهش طول موج یا به عبارتی باریکتر کردن شعاع اشعه لیزر تابنده که برای ایجاد حفره ها روی دیسک بکار می رود و دیگری دو برابر کردن حجم اطلاعات ذخیره شده یعنی در هر حفره به جای یک بیت، دو بیت اطلاع ذخیره می شود. تکنولوژی دیسکهای نوری-مغناطیسی بر پایه یافته های یک فیزیکدان اسکاتلندی به نام جان کر^۷ قرار دارد. او در سال ۱۸۷۶ کشف کرد که اگر پرتو نوری به سطحی مغناطیسی تابانده شود، نور بازتابی دارای چرخشی مختصر خواهد بود. جهت چرخش نور منعکس شده بستگی به پلاریته مغناطیسی بازتابانده دارد. این پدیده به نام اثر کر معروف شد.

در دیسکهای نوری-مغناطیسی از این خاصیت برای بازخوانی اطلاعات استفاده می شود. برای ذخیره سازی اطلاعات یک لیزر کوچک ولی

محدویت حافظه همیشه یکی از رایجترین ملاحظات بوده که طراحان کامپیوتر و برنامه نویسان می بایست مد نظر داشته باشند. ولی امروزه حجم اطلاعاتی که باید ذخیره و پردازش شوند آنچنان افزایش یافته که رسانه های ذخیره سازی رایج از قبیل دیسکهای ثابت یا کوچک نمی توانند این همه اطلاعات را در خود جای دهند. چاره کار در بکارگیری رسانه ای جدید با ظرفیتی زیاد، یعنی دیسکهای نوری، است که با اشعه لیزر کار می کنند.

دیسکهای رایج برای ذخیره سازی اطلاعات بر اساس تکنولوژی مغناطیسی عمل می کنند: سیگنالهای الکتریکی، شدت میدان مغناطیسی ایجاد شده توسط نوک ضبطکننده^۱ را تغییر می دهند. این میدان، ذرات اکسیده روی سطح دیسک را مغناطیس کرده و الگویی رقمی ایجاد می کند که داده هادر آن ضبط می شوند. در دیسکهای نوری، لیزر نوک ضبط کننده، سطح دیسک را گرم می کند. این سطح از مخلوط ویژه ای ساخته شده که تنها هنگامی که گرم است می تواند مغناطیس شود. وقتی مخلوط سرد می شود میدان مغناطیسی به طور کامل در آن شکل می گیرد. تاکنون سه نسل دیسک نوری تولید شده که هر سه، اطلاعات را به یک روش ذخیره می کنند، یعنی درون حفره هایی میکروسکوپی که در قطعه ای مارپیچی روی یک ماده شبیه آینه شکل گرفته اند. برای بازخوانی اطلاعات، نور حاصل از یک لیزر ریزتراشه^۲ به دیسک تابانده می شود. یک سنجنده، نور بازتابی را دریافت و استخراج کرده و عکس العمل حفره های ذکر شده را به بیت های اطلاعاتی تبدیل می کند.

در نسل اول رسانه های ذخیره سازی نوری، که بسیاری از راه دیسکهای فشرده صوتی با آن آشنا هستند فقط امکان خواندن اطلاعات وجود داشت. نسل دوم علاوه بر امکان فوق به کاربر اجازه می داد یکبار در ابتدا اطلاعات دلخواهش را روی دیسک بنویسد. این نسل از دیسکها برخی کاربردهای محدود خود را به عنوان یک وسیله نگهداری پرونده های حجیم بخصوص در دفاتر دولتی ثبت اختراعات پیدا کرد. اما نیاز به پیشرفت تکنولوژیک دیگری بود تا محدودیتهای خواندن و نوشتن را از میان بردارد. اینجا بود که دیسکهای نوری قابل خواندن و نوشتن به عنوان نسل سوم عرضه شدند و با ورود آنها رقابت بین رسانه های ذخیره سازی نوری و مغناطیسی جدی شد. دیسکهای نسل سوم در سال ۱۹۸۸ روانه بازار شدند و کارخانه کانن اولین شرکتی بود که اقدام به تولید انبوه دیسکهای نوری ۵/۲۵ اینچی کرد. این دیسکها در ابتدای امر و با توجه به قابل حمل بودن - در مقابل نقطه ضعف دیسکهای ثابت که قابل حمل نیستند - شروع بسیار خوبی

Optical Disks- Memories Are Made of This, Chronicle, *
July-August 1990

head ۱
microchip ۲

driver ۳
verify ۴
magnetic field switching ۵
magneto-optical ۶
John Kerr ۷

می‌سازد، محدود کردن خواسته‌ها از نرم‌افزار را در عرضه انواع مختلف فونت، نادرست خواند. علاوه بر این وی همکاری بین حروف‌چین‌ها و طراحان نرم‌افزار و استفاده از تجربیات دست‌اندرکاران نشر را در طراحی نرم‌افزارهای بهتر ضروری دانست. سخنران همچنین ضمن مهم توصیف کردن تنوع، کثرت و زیبایی قلم‌های مورد استفاده به امکانات عرضه شده توسط نرم‌افزارها برای سطر بندی، ستون بندی، جدول بندی و درج پاوری اشاره کرد.

عنوان سخنرانی بعدی "صفحه‌آرایی به کمک کامپیوتر در سیستم‌های جامع نشر" بود که توسط آقای مهدی یوسفی از شرکت تدارک کامپیوتر ایران ایراد شد. به گفته ایشان سیستم‌های صفحه‌آرایی دستی، زمانبر و پرهزینه بوده و نتیجه کار آنها همراه با اشتباهات و اشکالات فراوان است در حالیکه سیستم‌های کامپیوتری سرعت و دقت بالاتر، انعطاف پذیری و زیبایی را به همراه دارند. آنگاه روشهای وارد کردن متن برای حروفچینی بررسی شد و به تواناییهای نرم‌افزارهای نشر در عرضه امکانات ویرایشی از قبیل غلط‌گیری، اضافه کردن متن یا حذف از آن، امکانات گرافیکی مثل گنجاندن عکس و طرح اشاره رفت.

"صفحه‌آرایی به کمک کامپیوتر" عنوان سخنرانی بعدی بود که توسط آقای هومن مهر از شرکت سیستان ارائه شد. آقای مهر مسائل مطرح در صفحه‌آرایی را تحت عناوین زیر بررسی کرد: تنوع نیازها که ناشی از وجود انواع مختلف کتابها از قبیل کتابهای شعر، درسی، علوم، روزنامه‌ها، مجلات و... است؛ ستون بندی مناسب مخصوصاً هنگامی که متن حاوی شکل و فرمول باشد، چیدن درست شکلها، مشکل مربوط به صفحات روبرو، گنجاندن جلوه‌های ویژه در صفحه‌آرایی مطبوعات و تبلیغاتی، تفکیک رنگها و مشکل و زمانگیر بودن اصلاحات و تغییرات. آنگاه نقش کامپیوتر در سیستم نشر بررسی شد و به تواناییهایی که کامپیوتر در ساده‌تر کردن کار و عرضه امکانات جدید دارد اشاره رفت که از بین آنها می‌توان امکان اعمال الگوهای استاندارد موسسه‌های انتشاراتی، سرعت عمل و امکان بایگانی و استفاده مجدد از متون را نام برد. آنگاه در توصیف سیستم‌های موجود به گرایش رایج در گرفتن مرکزیت از کامپیوتر و سهل‌الاستفاده‌تر کردن کامپیوترها برای ناشران اشاره رفت. در پایان نیز امکانات و لوازم مورد نیاز سیستم‌های نشر کامپیوتری مورد بررسی قرار گرفت.

سخنرانی بعد توسط ناطقی از شرکت میکرو نرم‌افزار، تحت عنوان "محیط کار متن و گرافیک" ایراد شد. سخنران به تبدیل و تغییر جهت گیری از انسان ماشین‌گرا به ماشین انسان‌گرا و ساده‌تر شدن استفاده از برنامه‌ها اشاره کرد و عناصر اصلی محیط‌های تصویری را تحت عنوان پنجره، نماد، موش و منو نام برد. سپس راحت بودن کار با سیستم‌های تصویری، نیاز به آموزشی کوتاه‌مدت، زیبایی، عرضه امکانات بیشتر، تطبیق تصویر با متن چاپ شده، کمی اشتباه و عدم نیاز به کدهای کنترلی به عنوان محاسن سیستم‌های تصویری بر شمرده شدند. آنگاه به محدودیت‌های محیط تصویری نیز اشاره رفت که از بین آنها نیاز

قوی برای ایجاد حفره‌های میکروسکپی به قطر کمتر از یک هزارم میلی‌متر، قطعه‌های موجود روی سطح دیسک را گرم می‌کند. گرما ماده سطح دیسک را آماده قبول میدان مغناطیسی که توسط یک سیم پیچ تولید می‌شود، می‌کند. تغییرات پلاریته سیم‌پیچ قابلیت ایجاد الگوهای رقمی به شکل صفر و یک را ایجاد می‌کند. برای بازخوانی داده‌های ضبط شده از همان لیزر اما با قدرتی کمتر استفاده می‌شود. سنجنده‌های نوری از روی پرتو بازتابی و توسط اثر کر مشخص می‌کنند که چه نوع داده‌ای - صفر یا یک - در هر حفره نوشته شده است. برای پاک کردن اطلاعات تنها کافیست محلهای مورد نظر مجدداً گرم شوند و سپس اطلاعات جدید قابل نوشتن خواهند بود.

در سالهای اخیر دستگاههای جانبی کامپیوترها بسیار پیشرفت کرده‌اند و بعید نیست که در آینده‌ای نه چندان دور شاهد ساخت حافظه‌های بسیار حجیم با ظرفیتی در حدود چندین گیگابایت باشیم.

ترجمه: مجتبی نیرودی

سمینار نشر کامپیوتری

اولین سمینار نشر کامپیوتری همزمان با نمایشگاه بین‌المللی کتاب تهران در ۱۸ اردیبهشت ماه از سوی شورای عالی انفورماتیک و با هدف معرفی امکانات جدید نشر با کامپیوتر برای مسئولان و دست‌اندرکاران عمده چاپ و نشر کشور در سالن کنفرانس نمایشگاه برگزار شد. بعد از مراسم افتتاح که آقای صباح زنگنه از مسئولان وزارت ارشاد و آقای داوری معاون انفورماتیک سازمان برنامه نیز در آن شرکت داشتند، برنامه سخنرانها آغاز شد.

ابتدا آقای شربانی از دبیرخانه شورای انفورماتیک تحت عنوان "تاریخچه استفاده از کامپیوتر در انتشارات" به ایراد سخنرانی پرداخت. ایشان ابتدا به توان محاسباتی و قدرت و سرعت کامپیوترها و کاربردی که این توانایی در صنعت چاپ می‌تواند داشته باشد اشاره کردند. آنگاه نقش خط در سیستم انتقال اندیشه مورد بحث قرار گرفت و تاریخچه‌ای از خط فارسی بیان شد. صنعت چاپ بعنوان وسیله دیگری برای انتقال اندیشه معرفی شد و ضمن ارائه تاریخچه چاپ مکانیکی در ایران، به زمانگیر بودن روشهای قدیمی چاپ اشاره شد. سخنران سال ۱۹۵۰ را بعنوان تاریخ شروع استفاده از کامپیوتر در چاپ عنوان کرد و گران بودن نرم‌افزار و سخت‌افزار سیستمهای اولیه را از موانع کاربرد گسترده آنها به حساب آورد. بعد از اشاره به تعدادی از نرم‌افزارهای ویراستار فارسی آقای شربانی سخنرانی خود را با توصیه‌های برای انتخاب و طراحی سیستم‌های نشر پایان داد.

سخنرانی بعدی تحت عنوان "امکانات ویرایشی و صفحه‌آرایی در نرم‌افزارهای نشر به کمک کامپیوتر" توسط آقای محمد صنعتی از شرکت نرم‌افزاری سینا ارائه شد. آقای صنعتی ضمن تأکید بر توانایی بالای نرم‌افزارها در عرضه امکاناتی که نیازهای ناشران را کاملاً بر آورده

معرفی کتاب

مقدمه‌ای بر سیستم و ساختار فایلها
(ذخیره و بازیابی اطلاعات: جلد اول)

سید محمد تقی روحانی (انکوهی)

کانون انتشارات علمی

۴۲۷ صفحه، ۳۱۱۱ ریال

نویسنده، عضو هیأت علمی دانشکده مهندسی برق و کامپیوتر دانشگاه شهید بهشتی تهران است و کتاب به عنوان کتاب درسی برای دانشجویان دوره کارشناسی در رشته مهندسی و علوم کامپیوتر و رشته‌های همسایه تهیه شده است. مأخذهای اصلی این کتاب عبارتند از:

- Data Base Design, Wiederhold, 1985.
- File Organization for Data Base Design, Wiederhold, 1987.

کتاب، شامل هفت فصل است که در آنها، رسانه‌های ذخیره‌سازی و ساختارهای مختلف پرونده‌ها مورد بررسی قرار گرفته‌اند.

مرجع کامل و راهنمای طراحی با AutoCAD

عباس روشن

جهاد دانشگاهی دانشگاه صنعتی امیرکبیر

۳۲۹ صفحه، ۱۵۱۱ ریال

این کتاب، ترجمه آقای مهندس عباس روشن، عضو هیأت علمی دانشکده مهندسی عمران دانشگاه صنعتی امیرکبیر است. منبع اصلی، کتاب

- AutoCAD (The Complete Reference), Nelson Johnson

است. این کتاب، شامل چهار قسمت است که در آنها، طراحی و پردازش نقشه‌ها، گسترش و تکمیل طرحها، روشهای پیشرفته و مرجع دستورات و توابع گنجانده شده‌اند.

مبانی بیسیک برای کامپیوترهای کوچک

بهرام اشتری

انتشارات بهینه

دو جلد، ۳۶۲+۳۵۲ صفحه، هر جلد ۱۴۱۱ ریال

کتاب بر اساس زبان بیسیک کامپیوترهای خانگی اسپکتروم نوشته شده است و می‌توان از آن به عنوان اولین کتاب برای آموختن برنامه‌نویسی به زبان بیسیک استفاده کرد.

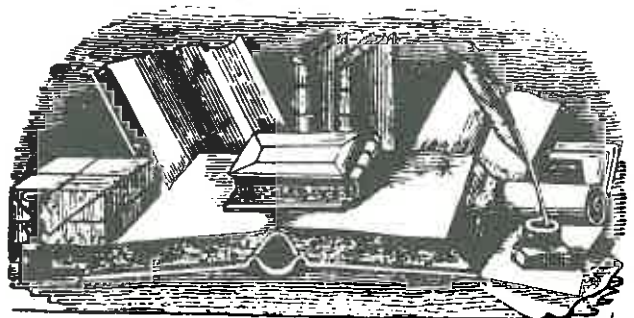
به پردازش زیاد، حافظه مصرفی فراوان و گرانی چنین سیستمهایی را می‌توان نام برد. در ادامه سخنرانی محاسن و معایب محیط‌های متن نیز مورد بررسی قرار گرفتند.

نیاز نداشتن به سیستم‌های نمایش رنگی و سخت‌افزار و نرم‌افزار گرانقیمت در بین محاسن و دوره‌های آموزش طولانی، ساده نبودن کار در محیط متن، امکانات محدود و تکیه بر روشهای سعی و خطا در آماده‌سازی متن جزو معایب محیط متن محسوب شد.

آقای احمد یزدی‌پور از شرکت داده‌کای سخنرانی خود را با عنوان "طراحی فونت" ایراد کرد. ایشان ضمن ارائه تعریفی از فونت، ویژگیهای فونت خوب را خسته‌کننده نبودن و مناسب بودن فاصله‌های بین حروف برشمرد. سپس روشهای دستی تهیه فونت مورد بحث قرار گرفت و اشراف بر خوشنویسی و طراحی، شناخت ابزارهای چاپ و داشتن نگاه تیزبین از ویژگیهای طراح فونت محسوب شد. آنگاه به توانایی کامپیوتر در طراحی فونت اشاره رفت و دو روش استفاده از پویشگر و دیگری طراحی ریاضی و پارامتریک بررسی شد و ویژگیهای هر کدام توضیح داده شد.

سخنرانی بعدی زیر عنوان "محیط کار" توسط آقای حسین بادبانچی از شرکت داده‌کای انجام شد. سخنران محیط‌های اجرای نرم‌افزارهای حروف‌چینی و صفحه‌بندی را به دو دسته منطقی و بصری تقسیم کرد. آنگاه مراحل کار در هر دو روشی طراحی بصری و منطقی بررسی شد و به امکاناتی که هر کدام از این دو روش برای اعمال تغییرات عرضه می‌کنند و نیز به تواناییها و نقاط ضعف هر کدام اشاره شد و زمینه‌های کاربردی مناسب هر یک از دو روش نیز بر شمرده شد. آقای بادبانچی همچنین تحت عنوان "حروف‌چینی" به بررسی نکاتی که حروف‌چینی با کیفیت بالا را از حروف‌چینی ظریف متمایز می‌سازد، پرداخته و موضوعات سطر بندی و حروف‌چینی را اجمالاً بررسی کرده و معیارهایی را برای ارزیابی کیفی انواع حروف‌چینی - اعم از کامپیوتری و غیر کامپیوتری - ارائه داد.

بعد از پایان آخرین سخنرانی تحت عنوان "Postscript" سخنرانان به سئوالات حاضران پاسخ گفته و سپس شرکت کنندگان در سمینار برای بازدید از امکانات عملی نشر کامپیوتری شرکتها، به غرفه مربوط هدایت شدند. مجموعه مقالات عرضه شده در سمینار، قبلاً به صورت ویژه‌نامه نشریه "خبرنامه انفورماتیک" آماده شده و در روز برگزاری سمینار در معرض فروش قرار داشت.



واژه‌نامه

تغییرات و اضافات واژه‌نامه مقدماتی

کمیتهٔ بازبینی واژه‌نامه در نشست‌های متعدد خود، بازبینی واژه‌هایی را که با حرف D شروع می‌شوند به پایان رسانده است. در زیر، نتیجهٔ این بازبینی و تغییرات و اضافاتی که توسط کمیته صورت گرفته است، جهت اطلاع علاقه‌مندان درج می‌گردد.

اعضای گرامی انجمن و دیگر خوانندگان ماهنامه می‌توانند برای ابراز نظرات خود و یا دانستن علل تغییرات اعمال شده و واژه‌های پیشنهادی، کتباً با کمیتهٔ بازبینی واژه‌نامه به نشانی انجمن تماس حاصل نمایند. کار بازبینی واژه‌نامهٔ مقدماتی ادامه دارد و نتایج حاصله، حرف به حرف به اطلاع اعضای گرامی انجمن و خوانندگان ماهنامه خواهد رسید. کمیتهٔ بازبینی واژه‌نامه امیدوار است که از نظرات سازندهٔ صاحب‌نظران و علاقه‌مندان برخوردار شود. قبل از ارائهٔ فهرست واژه‌ها، یادآوری نکات زیر را لازم می‌دانیم:

- ۱- دریافتن معادل فارسی برای واژه‌ها، غالباً جنبهٔ «اسمی» واژه‌ها در نظر گرفته شده است.
- ۲- واژه‌های غیرکامپیوتری واژه‌نامهٔ مقدماتی حذف گردیده‌اند.
- ۳- معادل بعضی از واژه‌ها کلاً تغییر یافته است و برای بعضی دیگر معادل یا معادل‌های جدیدی به معادل‌های قبلی اضافه شده و یا تقدم و تأخر معادل‌های پیشنهادی تغییر یافته است.
- ۴- بدیهی است که به‌مرور زمان ممکن است تغییرات و اضافات دیگری نیز پیشنهاد شوند که نتایج مربوطه به تدریج اعلام خواهند شد.

D

Daisy wheel printer	چاپگر چرخ فلکی	Data dictionary	قاموس داده‌ها، فرهنگ داده‌ها	Data model	الگوی داده‌ای، مدل داده‌ای
Dangling reference	ارجاع معلّق	Data element	عنصر داده	Data object	موضوع داده
Dash	خط تیره	Data encapsulation	محصورسازی داده‌ها	Data phone	تلفن - داده
Data	داده، داده‌ها	Data encryption	رمزبندی داده‌ها	Data processing	داده‌پردازی
Data abstraction	تجرید داده‌ها	Data entry	ورود داده‌ها	Data record	رکورد داده
Data asquisition	کسب داده‌ها	Data field	حوزهٔ داده‌ها	Data security	امنیّت داده‌ها
Data aggregate	جمع‌آوری داده‌ها	Data flow	جریان داده	Data service unit	واحد خدمات داده‌ای
Data bank	بانک داده‌ها	Data flow language	زبان جریان داده‌ای	Data set	دستهٔ داده
Database	پایگاه داده‌ها، پاد	Data format	قالب داده	Data station	جایگاه داده
Database machine	ماشین پایگاه داده	Data gathering	گردآوری داده‌ها	Data structure	ساختار داده‌ها، ساختار داده
Database management system	سیستم مدیریت پایگاه داده	Data gram	داده‌نگار	Data sub-language	زیر زبان داده‌ای
Data carrier	حامل داده‌ها، داده‌بر	Data handling	داده‌گردانی	Data switching equipment	تجهیزات راه‌گزینی داده‌ها
Data cell	سلول داده	Data hiding	پنهان سازی داده‌ها	Data system integration	مجموع‌سازی سیستم داده‌ها
Data center	مرکز داده‌ها	Data hierarchy	سلسله مراتب داده‌ها	Data terminal equipment	تجهیزات پایانی داده‌ها
Data chain	زنجیر داده‌ها	Data independence	استقلال داده‌ها، ناپستگی داده‌ها	Data track	شیار داده، ترک داده
Data channel	مجرای داده‌ها	Data item	قلم داده، فقرهٔ داده	Data transfer	انتقال داده‌ها
Data circuit-terminating equipment	تجهیزات ختم مدار داده‌ها	Data level	سطح داده	Data transmission	ارسال داده‌ها
Data communications equipment	تجهیزات ارتباط داده‌ها	Data library	کتابخانهٔ داده‌ها، داده‌خانه	Data type	داده‌گونه، نوع داده
Data concentrator	متمرکزکنندهٔ داده‌ها	Data link	پیوند داده‌ها	Data unit	واحد داده
Data definition language	زبان تعریف داده‌ها	Data link layer	لایهٔ پیوند داده‌ها	Data validation	اعتبار سنجی داده‌ها
Data dependence graph	نمودار وابستگی داده‌ها	Data management	مدیریت داده‌ها	Data word	کلمهٔ داده
		Data manipulation	آمایش داده‌ها، دستکاری داده‌ها	Datum	داده
		Datamation	داده‌پردازی		
		Data medium	محیط داده‌ها		

Deadend condition	وضعیت بن بست	Demand-driven execution	اجرا بر حسب نیاز	Dialogue system	سیستم محاوره‌ای
Deadlock	بن بست	Demand-paging	صفحه بندی بر حسب نیاز	Dial pulse	تکانه شمارگیری
Deadly embrace	بن بست	Demand processing	پردازش بر حسب نیاز	Dial-up line	خط شمارگیری
Dead space	فضای بی مصرف، فضای مرده	Demodulation	تفکیک، پیاده کردن	Dibit	دوبیت
Deallocation	بازستاندن، واپس گیری	Demultiplexor	اشباعگر	Dicap storage	حافظه دوخازنی
Debatable time	زمان بی صاحب	Dense binary code	رمز دودویی چگال	Dichotomizing search	جستجوی دو شقی
Deblocking	بلوک شکنی، از حالت بلوکی در آوردن	Dense indexing	نمایه بندی چگال	Die sorting	مرتب سازی تاسی
Debugger	اشکال زدا	Density packing	فشرده سازی چگال	Difference engine	موتور تفاضلی
Debugging	اشکال زدایی	Dependency graph	نمودار وابستگی	Differential	تحلیل گر تفاضلی
Debugging tools	ابزارهای اشکال زدایی	Depth	عمق، ژرفا	Digit	رقم
Decay	میرایی	Depth-first search algorithm	الگوریتم جستجوی ژرفایی	Digital	رقمی، دیجیتال
Decentralization	عدم تمرکز، تمرکز زدایی	Dequeue	صف دو طرفه، دو صف	Digital-to-analog converter	مبدل رقمی به تمثیلی
Decimal code	رمز ددهمی	Derivation	اشتقاق	Digitization	رقمی سازی
Deciphering	رمزگشایی	Derivation tree	درخت اشتقاق	Digitizer	رقمی ساز
Decision box	جعبه تصمیم گیری، جعبه تقسیم	Descending order	ترتیب نزولی	Direct	مستقیم
Decision making	تصمیم گیری	Descriptive language	زبان توصیفی	Direct access	دستیابی مستقیم
Decision mechanism	سازوکار تصمیم گیری	Descriptive programming	برنامه سازی توصیفی	Direct address	نشانی مستقیم
Decision support system	سیستم پشتیبانی تصمیم گیری	Design	طرح	Direct-coupled	ترویج شده مستقیم
Decision table	جدول تصمیم گیری	Design automation	خودکار سازی طراحی	Direct memory access	دستیابی مستقیم به حافظه
Deck	دسته	Design review	مرور طراحی	Directive	رهنمود
Declaration	اعلان، اظهار	Desk calculator	ماشین حساب رومیزی	Directory	فهرست راهنما
Declarative language	زبان اعلاتی	Deskew	راست کردن	Disable signal	علامت تعلیق، علامت ناممکنی
Decoder	رمزگشا	Desk-top computer	کامپیوتر رومیزی	Disc	دیسک
Decoding	رمزگشایی	Desk-top-publishing	نشر رومیزی	Discard	دورانداختن
Decollate	صفحه صفحه کردن	Destination address	نشانی مقصد	Disconnect signal	علامت قطع
Decomposition	تجزیه	Destruction	تخریب	Disk	دیسکت
Decreasing order	ترتیب نزولی	Destructive read operation	عمل خواندن مخرب	Disk controller	کنترل کننده دیسک
Decrementation	کاهش، نزول	Detachment	جدایی	Disk drive	گرداننده دیسک، دیسک گردان
Decryption	رمزگشایی، رمزشکنی	Detail file	پرونده جزئیات	Disk file	پرونده روی دیسک
Dedicated	اختصاصی	Detailed design	طراحی تفصیلی	Disk Operating System (DOS)	سیستم عامل دیسکی
Dedication	اختصاصی دهی	Detection	آشکار سازی	Disk pack	بسته دیسک
Deductive database	پایگاه داده‌های قیاسی	Determine	سبب شدن	Disk platter	سینی دیسک
Deductive inference	استنباط قیاسی	Determiner	مستب	Disk storage	حافظه دیسک
Default	پیش فرض	Determinant key	کلید سبب ساز	Disk unit	واحد دیسک
Default condition	وضعیت پیش فرض	Deterministic	قطعی	Diskette	دیسکت
Default logic	منطق فرضهای ناگفته	Deterministic search	جستجوی قطعی	Dismount	پیاده کردن
Default value	مقدار پیش فرض	Development	توسعه	Dispatching	تقسیم
Defective track	شیار خراب، ترک خراب	Device	دستگاه	Displacement	جابجایی
Defensive programming	برنامه سازی مداخله‌ها	Device-end condition	وضعیت ختم کار دستگاه	Display	نمایش
Deferred	معوق	Diagnosis	خطایابی	Display console	پیشانه نمایش
Definability	تعریف پذیری	Diagnostic check	ارزیابی خطایابانه	Display controller	کنترل کننده صفحه نمایش
Definition	تعریف	Diagnostic message	پیغام خطایابی	Display memory	حافظه نمایش
Deflection yoke	صفحه منحرف کننده	Diagnostic program	برنامه خطایاب	Display menu	فهرست نمایش
Degradation	کاهش درجه، تنزل درجه	Diagnostic test	آزمون خطایابی	Display screen	صفحه نمایش
Delay	تاخیر	Diagnostics	امکانات خطایابی	Display terminal	پایانه نمایش
Deletion	حذف	Diagonal	قطری	Distributed database	پایگاه داده توزیع شده
Delimitation	فاصله اندازی	Diagonal method	روش قطری	Distributed processing	پردازش توزیع شده
Delimiter	فاصل	Diagram	نمودار	Divide-and-conquer method	روش تقسیم و تقو
		Dial	شمارگیری	Document	مدرک، سند

Documentation	مستندسازی، مدرک نگاری	Drawing	ترسیم	Dumping	تخلیه
Docuterm	کلید واژه اسناد	Drift error	خطای خرنده	Duodecimal system	دستگاه شمار دوازده گانی
Don't care	بی اهمیت	Drop-dead halt	ایست مختومه	Duplex	دو سویه
Dot matrix display	صفحه نمایش با ماتریس نقاط	Drop-down menu	فهرست نمایش طوماری	Duplicate key	کلید مکرر
Dot matrix printer	چاپگر ماتریسی نقطه‌ای	Drum	طبله	Duplication	تکرار دوگانه
Dot plots	تصاویر نقطه‌ای	Drum printer	چاپگر طبله‌ای	Duration	دوام
Double precision arithmetic	حساب با دقت مضاعف	Dry run	اجرای آزمایشی	Duty cycle	چرخه انجام وظیفه
Double punch	منگنه مضاعف	Dual	دوگانه	Dyadic operation	عمل دوتایی
Double-length	با طول مضاعف	Dual coding	برنامه نویسی دوگانه	Dynamic allocation	تخصیص پویا
Double-rail logic	منطق دو خطی	Dumb terminal	پایانه ناهوشمند	Dynamic binding	متین سازی پویا
Double-word	کلمه مضاعف، دو کلمه	Dummy	ساختگی، ظاهری	Dynamic error	خطای پویا
Doubly-linked list	لیست با پیوند دو طرفه	Dummy argument	نشاندن ساختگی، نشاندن ظاهری	Dynamic memory	حافظه پویا
Down-count	پایین شمار	Dump	تخلیه	Dynamic programming	برنامه سازی پویا
Down time	زمان بیکاری	Dump check	ارزیابی تخلیه	Dynamic relocation	جابجایی پویا
		Dump tape	نوار تخلیه		

اعضا

اعضاء جدید

ع	۱۳۲۳	رحمت اله	رئیس اردکانی	ع	۶۳۱۵	بیبا	سینی
ع	۱۴۸۹	علی اصغر	مزینانی	ع	۶۳۱۶	علیرضا	شهرابی فراهانی
ع	۱۵۰۲	کاظم	ورعی	ع	۶۳۱۷	نعمت اله	سلیم پور
ع	۱۵۵۵	مونسعلی	فرمنش	ع	۶۳۱۸	ناصر	عسکری
ع	۱۶۵۳	ژوزف	بهنامی	ع	۱۱۴۱	فرید	اثنی عشری
ع	۱۷۶۲	احمد	پاشا مقدم	ع	۱۱۴۲	مهرداد	امین فر
ع	۱۷۹۴	علی	حسینیان	ع	۱۱۴۵	مینا	اجلی
ع	۱۸۰۱	سیدابراهیم	ابطحی	ع	۱۱۴۶	سیدمرتضی	احمدی
از د به ع	۱۹۴۳	نادر	اردشیری	ع	۱۱۴۷	محمود	محسنی نیا
از د به ع	۱۹۴۴	پرویز	نیک آذر	ع	۱۱۴۸	مری	شاخو
				ع	۱۱۴۹	محمد رضا	باطنی
				و	۳۲۸۳	علیرضا	همتی
				و	۳۲۸۴	مجتبی	نیک آذر
				و	۳۲۸۵	محمد علی	صفایی
				و	۳۲۸۶	امیر	سر رشته داری
					۸۰۴۵	روابط عمومی	شرکت ایران خودرو ح
تمدید عضویت							
	۵۷۱۳	مریم	برادران	د	۵۹۴۹	حمید رضا	ملاحسنی مجدآبادی
	۶۰۰۵	انور	مهاجری ناو	د	۶۱۳۶	سیده مریم	عطار سیدی
	۶۳۱۶	محمد علی	نیک زاده	د	۶۳۱۷	ناصر	عسکری
	۶۳۱۸	فرید	اثنی عشری	ع	۱۱۴۱	فرید	اثنی عشری
	۱۱۴۲	مهرداد	امین فر	ع	۱۱۴۵	مینا	اجلی
	۱۱۴۶	سیدمرتضی	احمدی	ع	۱۱۴۷	محمود	محسنی نیا
	۱۱۴۸	مری	شاخو	ع	۱۱۴۹	محمد رضا	باطنی
	۳۲۸۳	علیرضا	همتی	و	۳۲۸۴	مجتبی	نیک آذر
	۳۲۸۵	محمد علی	صفایی	و	۳۲۸۶	امیر	سر رشته داری
	۸۰۴۵	روابط عمومی	شرکت ایران خودرو ح				
	۶۲۹۷	شهریار	طاوسی تفرشی	د	۶۲۹۸	سیامک	رضائی
	۶۲۹۹	محسن	ظروفچی بنیسی	د	۶۳۰۰	رحیم	صوفی زاده
	۶۳۰۱	حجت اله	مدیریان	د	۶۳۰۲	فرانک	بهنام
	۶۳۰۳	نسرین	غمخوار	د	۶۳۰۴	رضا	پارسا
	۶۳۰۵	ابوالفضل	ثاقب یزدنژاد	د	۶۳۰۶	شهریار	پزشگی
	۶۳۰۷	سپیده	فرازم	د	۶۳۰۸	نادر	جامی مقدم
	۶۳۰۹	لیلا	شریفی اردجان	د	۶۳۱۰	بهرداد	غیاث الدین
	۶۳۱۱	مسعود	جمشیدی	د	۶۳۱۲	مهران	فلقی
	۶۳۱۳	امید	خطیبی	د	۶۳۱۴	محمد علی	نیک زاده

نشریات ارسالی برای افراد زیر برگشت داده میشود، بدین وسیله از ایشان درخواست می گردد تا آدرس اصلاح شده خود را به اطلاع انجمن برسانند.

۱۸۳۹ علی رضا جواهری
۳۲۵۷ هادی چیردار
۶۱۱۹ ناصر دادرس (آلمان)
۶۱۴۰ مجید کریمزاده
۳۲۵۵ مجتبی ساجدی
۳۲۴۷ محمد رضا ریاحی

بدین ترتیب آمار اعضای انجمن تا پایان اردیبهشت ۱۳۷۰ به قرار زیر است:

عادی	دانشجویی	وابسته	برجسته	حقوقی	کل
۲۴۰	۳۸۳	۴۱	۱	۲۵	۶۹۰

ترجمه یک کتاب

نفر-ماه افسانه‌ای (۷)

نوشته: فردریک بروکز

ترجمه: علی پارسا

۷- چرا برج بابل به جایی نرسید؟

بررسی پروژه برج بابل از دید مدیریت

بر اساس آنچه که در "سفر پیدایش" انجیل آمده است، برج بابل بعد از کشتی نوح دومین کار بزرگ مهندسی انسان بود. در عین حال این برج اولین اقتضاح مهندسی به شمار می‌رود.

داستان برج بابل از چند نظر عمیق و آموزنده است. اما بیائیم و آن را صرفاً به عنوان یک پروژه مهندسی در نظر بگیریم و ببینیم چه درسهای مدیریتی می‌توان از آن آموخت. آیا این پروژه از اسباب و

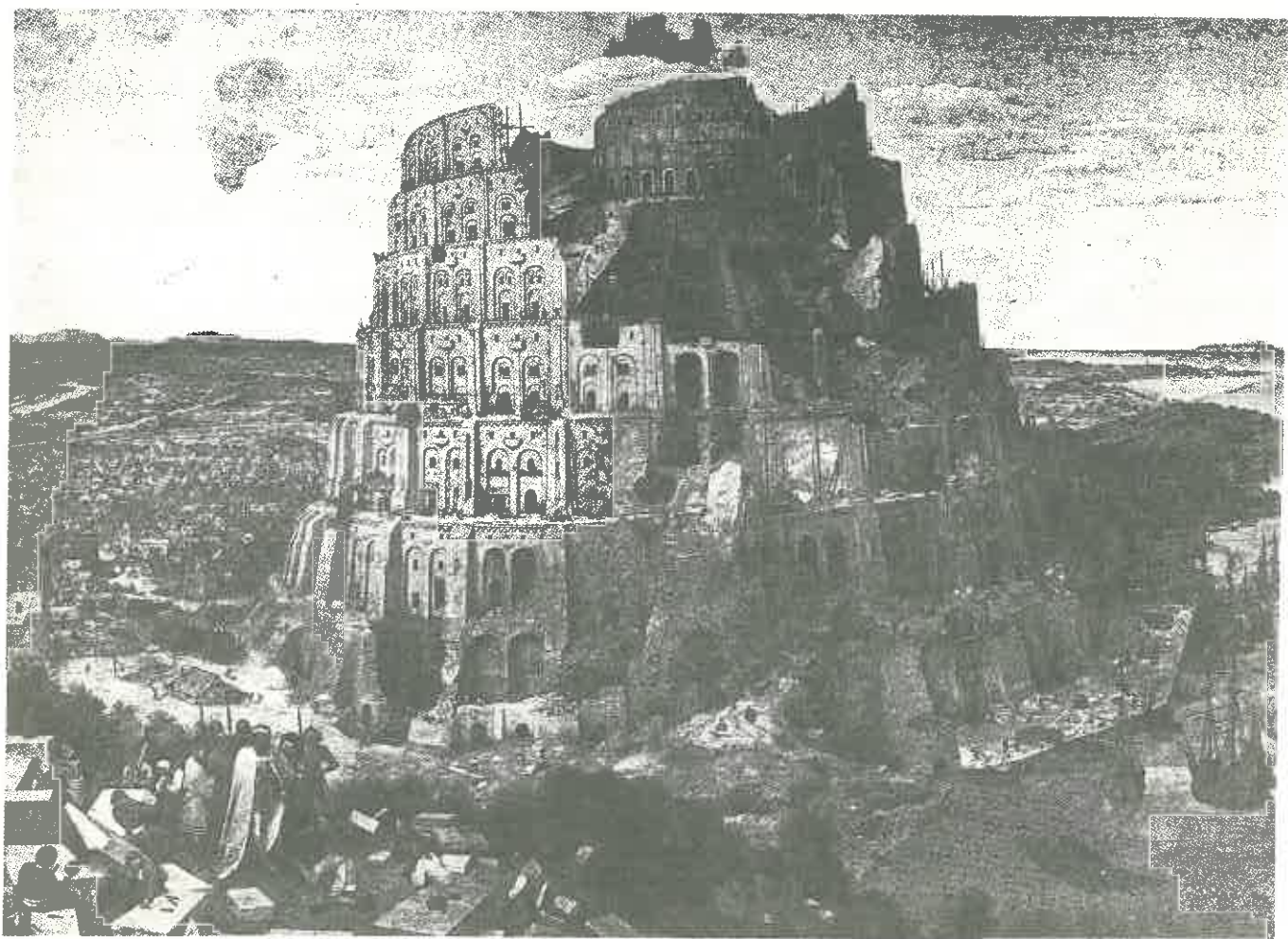
لوازم موفقیت برخوردار بود؟ آیا:

۱- وظیفه آن معلوم بود؟ بله، هر چند که ساده‌دلانه و غیر ممکن بود. البته این پروژه مدتها قبل از رسیدن به محدودیت‌های اساسی خود شکست خورد.

۲- نیروی انسانی داشت؟ خیلی هم زیاد.

۳- مواد و مصالح موجود بود؟ خاک رس و ماسه در بین‌النهرین به وفور موجود است.

۴- زمان کافی در دسترس بود؟ بله، نشانه‌ای از محدودیت زمانی برای آن در دست نیست.



جزوه پروژه

چيست؟ جزوه پروژه خودش جزوه مستقلی نیست بلکه ساختاری است که بر تمام مدارکی که بهر حال در پروژه تولید می شود، حاکم است.

تمام مستندات پروژه باید جزئی از این ساختار باشند. این امر شامل شرح اهداف، مشخصات بیرونی، مشخصات فاصلها، استانداردهای فنی، مشخصات درونی و یادداشتهای اداری می شود.

برای چيست؟ نوشته های فنی تقریباً جاودانی اند. اگر کسی شجره نامه یک جزوه راهنمای مشتری را که برای محصول سخت افزاری یا نرم افزاری نوشته شده بررسی کند نه تنها می تواند اندیشه های اولیه طراحی بلکه در مواردی عین جملات و بندهای اولیه مدارک پیشنهاد تولید محصول و طراحی را بازشناسد. برای نویسندگان فنی قدرت جعبه چسب از قدرت قلم کمتر نیست.

به همین دلیل، و به دلیل آنکه جزوه های محصولات فردا از یادداشتهای امروز ناشی خواهد شد، تنظیم ساختار مدارک و مستندات پروژه اهمیت بسیار زیادی دارد. تنظیم جزوه پروژه در مراحل اولیه باعث می شود که ساختار مستندات پروژه، حساب شده و منظم، و نه بی ترتیب و درهم باشد. علاوه بر این، ایجاد ساختاری برای مستندات، نوشته های بعدی را به صورتی دسته بندی شده در می آورد که با ساختار تعیین شده هماهنگ باشد.

دلیل دوم برای لزوم ایجاد جزوه پروژه همانا نظارت بر جریان توزیع اطلاعات است. البته مسئله، جلوگیری از اطلاع رسانی نیست بلکه خاطر جمع شدن از این است که اطلاعات لازم به تمام کسانی که به آن نیاز دارند برسد.

قدم اول این است که تمام یادداشتهای را شماره گذاری کنیم که هر کارمند پروژه به فهرست منظم شده ای از عناوین مدارک دسترسی داشته باشد تا بتواند آنچه را که می خواهد بیابد. البته سازمان دهی جزوه پروژه باید از این هم فراتر رود و ساختاری درختی برای یادداشتهای فراهم آورد. با ساختار درختی می توان در صورت نیاز فهرست هائی از زیر-درختها ایجاد کرد.

نحوه کار. وخیم تر شدن مسئله یادداشتهای فنی هم، مثل سایر مسائل مدیریت پروژه های برنامه سازی، با افزایش حجم کار نسبتی غیر خطی دارد. اگر ده نفر در پروژه داشته باشیم می توان مدارک را فقط شماره گذاری کرد. اگر صد نفر داشته باشیم چند دنباله خطی شماره گذاری شده، کفایت می کند. وقتی هزار نفر باشند، که طبعاً از لحاظ جغرافیائی در چند محل مختلف مستقرند، نیاز به جزوه ای ساخت یافته و نیز حجم این جزوه افزایش می یابد. در این صورت چگونه باید روند کار را هدایت کرد؟

به نظر من این کار در پروژه OS/360 به خوبی انجام گرفت. او.اس. لاکن که قبلاً در پروژه سیستم عامل ۷۰۱۰-۱۴۱۰ فایده جزوه های ساخت یافته پروژه را دیده بود، مصرانه بر اهمیت آن تأکید کرد.

۵- از تکنولوژی مناسب بهره می گرفت؟ بله، شکل هرمی یا مخروطی، ذاتاً پایایی دارد و نیروهای فشاری را به خوبی پخش می کند. فن بنائی در آن روزگار فنی شناخته شده بود و پروژه قبل از رسیدن به محدودیت های فنی شکست خورد.

خوب، اگر همه این عوامل جمع بود پس چرا این پروژه موفق نشد؟ چه عاملی کم بود؟ پاسخ آنست که دو عامل: اول ارتباطات و در نتیجه آن سازمان دهی. افراد پروژه نمی توانستند با یکدیگر صحبت کنند و به همین دلیل نمی توانستند کارها را هماهنگ کنند. وقتی که هماهنگی از بین رفت، کار هم خوابید. می توان تصور کرد که نبود ارتباط منجر به اختلاف و نارضایتی و چشم و همچشمی گروهی شد. دیری نگذشت که هر گروه راه خود را در پیش گرفت و قطع ارتباط را بر مشاجره برگزید.

ارتباطات در پروژه های بزرگ برنامه سازی

امروز هم همین اوضاع هست. فاجعه کم آوردن زمان، ناهماهنگی عملیاتی و اشکال های سیستمی، همه در اثر نبودن ارتباط صحیح بین بخشهای مختلف پروژه پیش می آید. در جریان پیشرفت کار، هر گروه به تدریج عملیات و اندازه و سرعت برنامه های خود را تغییر می دهد و فرضیات خود را در مورد ورودی های موجود و خروجی ها، به طور صریح یا ضمنی، عوض می کند.

مثلاً سازندگان یک روال جایگزاشت برنامه ها به مشکلی بر می خورند و سرعت برنامه خود را کم می کنند و در این امر به آماری تکیه دارند که می گوید این روال در برنامه های کاربردی کمتر بکار می رود. در همین زمان همکاران آنان در کار طراحی بخش مهمی از برنامه مبصر هستند که بسیار به سرعت کار این روال وابستگی دارد. چنین تغییری در سرعت، خود تغییری اساسی در مشخصات محصول است و باید به اطلاع عموم برسد و از نظر ملاحظات سیستمی سبک و سنگین شود.

بسیار خوب، قبول داریم که گروه ها باید با یکدیگر ارتباط داشته باشند، ولی چگونه و از چه راههائی؟ پاسخ اینست که از راههای متعدد:

- ارتباط غیر رسمی. وجود ارتباط مناسب تلفنی و تعیین مشخص نحوه ارتباط گیری بین گروهی باعث برقراری صداها تماس تلفنی می شود که برای رسیدن به تعبیر مشترکی از مستندات و مطالب نوشته شده مفید است.

- جلسات. جلسات منظم پروژه که در آن هر گروه گزارش فنی مختصری ارائه می کند، بسیار ارزشمند است. با این روش صداها سوتفاهم جزئی دود می شود و به هوا می رود.

- جزوه پروژه. یک جزوه رسمی پروژه را باید از همان آغاز ایجاد کرد. توضیح این کار خود بخش مستقلی را می طلبد.

دلار صرفه جویی کردیم. توانستیم زمان آماده شدن میکروفیش‌های مربوط به تغییرات را بسیار کم کنیم و حجم جزوه پروژه با این کار از یک سوم متر مکعب به شصت سانتی متر مکعب تقلیل یافت. از همه مهمتر اینکه تغییرات صفحات به صورت صدتا صدتا اعمال می‌شد و کار جاگذاری صفحات را صدبار کمتر کرد.

البته میکروفیش هم مشکلاتی دارد. از نظر مدیران کار طاقت فرسای جاگذاری ورقه‌های تغییر یافته این حسن را داشت که افراد تغییرات را می‌خواندند و هدف تهیه جزوه پروژه هم همین بود. ولی با میکروفیش کار اعمال تغییرات بیش از حد لازم ساده می‌شد و برای جبران این امر باید ورقه‌هایی همراه با آنها توزیع می‌شد و در آن تغییرات انجام شده بر شمرده می‌شد.

علاوه بر این میکروفیش را نمی‌شود علامت زد یا با مداد روی آن نوشت. جزوه‌هایی که خواننده بتواند روی آنها نظرش را بنویسد، هم برای نویسنده مؤثر است و هم برای خواننده.

در مجموع به نظر من استفاده از میکروفیش کار بجائی بود و من آن را در پروژه‌های بسیار بزرگ بر کار کاغذی ترجیح می‌دهم.

امروز چگونه این کار را انجام می‌دهیم؟ با تکنولوژی امروزی به نظر من می‌توان جزوه پروژه را همراه با تغییرات و تاریخ تجدید نظرها روی پرونده‌های دستیابی مستقیم نگهداشت. هر کاربر می‌تواند با یک پایانه نمایشی به جزوه دسترسی داشته باشد (استفاده از پایانه‌های تاپی کند خواهد بود). فهرستی از تغییرات را، که به طور روزانه تهیه می‌شود، می‌توان به ترتیب عکس ورود در جای ثابتی نگهداشت. هر برنامه‌نویس روزانه به این فهرست مراجعه می‌کند و اگر روزی این کار را نکرد فردا باید فهرست طولانی‌تری را بخواند. وقتی که خلاصه تغییرات را می‌خواند می‌تواند به سراغ خود متن برود و آنرا نیز بررسی کند.

توجه داشته باشید که در این روش خود جزوه پروژه تغییر نمی‌کند. این جزوه همچنان مجموعه‌ای از تمام مدارک پروژه است که بر اساس طرحی حساب شده سازمان داده شده است. تنها روش توزیع و مراجعه به آن تغییر کرده است. دی.سی. انگلبارت و همکارانش در مؤسسه تحقیقات استنفورد چنین سیستمی را ساخته‌اند و از آن در نگهداری مستندات شبکه "آرپا" بهره می‌گیرند.

دی.ال. پاراناس از دانشگاه کارنگی-ملون راه حلی انقلابی‌تر را ارائه داده است. نظر او این است که برنامه‌نویس وقتی بارآور است که بجای دسترسی داشتن به جزئیات بخشهای سیستم که جزو مسئولیت او نیست، از دسترسی به آنها منع شود. پیش فرض این امر تعریف جامع و دقیق تمام فاصل‌های بین بخشهای سیستم است. اگر چه این امر نشانه برخورداری از طراحی خوب و مناسب است ولی تکیه بر تأمین تمام و کمال آن به فاجعه منجر خواهد شد. هر سیستم اطلاعات خوب باید هم خطاهای فاصل‌ها را نشان دهد و هم اصلاح آنها را تشویق و ترغیب کند.

از همان ابتدا تصمیم گرفتیم که هر برنامه‌نویسی باید تمام مدارک و مستندات را ببیند، یعنی باید یک نسخه از جزوه پروژه را در محل کارش داشته باشد.

نو کردن جزوه نیز اهمیتی حیاتی دارد. جزوه پروژه باید وضع فعلی پروژه را منعکس کند. این امر در صورتی که لازم باشد مدارکی را برای تغییرات دوباره تایپ کرد، بسیار مشکل خواهد بود. ولی اگر جزوه پروژه به صورت اوراق سوراخ شده کلاسور تهیه شود، آنگاه می‌توان فقط صفحاتی را که تغییر می‌کنند عوض کرد. ما یک سیستم متن‌پرداز کامپیوتری در اختیار داشتیم که برای کار نگهداری به موقع مدارک بسیار مفید بود. نسخه‌های اصلی آماده عکس برداری را با کامپیوتر تهیه می‌کردیم و صفحات تغییر یافته ظرف کمتر از یک روز توزیع می‌شد. البته کسی که این گونه صفحات تغییر یافته را دریافت می‌کند در جذب مطالب آنها مشکل خواهد داشت. مثلاً اولین باری که یک چنین صفحه‌ای را می‌گیرد می‌خواهد بداند "چه چیزی را تغییر داده‌اند؟" و بعدها که به آن صفحه مراجعه می‌کند می‌خواهد بداند "امروز تعریف این مطلب چیست؟"

این خواسته آخری را می‌توان با تولید مدارکی که دائماً نو می‌شوند بر آورد. ولی البته مشخص کردن تغییرات انجام شده مستلزم کارهای دیگری است. اول اینکه باید متن تغییر یافته را در صفحه علامت بزنند، مثلاً با افزودن یک خط سیاه عمودی در حاشیه هر یک از خط‌های تغییر یافته متن. دوم اینکه باید با هر صفحه تغییر یافته یک صفحه دیگر همراه کنند و در آن خلاصه‌ای از تغییرات انجام شده و اهمیت آنها را بیاورند.

جزوه پروژه خودش جزوه مستقلی نیست بلکه ساختاری است که بر تمام مدارکی که بهر حال در پروژه تولید می‌شود، حاکم است.

هنوز شش ماه از آغاز پروژه ما نگذشته بود که به مسئله دیگری برخوردیم. مسئله این بود که ضخامت جزوه پروژه به بیش از یک مترونیم رسیده بود! اگر می‌خواستیم صد نسخه از این جزوه را که برنامه‌نویسان ما از آن استفاده می‌کردند روی هم بچینیم از ارتفاع ساختمان تایم-لایف در مانهاتان که در آن کار می‌کردیم هم بالاتر می‌زد. علاوه بر این میزان تغییرات روزانه این جزوه هم به طور متوسط پنج سانتیمتر بود، یعنی در حدود ۱۵۰ صفحه که باید لابه‌لای تمام جزوه می‌گذشتیم. به این ترتیب نگهداری جزوه پروژه کم کم بخش زیادی از یک روز کاری را می‌گرفت.

در این موقع بود که ما به میکروفیش رو آوردیم و با این کار با وجود گذاشتن یک دستگاه میکروفیش خوان در هر یک از دفترهای کارمان یک میلیون

سازمان‌دهی پروژه‌های بزرگ برنامه‌سازی

پروژه" است. بیشتر ارتباط‌گیری‌های او در درون گروه انجام می‌شود و کارش تقریباً تماماً فنی است.

اکنون روشن می‌شود که استعداد لازم برای این دو نقش کاملاً متفاوت است. استعداد آدم‌ها ترکیبات مختلفی دارد و بسته به نحوه ترکیب استعدادهاشان، تهیه‌کننده و کارگردان فنی روابط متفاوتی با یکدیگر خواهند داشت. هر تشکیلاتی را باید بر اساس آدم‌هایی که در دسترس قرار دارند سازمان داد نه اینکه افراد را در سازمان‌هایی فرضی جاگذاری کرد.

متفکرین کم‌اند، اهل عمل کم‌تراند و متفکرین اهل عمل کم‌ترینند.

ممکن است تهیه‌کننده و کارگردان فنی یک نفر باشد. این کار را می‌توان در گروه‌های بسیار کوچک سه نفری تا شش نفری انجام داد. به‌دو دلیل در پروژه‌های بزرگتر این کار عملی نیست. اول اینکه پیدا کردن فردی که هم از لحاظ استعداد مدیریت و هم از لحاظ فنی قوی باشد مشکل است. متفکرین کم‌اند، اهل عمل کم‌تراند و متفکرین اهل عمل کم‌ترینند.

دلیل دوم آنکه در پروژه‌های بزرگ هر نقشی یک کار تمام وقت یا حتی بیشتر از آن است. مشکل است که تهیه‌کننده بتواند آن قدر از وظایف خود را به دیگران محول کند تا وقتی برای کار فنی بیاورد. و اگر کارگردان بخشی از وظایف خود را به دیگران محول کند به یکپارچگی مفهومی کار لطمه می‌خورد.

می‌شود تهیه‌کننده، رئیس باشد و کارگردان، وزیر دست‌راست او. مشکل این امر در حفظ قدرت تصمیم‌گیری فنی کارگردان در عین محول کردن کارهای مدیریتی به او است.

مسلم است که تهیه‌کننده باید اقتدار فنی کارگردان را رسماً اعلام کند و در اغلب آزمون‌هایی که برای این اقتدار پیش می‌آید باید از او صد درصد حمایت کند. برای امکان‌پذیر شدن این امر تهیه‌کننده و کارگردان باید تلقی یکسانی از روش فنی برخورد به مسائل داشته باشند. آنها باید مشکلات فنی را در خلوت و قبل از آنکه دیر شود مطرح کنند و تهیه‌کننده باید احترام زیادی برای توانایی‌های فنی کارگردان قائل باشد.

نکته‌ای که چندان بدیهی نیست این است که تهیه‌کننده باید با استفاده ظریف از انواع نشانه‌های برتری (مثل اندازه اتاق کار، فرش این اتاق، کیفیت میز و صندلی، وسائل اداری و غیره) نشان دهد که کارگردان با وجود بیرون بودن از سلسله مراتب مدیریتی، دارای قدرت تصمیم‌گیری است.

این کار را می‌توان به‌خوبی اجرا کرد و متأسفانه کمتر به آن توجه شده است. کاری که از این هم کمتر با موفقیت اجرا شده است استفاده مناسب مدیر پروژه از نابغه‌های فنی است که چندان قدرتی در امور مدیریت ندارند.

اگر در پروژه‌ای n نفر باشند، آنگاه $(n^2 - n)/2$ مجرای ارتباطی وجود خواهد داشت و تقریباً 2^n گروه را باید با هم هماهنگ کرد. هدف سازمان‌دهی کم کردن میزان ارتباط‌گیری و هماهنگی مورد نیاز است. بنابراین کار سازمان‌دهی، حمله‌ای بنیادی به مسائل ارتباطی بالا است. ابزارهای برطرف کردن مسائل ارتباطی عبارتند از تقسیم کار و تخصصی کردن فعالیت‌ها. ساختار درختی پروژه‌ها نشانه کم شدن نیاز به ارتباط‌گیری‌های جزئی در صورت تقسیم و تخصصی کردن کارها است.

در عمل یک سازمان درختی از ساختار اختیارات و مسئولیت‌ها ناشی می‌شود. این اصل که هر کس باید فقط یک رئیس داشته باشد ناچار منجر به ساختار درختی می‌شود. اما ساختار ارتباطی اینقدر محدود نیست و ساختار درختی اصلاً ربطی به ساختار واقعی ارتباطی پروژه، که در واقع یک شبکه است، ندارد. کاستی‌های سازمان درختی منجر به ایجاد گروه‌های کارکنان، گروه‌های ویژه کمیته‌ها و حتی سازمان‌دهی ماتریسی شده‌است که در بسیاری از مؤسسات مهندسی یافت می‌شود. ببینیم یک سازمان درختی برنامه‌سازی چگونه است و خصوصیات الزامی هر زیر درخت آن باید چه باشد تا به‌کارآئی لازم دست یابد. این خصوصیات عبارتند از:

۱- یک ماموریت

۲- یک تهیه‌کننده

۳- یک کارگردان فنی یا معمار

۴- یک برنامه زمانی

۵- یک تقسیم کار

۶- فاصل‌های تعریف شده بین بخشها

تمام این خصوصیات بدیهی و رایج‌اند مگر تفاوتی که بین تولیدکننده و کارگردان فنی قائل شده‌ایم. ابتدا تفاوت این دو نقش را بررسی می‌کنیم و سپس رابطه آنها را بیان خواهیم کرد.

نقش تهیه‌کننده چیست؟ او گروه را گرد هم می‌آورد، کارها را تقسیم می‌کند، و برنامه زمانی را تعیین می‌کند. در نتیجه بخش مهمی از کار او برقراری ارتباط با خارج گروه است، یعنی هم ارتباط با بالاتر و هم ارتباط با سایر گروه‌ها. او نحوه ارتباط‌گیری و گزارش‌دهی درون گروه را نیز تعیین می‌کند. و بالاخره او مراقب اجرای برنامه زمانی است و با جابجا کردن منابع و تغییر دادن سازمان‌دهی گروهی نسبت به اوضاع تغییر یابنده عکس‌العمل نشان می‌دهد.

نقش کارگردان فنی چیست؟ او طرحی را که باید اجرا شود می‌ریزد، اجزای آن را مشخص می‌کند، شکل بیرونی آن را تعیین می‌کند، و ساختار درونی آن را اجمالاً معین می‌سازد. او وحدت و یکپارچگی مفهومی کل طرح را تأمین می‌کند و از این لحاظ عاملی است که میزان پیچیدگی محصول را محدود می‌کند. هر مسئله فنی که پیش می‌آید، او با ارائه یک راه حل یا اگر لازم باشد با تغییر دادن طرح، با آن مقابله می‌کند. به قول آل کپ او "مسئول درگیر شدن با کارهای مشکل درون

این به بعد او برده تو است. تو مهندس ارشد هستی و رئیس بلامنازع محسوب می شوی. جاک هم عالیجناب رئیس کل سایر امور است. دیگر نگران هیچ چیز نباش - مگر نگران جزئیات امر ساختن سفینه ماه نورد."

کاستر و برکلی دست دادند. برکلی بالحنی جدی گفت: "آقای کاستر فقط یک تقاضا دارم. مهم نیست که در هر موردی بدون اطلاع من تصمیمی بگیرید، بالاخره شما باید این کار را راه بیاندازید - ولی به خاطر خدا هر تصمیمی را که می گیرید روی نوار ضبط کنید تا من هم بدانم اینجا چه خبر است. می دهم یک کلید روی میز شما نصب کنند که یک ضبط صوت را روی میز من به کار می اندازد."

کاستر گفت: "بسیار خوب!" و هریمین با خود فکر کرد که کاستر جوان تر شده است.

برکلی گفت: "و اگر چیزی می خواهید که جنبه فنی ندارد، خودتان اینکار را نکنید، کافی است یک دکمه را فشار دهید و به من خبر دهید - حتماً انجام خواهد شد." بعد نگاهی به هریمین کرد و گفت: "رئیس می گوید که می خواهید با شما در مورد کار اصلی صحبت کند، من دیگر شما را تنها می گذارم که به کارتان برسید." بعد رفت بیرون.

هریمین نشست، کاستر هم بعد از او نشست و گفت: "پوف!"

هریمین گفت: "حالت بهتر شد؟"

"آره، از قیافه این برکلی خوشم آمده."

"خوبه، از این به بعد شما برادر دوقلوی هم هستید. هیچ نگران نباش. من قبلاً هم با او کار کرده ام. وقتی با او کار می کنی مثل این است که در یک بیمارستان بسیار مرتب بسر میبری."

این داستان محتاج تحلیل و توضیح نیست. همین کار را می توان در پروژه ها به نحو مؤثری اجرا کرد.

به نظر من همانطور که در فصل سوم تحت عنوان "گروه جراحی" شرح داده ام این راه سوم برای گروه های کوچک مناسب تر است. و فکر می کنم که تعیین تهیه کننده به عنوان رئیس در زیر درخت های بزرگ پروژه های بزرگ بهتر است.

شاید برج بابل اولین اقتضای مهندسی باشد ولی مسلماً آخرین نیست. ارتباط گیری و نتیجه منطقی آن یعنی سازمان دهی در موفقیت یا عدم موفقیت یک پروژه نقش اساسی دارند. روشهای ارتباط گیری و سازمان دهی به اندازه خود تکنولوژی نرم افزاری نیازمند تفکر و تجربه و قابلیت مدیر پروژه اند.

ممکن است کارگردان، رئیس باشد و تهیه کننده، وزیر دست راست. رابرت هاین لاین در کتابش به نام هردی که ماه ۱۰ فروخت این تمهید را به خوبی نشان داده است:

کاستر صورتش را با دست هایش پوشاند و بعد سرش را بالا کرد. "میدانم. میدانم که چه کار باید کرد - ولی هر وقت می خواهم بنشینم و روی یک مسئله فنی کار کنم یک ابله احمق می آید و می خواهد که در مورد کامیون ها یا تلفن ها یا یک زهرمار دیگر تصمیمی بگیرم. متأسفم آقای هریمین. من خیال می کردم که می توانم کار را انجام دهم." هریمین به آرامی گفت: "باب، خودت را ناراحت نکن. فکر کنم چند وقت است که خوب نخوابیده ای، نه؟ حالا بگویم که چه کار باید کرد. یک کلک میزنیم به فرگوسن. من چند روز میز تو را اشغال می کنم و ترتیبی می دهم که دیگر با اینجور چیزها درگیر نشوی. دلم می خواهد که مغز تو به بردارهای عکس العملی و کارآئی سوخت و تنش های طراحی فکر کند نه به قرارداد کامیون ها." هریمین رفت دم در، نگاهی به بیرون اتاق انداخت و مردی را که شاید کارمند دفتری بود، و شاید هم نبود، دید و گفت: "آهای، بیا اینجا!" آن مرد با حالتی شگفت زده بلند شد و آمد دم در و گفت: "بله!"

"می خواهم همین الان آن میزی که در آن گوشه هست با تمام وسایل روی آن به یک دفتر خالی در این ساختمان منتقل شود." او بر کار انتقال کاستر و میز دیگرش به دفتر دیگری در ساختمان نظارت کرد و گفت تلفن دفتر جدید را قطع کنند و بعد از کمی تأمل، داد یک کاناپه را هم به آن دفتر منتقل کردند. بعد به کاستر گفت: "یک پرورکتور و یک ماشین نقشه کشی و کتابخانه و این جور آشغال ها را همین امشب می آوریم. یک فهرست از هر چیزی که برای کار کردن روی مسائل مهندسی لازم داری تهیه کن." بعد به دفتر قبلی مهندس ارشد رفت و با روحیه و سرزندگی شروع به کار کرد تا ببیند وضع تشکیلات چگونه است و اشکالش در چیست.

چهار ساعت بعد او برکلی را برای ملاقات به دفتر کاستر برد. مهندس ارشد روی میز کار سرش را روی دستهای گذاشته بود و خوابیده بود. هریمین خواست برگردد و از اتاق خارج شود که کاستر بیدار شد. با خجالت گفت: "بیخسید، کمی چرت زدم."

هریمین گفت: "آن کاناپه را برای همین در اتاق گذاشته ام. از میز راحت تر است. حالا بیا با جاک برکلی آشنا شو. از



در انجمن انفورماتیک ایران عضو شوید

به هدف‌ها و فعالیت‌های انجمن.

کلیه اعضای انجمن از خدمات و امتیازات زیر برخوردار می‌شوند:

- دریافت رایگان گزارش کامپیوتر و استفاده از تخفیف در بهای سایر نشریات انجمن.
- حق شرکت در سخنرانی‌ها، سمینارها و دوره‌های مختلف علمی و فنی به صورت رایگان و یا با تخفیف قابل ملاحظه نسبت به افراد غیر عضو.
- دریافت اخبار و اطلاعات مربوط به انفورماتیک، امکان آشنایی و تبادل افکار با سایر کارشناسان این رشته و قرار گرفتن در جریان رویدادهای ملی و بین‌المللی در انفورماتیک.
- حق استفاده رایگان از کتابخانه و مرکز اسناد انجمن.
- حق عضویت سالانه انجمن (از مهرماه تا شهریورماه سال بعد) به شرح زیر است:
- عضو عادی، برجسته یا وابسته * ۲۵۰۰ ریال
- عضو دانشجویی * ۱۰۰۰ ریال
- عضو حقوقی * ۵۰۰۰۰ حداقل ریال

حق عضویت شامل اشتراک رایگان ماهنامه گزارش کامپیوتر (از زمان عضویت تا شهریورماه آینده) نیز هست. اعضا و مشترکین خارج از کشور باید برای هزینه پست مبلغ ۲۵۰۰ ریال علاوه بر مبلغ فوق پرداخت نمایند.

شرایط و مزایای عضویت

تقاضای عضویت در انجمن انفورماتیک ایران باید با معرفی دو نفر از اعضای برجسته یا عادی انجمن، مدارک لازم و حق عضویت سالانه همراه باشد. شرایط انواع عضویت در انجمن به شرح زیر است:

- ۱- عضو عادی: داشتن درجه لیسانس یا بالاتر در رشته انفورماتیک و یا درجه لیسانس همراه با حداقل سه سال سابقه کار مفید در این رشته.
- ۲- عضو برجسته: داشتن حداقل پنج سال سابقه عضویت عادی در انجمن، سوابق درخشان در کمک به پیشبرد هدفهای انجمن و فعالیتهای برجسته (پژوهشی، علمی، اجرایی یا آموزشی) در رشته انفورماتیک.
- ۳- عضو افتخاری: سابقه فعالیت‌های سازنده و کمک‌های عمده در پیشبرد هدف‌های انجمن (فقط با دعوت).
- ۴- عضو وابسته: فعالیت در رشته انفورماتیک و یا رشته‌های وابسته. عضو وابسته می‌تواند طبق ضوابطی، که در آئین‌نامه‌های انجمن پیش‌بینی شده، به عضو عادی تبدیل شود.
- ۵- عضو دانشجویی: اشتغال به تحصیل در یکی از دانشگاه‌ها یا مراکز معتبر آموزش عالی داخل یا خارج کشور با گواهی موسسه مربوطه.
- ۶- عضو حقوقی: موسسه یا سازمان علاقه‌مند

انجمن انفورماتیک ایران در سال ۱۳۵۷ به عنوان یک انجمن مستقل و غیرانتفاعی تأسیس شد.

هدف‌های انجمن انفورماتیک ایران مطابق اساسنامه عبارتند از:

الف) ترویج کاربرد انفورماتیک از طریق وضع استانداردهای علمی، فنی و معنوی، بررسی اثر تکنولوژی کامپیوتر بر وضع اجتماع و محیط زیست، آشنا کردن افراد اجتماع با محاسن و خطرات استفاده از کامپیوتر، کمک در وضع قوانین حمایتی و حفاظتی مربوط به انفورماتیک و کاربردهای آن.

ب) بالا بردن سطح دانش در انفورماتیک از طریق کمک به بهبود کیفیت آموزش، توسعه و تشویق فعالیتهای پژوهشی، ایجاد همکاری بین مراکز علمی، صنعتی و اجرایی، و حفظ ارتباط بین متخصصین این رشته به منظور همکاری و تبادل افکار.

اساسنامه انجمن برای نیل به این هدف‌ها چاپ نشریات، تشکیل مجامع علمی، طرح‌ریزی و بررسی برنامه‌های آموزشی، ارزیابی تخصص‌ها، تشویق فعالیتهای پژوهشی، توسعه مبادلات علمی، فنی و نیروی انسانی، کمک در وضع استانداردها، و بالاخره ایجاد یک کتابخانه را پیش‌بینی کرده است.

با قبول عضویت و شرکت در فعالیتهای انجمن می‌توانید به پیشرفت و توسعه و در نهایت مفید بودن انجمن برای متخصصین رشته انفورماتیک و جامعه کشورمان کمک کنید.

تقاضای عضویت



لطفاً در این قسمت چیزی ننویسید .

نوع عضویت	شماره عضویت
-----------	-------------

اگر در سایر انجمنهای علمی (داخلی و خارجی) عضو هستید نام ببرید:

نشانی کامل پستی:

شهر:
نشانی:
کد پستی:

تلفن:

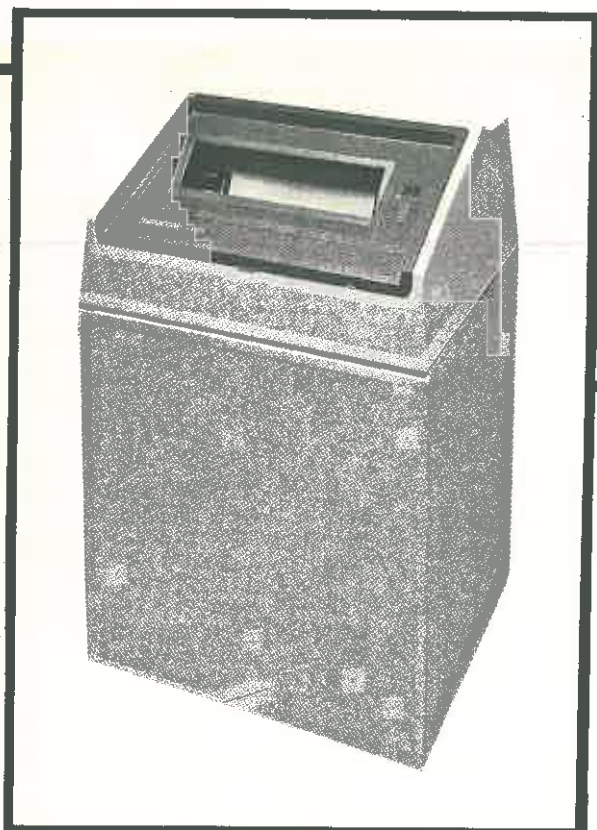
نام و خانوادگی: نام: خانوادگی: تاریخ تولد: شماره شناسنامه: جنسیت: زن ☐ مرد ☐ آیا قبلاً عضو انجمن انفورماتیک ایران بوده‌اید؟ بله ☐ خیر ☐ تاریخ تکمیل تقاضای عضویت: امضاء متقاضی: مشخصات دوتنفر عضوادی یا برجسته انجمن که صحت مطالب مندرج در این برگه را تأیید می‌کنند: نام معرف اول: نام معرف دوم: شماره عضویت: شماره معرف: امضاء معرف: امضاء معرف:

حق عضویت باید همراه این تقاضا نامه باشد. لطفاً برای تعدیل عضویت از این تقاضا نامه استفاده نکنید.

در صورت نداشتن معرف ارائه معرفی نامه دانشجویی از دانشگاه مربوطه برای سال تحصیلی جاری (برای دانشجویان) و یا فتوکپی آخرین مدرک تحصیلی برای سایرین کافی است.

برای عضویت، اشتراک ماهانه یا کسبه اطلاع بیشتر با نشانی «انجمن انفورماتیک ایران صندوق پستی ۱۱۹۶-۱۳۱۵۵ تهران تماس بگیرید.

و نحوه مورد نظر را به حساب جاری ۱۱۱۷ بانک صادرات شعبه ۱۶۵۲ واریز نموده، فیش بانکی را برای انجمن بفرستید.



۱- کمپانی CALCOMP بزرگترین سازنده تجهیزات گرافیکی در دنیا.

- PEN PLOTTER
- ELECTROSTATIC PLOTTER
- THERMAL TRANSFER PRINTER
- DIGITIZER

شرکت نگار عرضه کننده انحصاری محصولات:



با قابلیت اتصال به PC و IBM MAINFRAME MINI

۲- کمپانی CONTEX سازنده SCANNER سایز بزرگ برای سیستم CAD و بایگانی اسناد فنی.
● با قدرت تفکیک ۴۰۰ تا ۶۰۰ نقطه در اینچ در اندازه‌های A1 و A0 و قدرت تشخیص تا ۱۲۸ سطح خاکستری.

۳- کمپانی GENICOM از معتبرترین سازندگان چاپگرهای ماتریسی و خطی در دنیا.
● چاپگرهای خطی با تکنولوژی SHUTTLE و BAND با سرعت ۲۰۰۰ خط در دقیقه.

● چاپگرهای ماتریسی HEAVY DUTY با سرعت تا ۶۰۰ کارکتر در ثانیه.
● با قابلیت اتصال به انواع کامپیوترهای IBM MAINFRAME و MINI, PC

شرکت نگار

تهران - خیابان استاد مطهری - خیابان جم (فجر) - خیابان تابنده
(شهید لطفی) - کوچه بهرامی - پلاک ۷

تلفن: ۸۳۸۶۲۸ - ۴۴۰۸۴۵۰ تلکس: ۲۱۲۴۴۱

فاکس: ۸۳۵۹۹۷ صندوق پستی: ۱۵۸۷۵/۱۴۱۴

