

گزارش کامپیوتر

زبانهای برنامه‌سازی



- راه‌های تبیده
- گزارش سمینار آموزش و انفورماتیک
- کامپیوترهای چندرسانه‌ای

دی ۲۷۰



تصویر روی جلد: برج بابل، اثر پیتر بروگل

گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران

سردبیر: علی پارسا

گزارش کامپیوتر فعلاً هر دو ماه یکبار منتشر می‌شود.

مسئولان انجمن

رئیس: ابراهیم تقیبزاده مشایخ (۲۴۳۸۶۰)

نایب رئیس: داریوش جوان تبریزی (۸۶۳۳۶۷)

خزانه دار: میرعلی زیدی منفرد (۸۱۷۲۵۱۱)

دبیر: علی پارسا (۶۱۴۴۲۵)

سرپرستان کمیته‌ها

انتشارات: سعید وحید (۸۱۸۴۵۹)

روابط عمومی: داریوش جوان تبریزی (۸۶۳۳۶۷)

فعالیت‌های علمی و فنی: محمد حسن محوری

آموزش: ابراهیم تقیبزاده مشایخ (۲۴۳۸۶۰)

عضویت: میرعلی زیدی منفرد (۸۱۷۲۵۱۱)

دیگراعضای هیئت اجرایی

مسئول هماهنگی کمیته فعالیت‌های علمی و فنی: دکتر مهدی بهشتیان (۲۲۳۸۳۶)

مسئول هماهنگی کمیته آموزش: حسین کمالی (۲۸۰۹۵۸)

بازرس انجمن: 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. شکل‌ها باید با روان‌نویس مشکی ترسیم و به صورت آماده چاپ ارسال شوند.

انجمن انفورماتیک ایران انجمنی است علمی و مستقل که به منظور تبادل اطلاعات علمی و فنی و ارتقا سطح دانش انفورماتیک در ایران تأسیس 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 **TEX** یا **پارسی** حروف چینی و صفحه‌بندی شده است. در این شماره از همکاری شرکت داده کاوی ایران برخوردار بوده‌ایم که بدینوسیله از ایشان تشکر می‌کنیم. آقایان حسین بادبانچی و احمد یزدی‌پور در انجام مراحل مختلف صفحه‌بندی و حروفچینی ما را یاری داده‌اند که از ایشان نیز قدردانی می‌شود.

چاپ: چاپخانه سکه

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 ۱۱۹۶-۱۴۱۵۵ تهران.

* * *

GOZARESH-E-COMPUTER

(Computer Report)

Vol.13, No.110, December, January 1991-92

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

زبانهای برنامه‌سازی ۷

آرایه‌های تپنده ۱۵

فوتونیک ۲۰

بخش‌ها

اخبار ایران ۳

اخبار جهان ۴

نامه‌ها ۵

سرمقاله ۶

گزارش

سمینار آموزش و انفورماتیک ۲۵

کامپیوترهای چندرسانه‌ای ۲۸

به جای صفحه‌کلید ۲۹

المپیاد ملی انفورماتیک ۳۱

واژه‌نامه

حرف K و L ۳۲

اخبار ایران

ایستگاه کار سان در ایران

دو تن از مدیران اروپایی شرکت سان، آقایان Elie Simon و Mohammad Samer Tayan که به دعوت اولین کنفرانس بین‌المللی «کاربرد کامپیوتر در علوم و فنون پزشکی» در اصفهان حضور یافته بودند، در سمینارهای دیگری که با همکاری شرکت کامپیوتری نگاره در محل هتل عباسی اصفهان و هتل لاله تهران با حضور نمایندگان صنایع، سازمانهای دولتی و دانشگاهها ترتیب یافته بود، شرکت جستند. در این سمینارها ضمن مرور کلی برویژهای کامپیوترها و ایستگاههای کار سان، تاریخچه ده ساله شرکت سان و چگونگی تبدیل آن از شرکتی کوچک به بزرگترین فروشنده ایستگاه کار در دنیا مورد بحث و بررسی قرار گرفت. در پایان هر جلسه نیز به پرسشهای حاضران پاسخ داده شد. از جمله مهمترین پرسشها اخذ اجازه صدور این کامپیوترها از کشور آمریکا بود که آقای سیمون دریافت آن را برای دانشگاه امیرکبیر و تهران قطعی و حمل تجهیزات آنها را قریب الوقوع اعلام نمودند.

سمینار اسپارک

روز چهارشنبه ۱۱ دی ماه سمیناری با حضور دو طراح برجسته ایرانی، آقای دکتر سعید نامجو طراح اصلی پردازنده SPARC (که در کامپیوترهای سان به کار می رود) و مدیر کنونی بخش طراحی شرکت سان و آقای کامل مظفری کرمانی مدیر توسعه سخت افزار همان شرکت برگزار گردید. این سمینار با همکاری دانشکده انفورماتیک دانشگاه امیرکبیر، شرکت کامپیوتری نگاره و شرکت داده کاوی ایران در محل آمفی تئاتر دانشگاه امیرکبیر ترتیب یافته بود.

سمینار با حضور گروهی از استادان، دانشجویان و علاقمندان طراحی مدارهای یکپارچه و با سخنان آقای دکتر صفابخش آغاز شد. سپس دکتر نامجو به تفصیل علل و چگونگی شکل گیری ایده های اولیه

برای طراحی SPARC را بیان نمودند و تجربیات خود را در چگونگی پیشبرد پروژه و راهبری تیم تحت نظر خود شرح دادند. پس از آن آقای کرمانی تجربیات خود را در همکاری تیمی و چگونگی انجام پروژه های بزرگ، توضیح دادند. در نیمه دوم جلسه سخنرانان به پرسشهای حاضران پاسخ گفتند. از جمله نکات جالب مطرح شده این بود که هم اکنون حدود ۲۰۰ متخصص برجسته ایرانی در استخدام شرکت سان (در کالیفرنیا آمریکا) هستند.

نمایندگیهای آی بی ام

شرکت آی بی ام پس از هشت ماه مذاکره که با برخی از شرکتهای کامپیوتری ایران داشت با بررسی امکانات این شرکتهای پنج شرکت را از میان آنها به عنوان همکاران خود در ایران انتخاب نمود.

از بین این پنج شرکت، شرکتهای دی بی افئی و داده پردازی ایران به عنوان نماینده رسمی فروش (IBM Authorized Remarketter) نمایندگی فروش سخت افزارهای آی بی ام و نرم افزارهای آنها را به طور اعم، و شرکتهای دایاسیستم، مرکز ماشینهای اداری ایران و ترادیس به عنوان فروشنده رسمی (IBM Authorized Dealer) نمایندگی فروش سیستمهای شخصی آی بی ام (PS/1 و PS/2) را به طور اخص در ایران دارا خواهند بود. قراردادهای همکاری مذکور که در بهمن ماه سال جاری امضاء شده اند شامل استفاده از شبکه اطلاعات، خدمات آموزشی و پشتیبانی شرکت آی بی ام برای همکاران ایرانی این شرکت است.

لازم به توضیح است که قسمت عملیات مربوط به ایران در شرکت آی بی ام تحت نظر شاخه جنوب اروپا، خاور میانه و آفریقای آی بی ام (IBM SEMEA) اداره می شود و مقر آن در شهر رم (ایتالیا) است. مدیریت این قسمت را آقای اکبر کارزونی که از همکاران قدیمی آی بی ام و یکی از مدیران شعبه آی بی ام در ایران بوده اند به عهده دارند.

ارتباط با شبکه کامپیوتری آموزشی و پژوهشی

مرکز تحقیقات فیزیک نظری و ریاضیات که به نمایندگی از سوی ایران در شبکه آموزش و پژوهش اروپا (EARN) پذیرفته شده است، ارتباط موقت خود را با این شبکه برقرار نموده است. این شبکه، مهمترین شبکه کامپیوتری است که دانشگاهها و مراکز پژوهش اروپا، خاورمیانه و آفریقا را به یکدیگر مرتبط می سازند. امروزه بیش از هفتاد هزار پژوهشگر در هر ماه، روی بیش از سه هزار دستگاه کامپیوتر واقع در بیش از ششصد موسسه علمی در نزدیک به ۶۰ کشور از طریق این شبکه با یکدیگر تبادل اطلاعات می کنند.

از آنجا که خصوصیات EARN همانند خصوصیات شبکه BITNET (در آرژانتین، آمریکا، برزیل، تایوان، ژاپن، سنگاپور، شیلی، کره جنوبی، کلمبیا، مکزیک و هنگ کنگ) و شبکه NETNORTH (در کانادا) و GULFNET (در عربستان و کویت) می باشد، مبادله مستقیم داده ها و پیامها و پرونده ها بین همه این شبکه میسر است. همچنین از طریق تونلها به بیشتر شبکه های علمی دیگر مانند EU-INTERNET, CSNET, NSFNET, DEN, HEPNET, NET, JANET و NORDUNET نیز می توان دسترسی یافت در حال حاضر، تعداد گره ها در شبکه آموزش و پژوهش اروپا، ۸۹۲ گره و سایر شبکه های اصلی ۲۲۶۷ گره است.

هم اکنون یکی از کامپیوترهای مرکز تحقیقات فیزیک نظری و ریاضیات از طریق خط تلفن و با استفاده از مودم به گره دانشگاه لینتز اتریش در شبکه EARN متصل شده است.

اتصال دائم این مرکز با شبکه پس از نصب سیستم کامپیوتری خریداری شده این مرکز که یک سیستم MICROVAX 3100/20E است، در اوایل سال ۱۳۷۱ برقرار خواهد شد و طبق برنامه ریزی انجام شده قرار است تا پایان سال آینده دانشگاهها و موسسات علمی و

محیطهایی که می‌توانند از این ویژگی بهره گیرند، امکانات ارتباط بین شبکه‌ای نیز موجود خواهد بود.

درگذشتگان

خانم گریس هاپر از پیشگامان علم کامپیوتر در سن ۸۵ سالگی در آمریکا درگذشت. او در سال ۱۹۳۴ درجهٔ دکتری در ریاضی را از دانشگاه ییل گرفت و به تدریس در کالج دخترانهٔ وسار پرداخت. در سال ۱۹۴۳ به نیروی دریایی پیوست و از همان آغاز، به کار با کامپیوترهای ابتدایی آن زمان پرداخت. وی در طراحی کامپیوتر یونیواک، زبان کوپول و بسیاری از «اولین»های علم کامپیوتر شرکت داشت. ابداع اصطلاح دیباگ (debug) یعنی اشکال زدایی را به او نسبت می‌دهند. خود او طی خاطره‌ای از سالهای اولیهٔ کامپیوتر گفته‌است که روزی کامپیوتری به‌نحو مطلوب کار نمی‌کرده و هنگامی که در پی علت این امر می‌گشتند یکی از همکاران یک حشره (bug) را که در میان مدارهای کامپیوتر افتاده بود یافت و از آن به بعد اصطلاح «دیباگ» به معنی «حشره زدایی» به معنی زفع اشکال به کار رفت.



گریس هاپر در نیروی دریایی آمریکا درجهٔ دریاداری داشت و طی سالهای اخیر در کنفرانسهای کامپیوتری به عنوان سخنران افتخاری شرکت می‌کرد.

دسک‌تاپ معروف است نیز برنامه‌هایی را پشتیبانی می‌کند که با هر یک از استانداردهای OSF، پوزیکس ۱/۱۰۰۳، XPG3 و X/Open نوشته شده باشند. همچنین برنامه‌هایی که برای یونیکس‌هایی با پایهٔ پردازندهٔ میپس (مانند یونیکس شرکت دیجیتال معروف به اولتریکس) نوشته شده باشد در این سیستم قابل اجراست.

از لحاظ سخت‌افزاری دو دسته از سیستمها برای محیط ایس در نظر گرفته شده است. اول سیستمهایی با پردازنده‌های اینتل x86 و دوم سیستمهایی که مشخصات آدک (Advanced RISC Computing - ARC) را داشته باشند. مشخصات آدک بر اساس پردازنده‌های میپس توسط اعضای گروه ایس تعیین می‌شود و انتظار می‌رود که اولین محصولات سخت‌افزاری و نرم‌افزاری این گروه در سال ۱۹۹۲ عرضه شود.

شبکهٔ باز

شرکت Novell نسخهٔ ۳/۱۱ سیستم عامل شبکه‌ای NetWare را آماده کرده است و می‌گوید کامپیوترهایی که سیستم عامل داس، ویندوز، مکینتاش و یونیکس را تحت نسخهٔ جدید NetWare اجرا می‌کنند، قادرند اطلاعات و منابع یکدیگر را به سادگی به اشتراک گذارند. «نظام پروندهٔ شبکه‌ای» (Network File System - NFS) در نسخهٔ جدید NetWare که گونه‌ای از NFS شرکت سان است، برای برنامه‌های یونیکس امکانات چاپ و پرونده‌داری کاملاً طبیعی فراهم می‌کند. شرکت Novell همچنین می‌گوید که در هستهٔ سیستم عاملش برنامه‌های TCP/IP را نیز گنجانده، به این ترتیب در

پژوهشی کشور از طریق مرکز تحقیقات فیزیک نظری و ریاضیات به این شبکه متصل گردند. ایجاد یک شبکهٔ کامپیوتری ملی در سطح کشور، بدون شک نقش بسیار مهمی در اطلاع رسانی از فعالیتهای پژوهشی و هماهنگ‌سازی آنها و ارتقاء سطح علمی کشور خواهد داشت.

اخبار جهان

آشتی دادن یونیکس و کامپیوترهای شخصی

بیست و یک شرکت کامپیوتری به رهبری مایکروسافت، کامپک، میپس، و دیجیتال گرد هم آمده‌اند تا «محیط پیشرفتهٔ کامپیوتری» - ایس (Advanced Computing Environment - ACE) را عرضه کنند. هدف ایس تعریف محیط مشترک و استاندارد است که همسازی با کامپیوترهای شخصی موجود و سیستمهای یونیکس را دربرگیرد. سایر شرکتهایی که در این کار سهمی دارند عبارتند از: سونی، کوپوتا، نک، اولیوتی، زیمنس نیکسدورف، سنلیکان گرافیکس، تاندم و سانتاکروز آپریشن. غایبهای نامدار گروه عبارتند از سان، آی‌بی‌ام، هیولت‌پکارد و موتورولا.

برنامهٔ ایس سیستم عامل مایکروسافت به نام OS/2 3.0 و سیستم عامل یونیکس از سانتاکروز آپریشن را پشتیبانی می‌کند. OS/2 3.0 برنامه‌های کاربردی را که برای محیطهای ویندوز، OS/2 یا DOS نوشته شده باشد، و نیز برنامه‌هایی را که استاندارد پوزیکس را رعایت کرده باشند، پشتیبانی می‌کند.

محیط یونیکس سانتاکروز که به اوپن



نامه‌ها

از کتابخانه چه خبر؟

چندی است در گزارش کامپیوتر صحبتی از کتابخانه انجمن نبوده است. امیدوارم کتابخانه هم مانند سایر فعالیتهای انجمن مورد استقبال علاقه‌مندان زیادی قرار گرفته باشد و در حال رشد در خور خود باشد.

امیر سعید نایب‌ننده

انگلستان

کتابخانه انجمن فعلاً به علت در اختیار نداشتن محل مناسب دایر نیست ولی به مجرد فراهم شدن چنین مکانی، کتابها و مجلات موجود برای مطالعه در دسترس اعضا قرار خواهد گرفت. سردبیر

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.

منتشر شد

گزارش کامپیوتر

سال دوازدهم شماره‌های ۱ تا ۶

(شماره‌های بیایی ۱۰۰ تا ۱۰۵)

۱۳۶۹



انتشارات انجمن انفورماتیک ایران

بهای سالنامه ۲۰۰۰ ریال است. برای تهیه آن با دفتر انجمن تلفن ۶۵۶۶۸۷ تماس بگیرید.



سرمقاله

حساب و کتاب

کاری نداریم اعتنای زیادی به این برنامه‌ها نمی‌کنیم؟ یا شاید کاربران کامپیوترهای شخصی ما زیاد اهل حساب و کتاب نیستند و صرف وقت برای نوشتن برنامه‌های کوبول را به استفاده از برنامه‌های کاربرگ ترجیح می‌دهند؟ شاید خوانندگان با ارائه نظرات خود این امر را بهتر توضیح دهند. ولی من حداقل این را می‌دانم که کلمه «کاربرگ» را به «صفحه گسترده» ترجیح می‌دهم.

* * *

پس از برگزاری انتخابات هیأت اجرایی انجمن، و در اولین جلسه هیأت اجرایی جدید، تصویب شد که امر اداره کمیته انتشارات از سردبیری گزارش کامپیوتر جدا شود. در همین جلسه سرپرستی کمیته (پرزحمت) انتشارات به دوست و همکار گرامی آقای سعید وحید محول شد و کار سردبیری گزارش کامپیوتر نیز مجدداً به من واگذار شد یعنی خوانندگان گزارش کامپیوتر باید دو سال دیگر با خوب و بد این سردبیر بسازند.

چهار سال پیش که برای نخستین بار سردبیری را پذیرفتم گزارش کامپیوتر مجوز انتشار به صورت فعلی را نداشت و به صورت چند ورق کاغذ فتوکپی شده ارائه می‌شد. از آغاز سال ۱۳۶۹ اجازه انتشار به صورت فعلی داده شد. پس از چند شماره که کار حروف چینی را با دست انجام می‌دادیم از سیستم TEX پارک استفاده کردیم که به بهبود کیفیت ظاهری نشریه کمک کرد. طی این مدت از همکاریهای مادی و معنوی عده‌ای برخوردار بوده‌ایم که لازم می‌دانم از آنان تشکر کنم. همه کسانی که با ارسال مقاله‌های تألیفی و ترجمه باعث شده‌اند تا گزارش کامپیوتر به صورت نشریه‌ای آبرومند و پرمحتوا عرضه شود در صدر فهرست تشکر قرار می‌گیرند. از اعضای شرکت داده‌کاوی که با کمک روز و شب خود صورت فعلی نشریه را تأمین کردند به‌ویژه آقایان حسین بادبانچی و احمد یزدی پور همیشه تشکر کرده‌ام و باز هم تشکر می‌کنم. از مدیران شرکت داده‌سیستمهای ایران به‌ویژه آقایان داریوش و بهرام مهذب نیز تشکر می‌کنم که طی چهار سال گذشته با سعه صدر به من که کارمند شرکت بودم اجازه دادند بخش زیادی از وقت روزانه خود را صرف امور انجمن و گزارش کامپیوتر کنم و نیز از کمکهای دیگری که به انجمن کرده‌اند (مثل ارائه کمکهای مالی، تخصیص انباری برای ذخیره کردن کاغذهای گزارش کامپیوتر، ارائه خدمات حمل و نقل برای ارسال نسخه‌های چاپ شده به دفتر انجمن و پست کردن نسخه‌ها به اعضا و ...) نیز قدردانی می‌کنم و امیدوارم سایر شرکتهای کامپیوتری نیز از فعالیتهای انجمن پشتیبانی کنند.

سردبیر

این روزها در نوشته‌های فارسی مربوط به کامپیوتر به اصطلاح «صفحه گسترده» زیاد برخورد می‌کنیم. کسی که سابقه ذهنی نداشته باشد در برخورد اول در عجب می‌ماند که صفحه گسترده چیست و چه فرقی با صفحه نگسترده دارد! پس از بررسی معلوم می‌شود که منظور برنامه‌های کامپیوتری معروف به spread sheet است که از پرفروشترین انواع برنامه‌های کامپیوتری روی کامپیوترهای شخصی است. spread sheet در فرهنگ امریکایی و انگلیسی به ورقه‌های کاغذ ستون بندی شده نسبتاً عریضی گفته می‌شود که از ابزار کار حسابداران و مدیران مالی در تنظیم و تدوین حسابهاست. این ورقه در زبان حسابداران ایران به «فرم چند ستونی» یا «کاربرگ» معروف است. اتفاقاً نام دیگر (ورسمیت) spread sheet در انگلیسی work sheet است که دقیقاً معادل همان کلمه «کاربرگ» خودمان است.

کسانی که سعی کرده‌اند تا ستونهای متعددی از اعداد و ارقام را تنظیم کنند می‌دانند با تغییری در یک ستون تمام اعداد ستونهای دیگری که با آن در ارتباط هستند تغییر می‌کند و به همین دلیل مداد و مداد پاک‌کن از ابزار لازم برای تنظیم فرمهای چندستونی است. با ظهور کامپیوترهای شخصی در اوایل دهه هشتاد نوعی از برنامه کامپیوتری نیز به بازار آمد که کار تنظیم‌کنندگان فرمهای چندستونی یا کاربرگها را ساده می‌کرد. این برنامه‌ها در آن زمان electronic spread sheet نامیده می‌شد و به خاطر سهولت زیادی که در تنظیم فرمها به کاربر عرضه می‌کرد به سرعت جزو پرفروشترین نرم‌افزارهای کامپیوتر گردید و مقام دوم را پس از برنامه‌های کلمه‌پرداز از نظر تعداد نسخه‌های فروخته شده پیدا کرد. از VisiCalc که اولین کاربرگ الکترونیک بود تا Lotus 1-2-3 و Excel راه درازی طی شده و امکانات این برنامه‌ها هر بار بیشتر گشته ولی در قلب همه این برنامه‌ها امکان تنظیم ستونهایی از اعداد و امکان تعریف رابطه ستونها با یکدیگر قرار دارد. استفاده از این برنامه‌ها به قدری رایج شده‌است که حتی از آنها به عنوان ابزار شبیه‌سازی فرایندهای علمی و صنعتی نیز استفاده می‌کنند و مجموعه وسیع «تابع»های ریاضی که در دسترس کاربران است این کار را ساده می‌کند.

اما در محیط کار ایران با وجود افزایش روزافزون کامپیوترهای شخصی و گسترش نسبتاً زیاد برنامه‌های «کلمه‌پرداز»، برنامه‌های «کاربرگ الکترونیک» چندان با استقبال روبرو نشده‌است. علت این امر به درستی برای من روشن نیست. آیا فارسی نبودن برنامه‌ها عاملی در این امر است؟ آیا امریکائیه‌ها به دلیل مجبور بودن به تنظیم فرمهای مالیاتی سالانه خود قدر این نرم‌افزار را می‌شناسند و ما که چنین

مقاله

زبانهای برنامه‌سازی به‌مثابه مدل‌های تفکر (۱)

نوشته پیتز رخنبرگ*

ترجمه علی شکوفنده و عباس نودری

فکری می‌نامیم که رده‌های مختلف زبانهای برنامه‌سازی را از یکدیگر متمایز می‌سازد.

شکل ۱ بیانگر یک تقسیم‌بندی از زبانهای برنامه‌سازی از نقطه‌نظر مدل تفکر آنهاست. در اولین سطح، زبانها به دو گروه زبانهای فرمانی (که اغلب رویه‌ای نیز خوانده می‌شوند) و زبانهای غیرفرمانی (که به آنها زبانهای اعلانی^۲ نیز گفته می‌شود) تقسیم می‌شوند.

زبانهای فرمانی مبتنی بر مدلی است که در آن، محاسبه به‌عنوان یک «پردازش در زمان» نگریسته می‌شود. مجموعه مقادیر تمام متغیرها در یک نقطه از زمان، تعیین‌کننده حالت محاسبه بوده و معنی اجرای یک برنامه تبدیل حالتها در طی زمان است که با یک حالت اولیه شروع و به یک حالت نهایی خاتمه می‌یابد. تقریباً تمام زبانهای متداول برای داده پردازش عددی و عملیات مهندسی نرم‌افزار از بیسیک تا ایدا جزو این دسته از زبانها به حساب می‌آیند.

زبانهای رویه‌گرا در یک سطح پایین‌تر به دو گروه الگوریتم‌گرا و موضوع‌گرا تقسیم می‌شوند. در زبانهای الگوریتم‌گرا مرزبندی مشخص بین داده و الگوریتم وجود دارد و یک برنامه به‌عنوان الگوریتمی که مقادیر ورودی را به مقادیر خروجی تبدیل می‌کند در نظر گرفته می‌شود. در اینجا الگوریتم (به عنوان قوانین تبدیل حالت) در درجه اول توجه قرار می‌گیرد و داده‌ها در پشت صحنه قرار می‌گیرند. در زبانهای موضوع‌گرا، داده‌ها و الگوریتم در قالب واحدی به‌عنوان موضوع در نظر گرفته می‌شوند. یک الگوریتم، تنها فعل و انفعالی بین موضوعهاست و اجرای یک برنامه موضوع‌گرا را می‌توان به عنوان شبیه‌سازی یک سناریو در جهان واقعی یا مجرد در نظر گرفت که در آن، موضوعها (احتمالاً با یک ساختار بسیار پیچیده) با یکدیگر به فعل و انفعال می‌پردازند. موضوعهای موجود تغییر می‌یابند، موضوعهای جدید به‌وجود می‌آیند و موضوعهای بی‌استفاده نیز از بین می‌روند. برخی زبانها نظیر ایدا و ماجولا-۲ هر دو دسته خصوصیات موضوع‌گرایی و الگوریتم‌گرایی را دارند.

در زبانهای غیر فرمانی، دیدگاههای تبدیل حالت با مفاهیم دیگری جایگزین می‌شود. زبانهای عملیاتی مدل تفکرشان را بر مفهوم توابع ریاضی استوار می‌سازند که مستقل از زمان و به دور از تأثیرات جنبی است. زبانهای منطقی، مدل تفکرشان را بر مفهوم رابطه منطقی بنا

declarative ^۳

در این مقاله که بخش نخست آن در این شماره از نظراتان می‌گذرد، زبانهای برنامه‌سازی معاصر از نظر مدل تفکرشان مورد مقایسه و ارزیابی قرار می‌گیرند. منظور از مدل تفکر، خصوصیات یک خانواده از زبانهای برنامه‌سازی است که بر روی مکانیزم فکری برنامه‌سازان هنگام نوشتن یا خواندن برنامه‌ای به یکی از زبانهای آن خانواده، تأثیر می‌گذارد. زبانهای در نظر گرفته شده عبارتند از زبانهای فرمانی دنباله‌ای^۱ (شامل دو زیر خانواده الگوریتم‌گرا^۲ و موضوع‌گرا^۳) و زبانهای غیر فرمانی (شامل دو زیر خانواده عملیاتی و منطقی^۴). مفاهیم متمایزکننده متغیرها و حالتها مورد تأکید قرار می‌گیرند و زمینه‌های کاربردی در مورد هر جنبه مشخص می‌شود. در نتیجه این بررسی زبانهای برنامه‌سازی به دو دسته مقدارگرا^۵ و موضوع‌گرا تقسیم می‌شوند. ادعا شده است که مفهوم الگوریتم به‌عنوان پایه علوم کامپیوتر باید با مقوله دیگری که پردازشها و ارتباط آنها را مدل‌سازی می‌کند جایگزین گردد. زبانهای موازی و ترکیبات مدل‌های مختلف تفکر در این مقاله مورد توجه نیستند.

مقدمه

در بسیاری از کتابها، زبانهای برنامه‌سازی به طور منفرد مورد بررسی قرار گرفته‌اند و در برخی از آنها نیز رده وسیعتری از زبانهای متفاوت با هم مقایسه شده‌اند، از جمله کتابهای پرات [7] و گتزلی و جزایری [6]. زبانها را می‌توان از نقطه نظرهای متفاوت مورد بررسی و مقایسه قرار داد، به‌عنوان مثال می‌توان جنبه‌های نحوی^۵، کنترلی، داده‌ای، تجربیدی، پیاده‌سازی و کاربردی را نام برد. در این مقاله سعی شده است که زبانها از نقطه نظر شیوه تفکری که به برنامه‌ساز خود القاء می‌کنند مورد ارزیابی قرار گیرند. در حال حاضر چنین مرسوم است که این دیدگاه بررسی را دیدگاه الگوریتم^۶ بنامند. اما به نظر ما این نامگذاری گمراه‌کننده است و به همین خاطر ما آن را مدل‌های تفکر یا عاداتهای

Peter Rechenberg, "Programming Languages as Thought Models", *Structured Programming*, no. 11, 1990, Springer Verlag.

- 1 sequential imperative languages
- 2 algorithm-oriented
- 3 object-oriented
- 4 value-oriented
- 5 syntactical
- 6 paradigm-oriented

radius
1234 [20]
نام: radius
نشانی: 1234
مقدار: 20

شکل ۲- سه عنصر تعیین کننده یک متغیر

برای هر متغیر سه عنصر تعیین کننده را می توان برشمرد: نام، نشانی، مقدار (شکل ۲).

در برنامه مبدا^۱، هر متغیر توسط نامش مشخص می شود، در خلال اجرای برنامه دارای نشانی ثابتی در حافظه است که توسط آن قابل دستیابی است و همچنین یک مقدار را در خود نگهداری می کند. مقدار یک متغیر می تواند توسط عملیات تخصیصی تغییر کند که در نتیجه، دنباله زمانی مقادیر یک متغیر، بیانگر سابقه متغیر در طی عملیات محاسباتی خواهد بود. مفهوم اخیر توسط فرمول زیر به طور دقیق مشخص می شود:

$$x := x + 1$$

مشکلات این مدل چیست؟ به طور مشخص این مشکلات از دو ناحیه بروز می کنند:

۱- استفاده از نامهای مختلف برای یک ناحیه ثابت از حافظه (اشتراک^۱).

۲- به کارگیری نامهایی برای متغیرها در برنامه مبدا با معنی متفاوت. نام x می تواند بیان کننده هر یک از حالات زیر باشد:

- مقدار متغیر x (در عبارتهای محاسبه مقدار)،
- نشانی متغیر x (در اعلانها و در سمت چپ احکام تخصیصی)،
- ارجاعی به یک متغیر دیگر (اگر x یک متغیر اشاره گر باشد).

با تداخل دو اثر فوق، مسئله پیچیده تر نیز می شود. حتی اگر یک زبان برنامه سازی اجازه اشتراک را به طور صریح ندهد، دو متغیر از نوع اشاره گر می توانند به فضای یکسانی اشاره کنند و حالت اشتراک را به وجود آورند. در نتیجه نام یک متغیر، بنابر معنی زبان برنامه سازی می تواند دارای سه مفهوم باشد. بر اساس نظریه کلیولند [4] مفاهیم فوق را می توان، معنی مقداری، معنی حافظه ای و معنی اشاره گری نام نهاد.

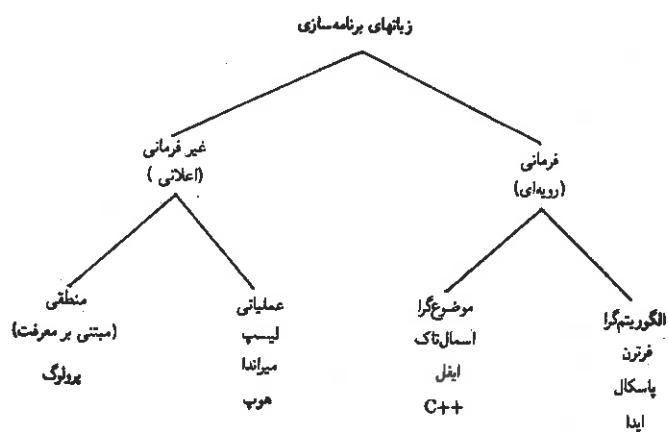
با توجه به معنی مقداری^{۱۰}، یک متغیر تنها یک مقدار است و مقوله های حافظه، تغییر حالت یا اشتراک در این مفهوم نادیده گرفته می شود. در معنی حافظه ای، یک متغیر فضایی از حافظه است که حاوی یک

source program ^۸
sharing ^۹
value semantics ^{۱۰}

می سازند. به این معنی که نتیجه $x+y$ را می توان به صورت رابطه $Add(x,y,z)$ تعریف کرد، با این معنی که « z حاصل جمع اعداد x و y است.» رابطه ها می توانند دارای ارزش درست یا نادرست باشند. اگر x و y متغیرهایی با مقدار عددی باشند و z تا این لحظه مقداری به خود نگرفته باشد، الگوریتم مقدار z را چنان تعیین می کند که رابطه $Add(x,y,z)$ ارزش درست یابد. چنین مدل تفکری اغلب عاری از تأثیرات جنبی است و ایده حالت در آن چندان مورد نظر نیست. از آنجایی که برنامه های تولید شده در زبانهای عملیاتی و منطقی بیشتر شبیه به مشخصات مسأله ای که باید حل شود هستند تا تشریح یک راه حل برای آن، چنین زبانهایی را غالباً اعلانی می نامند. چنانچه در ادامه نشان داده خواهد شد، این تنها یک نامگذاری است نه یک واقعیت. بر خلاف معانی لغوی، برنامه سازانی که با زبانهای اعلانی کار می کنند نیز باید به منظور اجرای صحیح برنامه هایشان، تفکر الگوریتمی بروز دهند. در ادامه این مقاله، بیشتر سعی بر مشخص کردن مدل های تفکر جهت ارائه درک درست از محدودیتها و محدوده عملیاتی آنهاست. کلیه این مدل های تفکر عمدتاً برای نمایش محاسبات دنباله ای به کار می روند ولی حداقل از نظر تئوری برای نمایش محاسبات موازی نیز مناسب هستند. این دیدگاه در این مقاله مورد توجه نیست.

مدل تفکر دستوری

این مدل تفکر بر مبنای یک مدل ماشین واقعی و به ویژه ماشین فون نویمان بنا شده است. مفهوم اصلی آن در نظر گرفتن متغیرها به عنوان مخزن نگهداری مقادیر است. کامپیوتر واقعی دارای حافظه ای است متشکل از یک دنباله خطی از سلولها که توسط نشانی قابل دسترسی هستند. یک متغیر، آن گونه که در برنامه معرفی می شود و مورد استفاده قرار می گیرد متناظر با بخشی از فضای حافظه است.



شکل ۱- یک دسته بندی از زبانهای برنامه سازی

در پس زمینه قرار دارند. الگوریتم، اصلی و ماندنی است و داده‌ها فرعی و گذرا.

• یک محاسبه (حداقل از نظر تئوری) یک فراروند خودکار در بستر زمان است که معمولاً با مجموعه‌ای از داده‌های ورودی شروع می‌شود و با تبدیل آنها به اطلاعات خروجی خاتمه می‌یابد. فرض بر این است که هیچ تاثیری از جهان خارج در حد فاصل مراحل شروع و خاتمه وجود ندارد. داده‌های خروجی در خلال محاسبات، یعنی فعل و انفعال متقابل انسان و ماشین، در نظر گرفته نمی‌شود.

• یک محاسبه متشکل از دنباله محدودی از گام‌ها است که با تولید خروجیهای مورد نظر خاتمه می‌یابد.

توجه داشته باشید، کاربردهایی که در آنها برنامه‌ها تنها بخشی از یک سیستم بزرگتر هستند، خصوصاً قسمتی از یک سیستم انسان-ماشین (نظیر سیستمهای عامل، پایگاه داده‌ها، خودکار سازی دفاتر کاری، شبیه‌سازی در در کنترل ترافیک و غیره)، جزو این مدل محسوب نمی‌شود. این سیستمها نتیجه‌ای مشخص به‌عنوان خروجی تولید نمی‌کنند و عمدتاً باید به‌طور بی‌پایان حلقه‌ای را تکرار کنند.

مدل تفکر فوق بهترین مدل شناخته شده است. بسیاری از زبانها نظیر فرتن، پاسکال و ایدا در این مدل جای دارند. اگر چه مدل مشترک کلیه این زبانها یکسان است ولی از بسیاری نظرها باهم متفاوتند. تفکر در محیط فرتن به معنی تفکر در جهان متشکل از تخصیص ایستای حافظه^{۱۶}، ساختار داده‌های ساده و ترجمه مستقل رویه‌ها است. پاسکال علاوه بر خصوصیات فوق شامل تخصیص پویای حافظه^{۱۷}، داده‌گونه‌های^{۱۸} بسیار ساختیافته و قابل تعریف توسط استفاده کننده، برگشت پذیری^{۱۹} و اشاره‌گرها است. ماجولا-۲ و ایدا داده‌گونه‌های تجریدی و عملیات موازی را نیز به مفاهیم قبل می‌افزایند. توازی در ماجولا-۲ به شکل سطح پایین توابع همروال^{۲۰} و در ایدا با شکل بالاتر فراروندهای دنباله‌ای^{۲۱} تجلی می‌یابد. همچنین در ایدا مفاهیمی نظیر پیمانه‌های غیراختصاصی^{۲۲} و روالهای پردازش استثنائات^{۲۳} نیز وجود دارند. هر یک از این خصوصیات، مدل تفکر را تا حدی اصلاح می‌کنند ولی همه آنها، مواردی از مدل الگوریتم‌گرایی فرمانی هستند. در همین مدل زبانهایی قرار دارند که روی به جهان سطح بالاتری می‌کنند. در زبان ای‌پی‌ال، قابلیت اعمال عملگرهای بسیار پیچیده روی ساختارهای داده‌های پیچیده با حذف بسیاری از کنترلها (مثل حلقه زنی) وجود دارد. در زبان اسنوبول-۴ نوع خاصی

مقدار است. در معنی اشاره‌گری، یک متغیر اشاره‌گری به یک ناحیه از حافظه است که حاوی یک مقدار است. زبانهای فرمانی اغلب هر سه مفهوم بالا را به‌کار می‌گیرند. یک متغیر غیر اشاره‌گر در یک عبارت محاسباتی، بر طبق معنی مقداری عمل می‌کند و در یک دستورالعمل اعلانی و یا در سمت چپ یک دستورالعمل تخصیصی بر طبق معنی حافظه‌ای. حالت برای متغیر اشاره‌گر بسیار پیچیده‌تر است زیرا یک متغیر اشاره‌گر مانند p گاه نشانگر مرجعی برای دستیابی به یک عنصر است (دلالت واقعی^{۲۱}) و گاهی نیز نشانگر عنصری که به آن اشاره می‌کند (دلالت مجرد^{۲۲}). فرض کنید p و q اشاره‌گرهایی به یک رکود از نوع R باشند و p در حال حاضر به x و q به y اشاره کنند (x و y هر دو از نوع R هستند)، حال دستورالعمل تخصیصی $q := p$ را در نظر بگیرید، در دلالت واقعی این دستورالعمل بیانگر تخصیص اشاره‌گر است، بدین معنی که بعد از اجرای این دستورالعمل، p و q هر دو به فضای x اشاره می‌کنند. در دلالت مجرد، این دستورالعمل بیانگر تخصیص مقدار x به y است، که به طور کلی مفهوم دیگری را ارائه می‌دهد.

مدل الگوریتم‌گرا مبتنی بر این ایده است که الگوریتم، داده‌های ورودی را به داده‌های خروجی تبدیل می‌کند.

اشتراک باعث تغییر ناخواسته و در عین حال نادیدنی مقدار یک متغیر در حین تخصیص به یک مقدار دیگر می‌شود. اشتراک و دلالت‌های دوگانه، نوشتن برنامه‌ها را پر خطا و درک آنها را مشکل می‌سازد. هر دو صورت الگوریتم‌گرا و موضوع‌گرایی زبانهای فرمانی از این مفهوم برای متغیرها، هم سود و هم زیان می‌برند.

مدل فرمانی الگوریتم‌گرا

مدل الگوریتم‌گرا مبتنی بر این ایده است که الگوریتم، داده‌های ورودی را به داده‌های خروجی تبدیل می‌کند. این طرز تفکر را معمولاً خط مشی ماشین تیورینگ^{۲۳} یا خط مشی ماشین خودکار^{۲۴} یا خط مشی تغییر حالت^{۲۵} نیز می‌نامند. خصوصیات اساسی این طرز تفکر با مفاهیم زیر بیان می‌شود:

• یک محاسبه فراروندی است گام به گام در بستر زمان که از داده‌ها استفاده می‌کند، الگوریتم و کنترل آن در پیش زمینه و داده‌ها

static storage allocation	۱۶
dynamic storage allocation	۱۷
data type	۱۸
recursion	۱۹
coroutine	۲۰
communicating sequential processes	۲۱
generic modules	۲۲
exception handling	۲۳

concrete denotation	۲۱
abstract denotation	۲۲
Turing machine approach	۲۳
automata theoretical approach	۲۴
state changing approach	۲۵

محاسبه در دیدگاه موضوع‌گرا شامل ارتباط بین موضوعهای فعال از طریق پیامهاست که به ایجاد موضوعهای جدید، از بین رفتن موضوعهای بی‌استفاده، تغییر داده‌ها در موضوعهای موجود و خروجی اطلاعات می‌انجامد. بدین ترتیب برنامه‌سازی موضوع‌گرا را می‌توان به عنوان شبیه‌سازی مجموعه‌ای از موضوعهای فعال و مرتبط در نظر گرفت.

پیش‌تاز چنین تفکری در برنامه‌سازی زبان سیمولا^{۲۸} (یکی از زبانهای معروف شبیه‌سازی) در اواخر دهه ۶۰ بود ولی این ایده تا رسیدن دهه ۸۰ و ظهور زبان اسمال‌تاک^{۲۹}، نفوذ و گسترش زیادی نیافت. چرخشی که از دیدگاه الگوریتمی به دیدگاه موضوع‌گرا به وجود آمده است به بهترین نحوه توسط مفاهیم عبارات «داده‌گونه تجریدی» و «رده»^{۳۰} نمایانده می‌شود. در زبانهای پیمانه‌ای مانند ماجولا-۲، یک داده‌گونه تجریدی در یک پیمانه پیاده‌سازی می‌شود. به عنوان مثال پیمانه زیر می‌تواند بیانگر مشخصات یک انباره باشد:

```
DEFINITION MODULE StackMod;
  TYPE Stack ; (*opaque*)
  PROCEDURE NewStack(VAR s:Stack);
  PROCEDURE Push(s:Stack; x:INTEGER);
  PROCEDURE Pop(s:Stack; VAR x:INTEGER);
  END StackMod.
```

این مثال از دو جهت جالب توجه است:

- پیمانه فوق نوع انباره را تعریف می‌کند ولی خود از نوع انباره نیست.
- هر رویه، دارای پارامتری از نوع انباره است.

استفاده کننده این پیمانه می‌تواند با اعلان متغیرهایی مانند

```
VAR a,b,c: Stack;
```

چند انباره را تعریف کند و متعاقب آن رویه‌های NewStack، Push و Pop را روی آنها به کار ببرد.

دیدگاه موضوع‌گرا در این مورد متفاوت است. در این دیدگاه، تعریف یک رده به شکل زیر جایگزین تعریف پیمانه می‌شود:

```
DEFINITION CLASS Stack;
  VAR stack : ...;
  (*the stack itself, e.g., as array*)
  Create;
  Push(x:INTEGER);
  Pop(VAR x:INTEGER);
  END Stack.
```

Simula ۲۸
Smalltalk ۲۹
class ۳۰

از داده‌ها و ساختار کنترل وجود دارد که الگو خوانده می‌شود و امکان تعریف عملگرهای بسیار سطح بالا را روی رشته‌های نویسه‌ای (مانند تحلیل کامل نحوی) در اختیار می‌گذارد. تمام این مساعی جهت متعالی کردن تفکر و استدلال درباره برنامه‌ها در سطحی مجردتر است

زبانهای پیمانه‌ای^{۲۴} نظیر ماجولا-۲ و ایدا دارای خصوصیتی هستند که آنها را به زبانهای موضوع‌گرا نزدیک می‌سازد. یک پیمانه، قابلیت تعریف داده‌گونه‌های تجریدی را در اختیار می‌گذارد و در نتیجه، مفهوم اصلی زبانهای موضوع‌گرا را در بر می‌گیرد ولی پیمانه در این زبانها مانند زبانهای موضوع‌گرا قابلیت استفاده مجدد را ندارد.

مدل تفکر فرمانی موضوع‌گرا

برای بسیاری از کاربردها مدل تفکر الگوریتم‌گرا مناسب نیست. برای مثال در یک سیستم پایگاه داده‌ها، داده‌ها مهمترین عناصر به شمار می‌روند و معمولاً دوره حیاتشان از الگوریتمهایی که آنها را به وجود می‌آورند و مورد دستیابی و تغییر قرار می‌دهند بیشتر است. سیستمهای عامل نیز فراوندها و منابع را در یک حلقه بی‌پایان، بدون محاسبه نتیجه خاصی شبیه یک الگوریتم، کنترل می‌کنند. برنامه‌سازی فعل و انفعال متقابل انسان و ماشین نیز نتیجه‌ای تولید نمی‌کند ولی عناصر از قبیل پنجره‌ها^{۲۵}، فهرستها^{۲۶} و غیره را از طریق محاوره با استفاده کننده کنترل می‌کند. کلیه شبیه‌سازیهای کامپیوتری، نگاشتهایی از موضوعات جهان واقعی یا تصویری و فعل و انفعال متقابل آنها به موضوعاتی در داخل کامپیوتر هستند.

در یک سیستم پایگاه داده‌ها، داده‌ها
مهمترین عناصر به شمار می‌روند و
معمولاً دوره حیاتشان از الگوریتمهایی که
آنها را به وجود می‌آورند و مورد دستیابی
و تغییر قرار می‌دهند بیشتر است.

چنین کاربردهایی به همراه ایده بسته‌بندی^{۲۷} داده‌ها و داده‌گونه‌های تجریدی به مدل تفکر فرمانی موضوع‌گرا می‌انجامد. در اینجا، داده‌ها هم به اندازه الگوریتم اهمیت می‌یابند و هر دو در یک مفهوم بالاتر ترکیب می‌شوند: «موضوع». هر الگوریتم را می‌توان متعلق به داده‌هایی که روی آنها عمل می‌کند، دانست، داده‌های خنثی در ترکیب با الگوریتم‌های فعال، موضوعهای فعال را به وجود می‌آورند. یک

module-oriented ۲۴
window ۲۵
menu ۲۶
encapsulation ۲۷

معرفی نمود، سپس می‌توان رده‌های چند ضلعی و دایره را به‌عنوان رده بعدی اشکال مسطح بسته تعریف کرد. آنها مولفه‌های رده پیشین خود را به‌ارث می‌برند و مولفه‌های جدید خود را نیز به آن می‌افزایند و یا آنها را برای خود تغییر می‌دهند. با استفاده از خاصیت وراثت می‌توان سلسله مراتب وسیعی از رده‌ها را با یک رده ساخت به‌نحوی که عمومی‌ترین آنها در بالاترین سطح قرار گیرد. این مفهوم بسیار مفیدی است زیرا با روشی که انسانها به سازماندهی دیدگاه خود از جهان می‌پردازند مطابقت دارد.

در زبانهای موضوع‌گرا افزودن یک رده جدید به‌عنوان پالایشی از یک رده موجود نباید به ترجمه مجدد رده موجود نیاز داشته باشد. این مفهوم از به‌کارگیری مجدد مولفه‌های نرم افزاری به روشی عمومی‌تر از شیوه‌های متداول در زبانهای مرسوم پشتیبانی می‌کند.

انقیاد پویا نتیجه مستقیم وراثت است. برای مثال رده مستطیل را به عنوان رده تمام مستطیها و رده مربع را به‌عنوان رده تمام مربعها در نظر بگیرید. مربع یک زیرخانواده مستطیل است. این دو رده دارای رویه‌های متفاوتی جهت محاسبه قطر هستند که در هر دو رده ممکن است نام یکسانی داشته باشد ولی با نوع پارامترهای مختلف: قطر رده مستطیل $\text{Diagonal}(x:\text{Rectangle})$ قطر رده مربع $\text{Diagonal}(x:\text{Square})$ این امر شبیه بارکردن اضافی^{۳۳} در زبان ایدا است. حال دو متغیر زیر را در نظر بگیرید:

```
var p: Rectangle;
    q: Square;
```

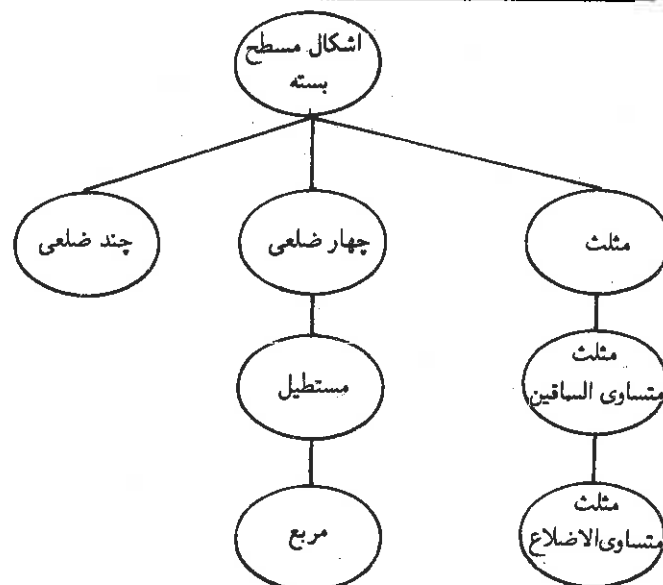
و همچنین عبارت تخصیصی

```
p:=q;
```

با وجود اختلاف در نوع p و q استفاده از چنین عبارت تخصیصی باید به‌خاطر آن که یک مربع، یک مستطیل نیز هست مجاز باشد. حال این سؤال پیش می‌آید که نوع متغیر p بعد از تخصیص چیست؟ در پاسخ باید گفت که متغیر p دارای دو نوع متفاوت است: مستطیل، نوع ایستا و مربع، نوع پویای آن است. عبارت تخصیصی بالا، متغیر p را به‌طور پویا به نوع مربع مقید می‌سازد. حال اگر فراخوانی رویه زیر را در نظر بگیریم:

```
Diagonal(p)
```

سیستم موظف است که نوع صحیح رویه Diagonal را بسته به نوع پویای p برای اجرا انتخاب کند. چون این کار نمی‌تواند در زمان ترجمه (کامپایل) صورت گیرد، انقیاد پویا ممکن است پر هزینه باشد. از متن ظاهری برنامه بر می‌آید رویه Diagonal یک رویه چندریختی^{۳۴} و پارامترهای ورودیش نیز از نوع چندریختی هستند.



شکل ۳- سلسله مراتب اشکال هندسی

بر خلاف پیمانه، یک رده دارای خصوصیات زیر است:

- رده نه تنها معرف نوع انباره بلکه خود از نوع انباره است.
- رویه‌ها دارای پارامتری از نوع انباره نیستند.

در زبانهای موضوع‌گرا موضوعهایی از نوع انباره را می‌توان به شکل زیر تعریف کرد:

```
VAR a,b,c: Stack;
```

ولی برای به‌کارگیری انباره باید از شکل متفاوتی استفاده شود:

```
a.Create; a.Push(x); a.Pop(x);
```

نشان‌گذاری اخیر بیانگر این دیدگاه است که یک موضوع از رده Stack شامل چهار مؤلفه است، خود انباره و سه رویه وابسته به آن. عبارت $a.\text{Push}(x)$ بدین معنی است که «پیام $\text{Push}(x)$ را برای موضوع a بفرست». موضوع a پس از دریافت پیام، خود را (یعنی داده‌های خود را) با الحاق x به مؤلفه Stack خود تغییر می‌دهد. در حالی که پیمانه StackMod و رویه‌های آن فقط یکبار وجود دارند رویه‌های از رده Stack باید در موضوع از رده Stack وجود داشته باشند.

مفهوم موضوعهای فعال با دو خصیصه دیگر توسعه می‌یابد: وراثت^{۳۱} و انقیاد پویا^{۳۲}. برای مثال سلسله مراتب اشکالی هندسی که در شکل ۳ نشان داده شده است را در نظر بگیرید. تمام اشکال مسطح خصوصیات مشترکی دارند (مساحت، محیط)، تمام چند ضلعیها نیز یک مجموعه خصوصیات مشترک دارند (برای مثال تعداد اضلاع) و به همین ترتیب چنین ساختاری می‌تواند در یک سیستم رده‌بندی نیز مدل‌سازی شود. در معرفی این به‌عنوان بالاترین سطح می‌توان رده موضوعی اشکال مسطح بسته را با دو رویه مساحت و محیط

overloading ۳۳
polymorphic ۳۴

inheritance ۳۱
dynamic binding ۳۲

- وراثت: رده‌های جدید را می‌توان از رده‌های موجود به دست آورد. رده‌های حاصله کلیه خصوصیات رده قبلی را به ارث می‌برند مگر آنکه صریحاً به عنوان مشخصه جدیدی تعریف شوند. حاصل این امر، ایجاد خانواده‌ای سلسله مراتبی از رده‌هاست.
- انقیاد پویا: متغیرها می‌توانند در زمان اجرا به موضوعهایی از رده‌های متفاوت مقید گردند. در این حالت، فراخوانی یک رویه (پیام) باید بر اساس رده‌های پویای پارامترهایش، در مورد انتخاب یکی از رویه‌های موجود، تصمیم‌گیری کند.

مدل تفکر عملیاتی

بر اساس مطالبی که پیرامون مدل تفکر فرمانی اعم از الگوریتم‌گرا یا موضوع‌گرا گفته شد می‌توان این خصوصیت را که متغیرها، پیمانه‌ای جهت مقادیر هستند (مدل پیمانه‌ای متغیرها)، به عنوان یک نقطه ضعف در این مدل تفکر محسوب نمود. این نکته و همچنین تمایل به ایجاد سطوح انتزاعی در تشریح محاسبات در دهه ۶۰، شرایط طرح ایده بیان محاسبات به عنوان توابع ریاضی را فراهم کرد. در ریاضیات برای محاسبه مقادیر ناشناخته از روی مقادیر شناخته شده، هرگز مقوله‌هایی نظیر زمان و محل‌هایی جهت نگهداری مقادیر میانی در نظر گرفته نمی‌شود. فرض کنید تابعی به شکل $D(e, x)$ با هدف محاسبه مشتق عبارت e نسبت به متغیر x در نظر گرفته شود، یعنی $\frac{de}{dx}$. اگر عبارت e تنها شامل عملگرهای جمع و ضرب باشد، مشتق گیری از آن می‌تواند به شکل زیر تعریف شود:

$$D(e, x) = 0 \quad \text{اگر } e \text{ یک مقدار ثابت یا عبارتی بر حسب}$$

متغیری بجز x باشد

$$D(x, x) = 1$$

$$D(e_1 + e_2, x) = D(e_1, x) + D(e_2, x)$$

$$D(e_1 \times e_2, x) = e_1 \times D(e_2, x) + D(e_1, x) \times e_2$$

بد نیست تفاوت‌های این تابع ریاضی با مفهوم تابع در یک زبان فرمانی را بررسی کنیم:

- یک متغیر در توابع ریاضی تنها بیانگر یک مقدار است و دلالت بر محتوی یک ناحیه از حافظه نمی‌کند و به محض آنکه متغیری به یک مقدار عددی مقید شد، تاثیر این انقیاد تا خاتمه کامل محاسبات باقی می‌ماند.
- پارامتر زمان در انجام محاسبات دخیل نیست. این بدین معنی است که در توابع ریاضی، اصل مطلب وجود یک ارتباط بین سمت راست و سمت چپ تعریف تابع است و نباید تابع را به عنوان دنباله‌ای از عملیات در بستر زمان در نظر گرفت و تابع تنها یک شیوه صوری جهت توصیف مسئله است.

توابع ریاضی به سه روش قابل پیاده‌سازی هستند:

زبانهای موضوع‌گرا طیف گسترده‌ای از زبانهای برنامه‌سازی را شامل می‌شوند. بسیاری از آنها از زبانهای الگوریتم‌گرا مشتق شده‌اند مانند ++C، آبجکت-پاسکال و آبجکتیو-سی برخی از آنها نظیر اسمال‌تاک و ایفل زبانهای موضوع‌گرای محض هستند. در این میان، اسمال‌تاک یکی از برجسته‌ترین زبانهای موضوع‌گراست. یک برنامه در زبان اسمال‌تاک دنباله‌ای از عبارات است که بسیاری از این عبارات دارای ساختار نحوی ساده $x \ P$ هستند که در آن، x یک موضوع و P یک پیام است. عبارت $x \ P$ بدین معناست که «پیام P را روانه موضوع x بنما و پاسخ موضوع x را به عنوان نتیجه عبارت، بازگردان». این عبارت را می‌توان به فراخوانی یک رویه به صورت $P(x)$ در یک زبان الگوریتم‌گرا تشبیه کرد. هر عنصر در زبان اسمال‌تاک، حتی یک مقدار عددی مانند 4 را می‌توان یک موضوع محسوب نمود که قابلیت دریافت یک پیام را داراست. یک عبارت نظیر $a+4$ که شبیه یک عبارت عادی به شکل میانوندی^{۳۵} است، در زبان موضوع‌گرا به گونه کاملاً متفاوتی در نظر گرفته می‌شود. عدد 4 را می‌توان به عنوان موضوع x در نظر گرفت و $a+4$ را پیام P محسوب کرد. چنین پیامی رایک پیام دودویی شامل خبر «جمع» و پارامتر a می‌نامند. کل عبارت $a+4$ بدین معنی است که «پیام a را جمع کن» را برای موضوع 4 بفرست و نتیجه را که حاصل جمع مقدار موضوع 4 و مقدار موضوع a است، بازگردان». این تفسیر غیر عادی از یک عبارت ساده بیانگر این است که در زبانی نظیر اسمال‌تاک شیوه تفکری جدید بر برنامه حاکم است. در نظر گرفتن یک «عدد» به عنوان یک موضوع که خود از پیامهای معینی تبعیت می‌کند بیانگر انحراف وسیعی از تفکر رایج در زبانهای برنامه‌سازی است و اغلب زبانهای موضوع‌گرا تا این حد پیش نمی‌روند. ولی سعی در تفکر با مدل موضوع‌گرا و تمام پیامدهای آن، تجربه جالبی است.

یک برنامه مجموعه‌ای از موضوعهای

فعال است که از طریق ارسال پیامها بر

یکدیگر عمل می‌کنند.

در خاتمه این بحث، مشخصات مدل تفکری موضوع‌گرا را بر حسب ویژگیهایش بیان می‌کنیم:

- موضوعهای فعال: یک برنامه مجموعه‌ای از موضوعهای فعال است که از طریق ارسال پیامها بر یکدیگر عمل می‌کنند.
- رده‌ها: هر موضوع به یک رده خاص تعلق دارد یعنی، داده‌گونه تجربیدی کتابخانه‌ای از رده‌های از پیش تعریف شده دارد و برنامه‌سازی می‌تواند از آنها به عنوان مواد اولیه ساخت یک برنامه استفاده کند.

هستند) با این وجود، زبانهای عملیاتی فاقد مفهوم حالتند ولی در عوض، دو خصیصه توابع بالا رتبه^{۳۷} و ارزیابی ضعیف^{۳۸} را ارائه می‌کنند که زبانهای فرمانی فاقد آنها هستند. به واسطه وجود همین دو خاصیت، نرم‌افزارهای تولید شده با استفاده از این زبانها نسبت به زبانهای فرمانی از حجم کمتری برخوردار هستند.

منظور از توابع بالا رتبه، توابعی هستند که پارامترهایشان خود تعریف یک تابع دیگر است. اگر چه بعضی از زبانهای الگوریتم‌گرا نیز پارامترهایی از نوع «رویه» را مجاز می‌دارند ولی شکل به‌کارگیری‌شان در زبانهای عملیاتی بسیار کلیتر است زیرا اولاً نه تنها اسم توابع بلکه تعریف کامل آنها نیز می‌تواند جزو پارامترها باشد و ثانیاً یک تابع می‌تواند به عنوان مقدار، تعریف تابع دیگری را حمل کند که این خاصیت در زبانهای فرمانی وجود ندارد. ذکر مثالی در این مورد از حوصله این بحث خارج است زیرا جهت بیان تعریف توابع باید با مقوله نشان‌گذاری آشنایی داشت. ارزیابی ضعیف (یا فراخوانی برحسب نیاز^{۳۹}) یک شیوه خاص از ارزیابی پارامترها است. روش رایج برای ارزیابی پارامترهای واقعی که مقادیر ثابت یا متغیرهای ساده نیستند، فراخوانی با مقدار^{۴۰} است: داخلی ترین پارامترهای واقعی در ابتدا و به‌طور کامل محاسبه و ارزیابی می‌شوند (ارزیابی قوی^{۴۱}). در زبان الگول-۶۰ شیوه فراخوانی بانام^{۴۲} نیز شناخته شده است: یک پارامتر واقعی، هر بار که مقدارش در محاسبات مورد باشد، به‌طور کامل ارزیابی و محاسبه می‌شود. «فراخوانی براساس نیاز» بین این دو قرار می‌گیرد: یک پارامتر واقعی، یک بار به‌هنگامی که مقدارش برای نخستین بار مورد نیاز باشد محاسبه می‌شود و سپس فقط تا حدی که مورد نیاز باشد. ارزیابی ضعیف، پذیرش تعاریف حلقه‌ای توابع را ممکن می‌سازد که در غیر این صورت به محاسبات بی‌پایان و نامحدودی می‌انجامیدند. به علاوه، ارزیابی ضعیف روشی جهت پیمانه‌ای نمودن برنامه‌های عملیاتی محسوب می‌شود.

منظور از توابع بالا رتبه، توابعی هستند که پارامترهایشان خود تعریف یک تابع دیگر است.

بیان مثالهایی برای نمایش قدرت ارزیابی ضعیف، به محاسبات روی لیستها بستگی دارد، متأسفانه در مورد این جنبه جالب نیز نمی‌توانیم بیش از این پیش برویم. ایده ارزیابی ضعیف با وجود این که قدرت زیادی در نظریه زبانها دارد ولی به‌تازگی مورد توجه قرار گرفته است.

high order functions	۳۷
lazy evaluation	۳۸
call by need	۳۹
call by value	۴۰
eager evaluation	۴۱
call by name	۴۲

- تمایز بر حسب مورد: نشان‌دهندهای مختلف می‌توانند سبب تعاریف متفاوت گردند.
- بارگشتی.
- ترکیب: نشانوند یک تابع خود می‌تواند یک تابع دیگر باشد.

در اینجا یک سؤال طرح می‌شود: آیا تمام الگوریتم‌ها را می‌توان به شکل توابع ریاضی بیان کرد؟ پاسخ این سؤال مثبت است. بدین دلیل که حلقه‌ها را می‌توان با استفاده از توابع بارگشتی پیاده‌سازی کرد و انشعابهای شرطی را نیز می‌توان توسط تعاریف تابع با حالتها و موارد مختلف نشان داد. برای مثال، الگوریتم زیر را که برای جمع اعداد طبیعی از 1 تا n، در یک زبان فرمانی فرضی به شکل زیر نشان داده می‌شود:

```
Sum(n);
  local i,s;
  begin
    s:=0;
    for i:=1 to n do s:=s+i end;
  return s
end Sum
```

می‌تواند به کمک تابع ریاضی زیر بیان شود:

$$\text{Sum}(0) = 0$$

زبانهای عملیاتی به‌طور گسترده از تعریف توابع تودرتو و توابع کمکی استفاده می‌کنند. برای مثال تابع زیر را در نظر بگیرید:

$$f(x) = a \times (x - 1)^2 + b \times (x - 1) + c$$

این تابع با کمی جابه‌جایی روی اجزای آن به شکل زیر تبدیل می‌شود:

$$f(x, a, b, c) = (a \times (x - 1) + b) \times (x - 1) + c$$

با یک استدلال ساده محاسباتی می‌توان نتیجه گرفت که این شکل نیز کارائی چندان ندارد زیرا عبارت $(x - 1)$ در آن دو بار محاسبه می‌شود. جهت بهبود بخشیدن به آن می‌توان از تابع کمکی $g(v, a, b, c)$ با تعریف زیر استفاده کرد:

با این تغییر، عبارت $(x - 1)$ تنها یکبار محاسبه می‌شود. بر اساس نظریه محاسبه‌پذیری^{۴۳} زبانهای عملیاتی به‌اندازه زبانهای فرمانی قدرت محاسباتی دارند. (یعنی کلیه الگوریتمها به‌وسیله هر دو قابل تعریف

در زبان لیسپ از فراخوانی با مقدار استفاده می شود ولی در زبانهای عملیاتی جدید نظیر میراندا و هوپ [3] از فراخوانی برحسب نیاز استفاده می شود.

مدل تفکر برنامه سازی عملیاتی در کتاب «مقدمه ای بر برنامه سازی عملیاتی» [3] نوشته بیرد و والدز چنین توصیف شده است:

برنامه سازی در یک زبان عملیاتی عبارتست از: بنای نهادن تعاریف و استفاده از کامپیوتر برای ارزشیابی عبارات. نقش اولیه برنامه ساز، ایجاد یک تابع برای حل یک مسئله مفروض است. این تابع که می تواند مشتمل بر تعدادی تابع کمکی نیز باشد به شیوه نگاشت طبیعی ریاضیات نشان داده می شود. کامپیوتر را می توان ماشینی در نظر گرفت که عبارات را دریافت می کند و حاصل محاسبه را باز می گرداند. در نظر گرفتن این نقش برای کامپیوتر، آن را به یک ماشین حساب جیبی شبیه می کند. تنها عنصری که ماشین حساب عملیاتی را از یک ابزار محدود، به یک ماشین محاسبه پرقابلیت بدل می سازد توان برنامه ساز جهت ارائه تعاریف به منظور افزایش توان محاسباتی ماشین است. عباراتی که حاوی نامهای توابع تعریف شده توسط برنامه ساز هستند، با استفاده از تعاریف مفروض به قوانین عنوان خلاصه سازی (یا کوتاه سازی) برای تبدیل عبارات به شکل قابل چاپ، محاسبه و ارزیابی می شوند.

یکی از مهمترین مشخصات زبانهای عملیاتی این است که اگر یک عبارت مقدار تعریف شده ای داشته باشد، ترتیب محاسبه عبارت توسط کامپیوتر روی نتیجه تاثیر نخواهد گذاشت به بیان دیگر، مفهوم یک عبارت، مقدار آن است و وظیفه کامپیوتر نیز به طور ساده به دست آوردن این مقدار است. در نتیجه، محاسباتی در این زبانها نظیر هر عبارت ریاضی دیگر، با استفاده از قوانین کم و بیش شبیه قوانین جبری، قابل ساختن، دستکاری و اثبات هستند. بدین ترتیب، یک چهار چوب مفهومی برای برنامه سازی ایجاد می شود که بسیار ساده، دقیق، انعطاف پذیر و قدرتمند است.

مفهوم یک عبارت، مقدار آن است و
وظیفه کامپیوتر نیز به طور ساده به دست
آوردن این مقدار است.

مقدار یک عبارت تنها بستگی به زیرعبارتهای تشکیل دهنده اش دارد و این زیرعبارات نیز آزادانه می توانند با عناصر دیگری که همان مقدار را دارند جایگزین شوند. یک عبارت می تواند حاوی «نامهای» معینی باشد که به جای مقادیر و کمیتهای ناشناخته قرار می گیرند. این نامها معمولاً «متغیر» خوانده می شوند، هر چند برداشت تمام ریاضیدانها

این است که متغیرها تغییر نمی کنند و اغلب مقدار ثابتی را مشخص می سازند. انواع مقداری که یک عبارت می تواند اختیار کند عبارتند از: اعداد، مقادیر منطقی، حروف، چند تائیه، توابع و لیستها.

این مدل تفکر ریاضیدانها است و با شیوه تفکر یک دانشمند کامپیوتر متفاوت است. آنچه که در نظر اول به عنوان ارزش برنامه سازی عملیاتی طرح می شود با کمی دقت نقطه ضعف آن نیز محسوب می شود زیرا در این شیوه تفکر، ایده «حالت» و در نظر گرفتن محاسبه به عنوان نوعی شبیه سازی نادیده گرفته می شود. البته این بدین معنی نیست که برنامه سازی عملیاتی به طور کامل قادر به نمایش حالت و شبیه سازی محاسبات نیست (این کارها به کندی قابل انجام است) ولی می توان خصوصیات مانند غیر طبیعی بودن از نظر برنامه سازی و پیچیدگی بی جهت را به عنوان عوامل منفی آن ذکر کرد. در نتیجه، زمینه به کارگیری زبانهای عملیاتی محض، محدود به مسائلی می شود که در آنها باید مقادیر نامعلوم را روی مقادیر معلوم محاسبه کرد. در چنین زمینه هایی برنامه های عملیاتی بسیار کوتاهتر، مطمئنتر، مجردتر از برنامه های فرمانی هستند. برای گسترش حیطه کاربردی چنین زبانهایی می توان برخی از خصوصیات زبانهای فرمانی را نیز به آنها افزود، همچنان که در زبان لیسپ انجام شده است. اما در اصل چنین کاری می تواند منبع اشتباهات جدید باشد، به کارگیری خصوصیات یک زبان عملیاتی و فرمانی به طور همزمان در بسیاری موارد کارایی تولید نرم افزار را کاهش می دهد.

ترجمه یک کتاب

در این شماره نتوانستیم فصل یازدهم کتاب نفرماه افسانه ای را برای چاپ آماده کنیم. امیدواریم از شماره بعد انتشار ترجمه این کتاب را از سر بگیریم.

منتشر شد

آشنایی با کامپیوتر و داده پردازی

تألیف لطفعلی بخشی

چاپ هفتم

از انتشارات جهاد دانشگاهی

مقاله

آرایه‌های تپنده

نوشتهٔ سابریت سینگ و جیا-یوان هان*
ترجمه و تلخیص از نادر صالحی

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

این عمل کاربردهای بسیاری در مسایل مهندسی از قبیل فیلترهای با پاسخ ضربهٔ محدود^۴، تبدیل فوریهٔ گسسته^۵ و تلفیق گسسته^۶ دارد. در زیر نمونه‌ای از یک برنامهٔ فرتن که عملیات فوق را انجام می‌دهد، دیده می‌شود.

```
DO 15 I = 1,N
Y(I) = 0
DO 10 J = 1,N
10 Y(I) = Y(I) + A(I,J) * X(J)
15 CONTINUE
```

در این برنامه Y(I) بردار نهایی به طریق زیر محاسبه می‌شود:

$$\begin{aligned} Y(1) &= A(1,1) * X(1) + A(1,2) * X(2) + \cdots + A(1,N) * X(N) \\ Y(2) &= A(2,1) * X(1) + A(2,2) * X(2) + \cdots + A(2,N) * X(N) \\ Y(n) &= A(N,1) * X(1) + A(N,2) * X(2) + \cdots + A(N,N) * X(N) \end{aligned}$$

معادله (۱)

در صورتیکه کامپیوتری دارای N پردازشگر باشد و این پردازشگرها قابلیت اجرای همزمان دستورات را داشته باشند، این امکان وجود دارد که کلیهٔ Y(I)ها در یک زمان محاسبه شوند. به این دلیل که هیچ یک از Y(I)ها به Y(I) دیگری بستگی ندارد. کامپیوتری که قادر به استفاده از خاصیت همزمانی در سایر محاسبات علمی (از جمله تبدیل

آرایه‌های تپنده انقلابی در ساخت کامپیوترهای موازی بسیار سریع به شمار می‌روند. سرعت محاسباتی بالا این روزها نیازی است که در همه جا به چشم می‌خورد. بسیاری از مسائل علمی، مهندسی و مدل‌سازی سیستم دارای محاسبات بسیار زیادی هستند. این قبیل مسائل اغلب در طبقه‌بندی مسایلی با تنگنای محاسباتی^۱ جای می‌گیرند زیرا از واحد پردازشگر مرکزی^۲ بیشتر از دستگاههای ورودی و خروجی کامپیوتر استفاده می‌کنند. شاید به سالها زمان نیاز باشد تا بتوان چنین مسایلی را توسط کامپیوترهای شخصی، حل کرد.

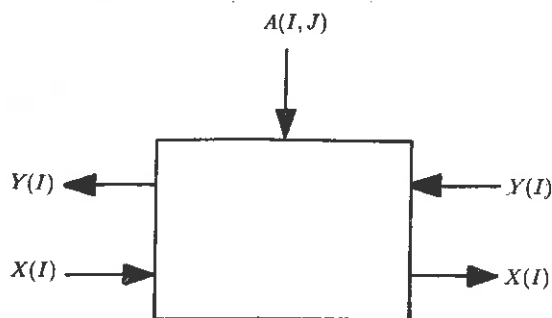
به‌طور مثال پیش‌بینی وضع هوا در ۲۴ ساعت آینده برای یک منطقهٔ جغرافیایی بزرگ احتمالاً به اجرای ۱۶۰۰ میلیارد دستور بر روی یک کامپیوتر نیاز دارد. فرض کنید برای این منظور کامپیوتری با سرعتی معادل ۱۰۰ برابر یک کامپیوتر شخصی موجود است. این کامپیوتر قادر به انجام صد میلیون دستور اعشاری در ثانیه^۳ است. در نتیجه زمان محاسبه چیزی در حدود ۲۴ ساعت بوده و پیش‌بینی وضع هوا در زمان آماده شدن نتایج، بی‌فایده خواهد بود. برای حاضر شدن نتایج در یک زمان معقول به کامپیوتری ۱۰ برابر سریعتر از کامپیوتر قبلی نیاز است. در صورتیکه به پیش‌بینی پیچیده‌تر و دقیقتری احتیاج داشته باشیم باز هم به کامپیوتر سریعتری نیاز داریم. بنابر این ابرکامپیوترها برای پاسخگویی به این نیاز فزاینده یعنی داشتن یک سیستم محاسباتی سریع به‌وجود آمدند. لازم به گفتن نیست که این قبیل کامپیوترها بسیار گران هستند. زیرا در این سیستمها هزینه فدای سرعت شده است. تمامی ابرکامپیوترها برای رسیدن به سرعت پردازش بالا به نحوی از پردازش موازی استفاده می‌کنند. توضیح کامل پردازش موازی در محدودهٔ این مقاله نمی‌گنجد و تنها به ذکر مثالی که بیانگر خاصیت پردازش موازی باشد بسنده می‌کنیم.

مسئلهٔ ضرب یک ماتریس $n \times n$ مانند $A = (a_{ij})$ را با یک بردار n تایی $X = (x_1, x_2, \dots, x_n)^T$ در نظر بگیرید.

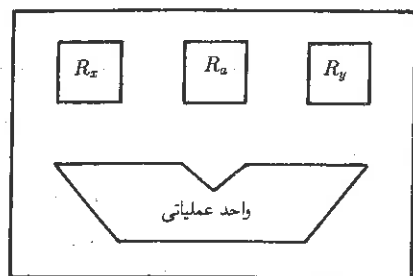
Supreet Singh and Jia-Yuan Han, "Systolic Arrays",
IEEE Potentials, February 1991.

compute bound problems	۱
Central Processing Unit-CPU	۲
million floating point instructions per second-mega	۳
flops	

Finite Impulse Response(FIR) filter ۲
Discrete Fourier Transform-DFT ۵
discrete convolution ۶



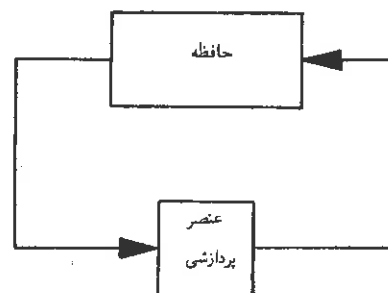
شکل ۲- الف عنصر پردازشی آرایه تپنده



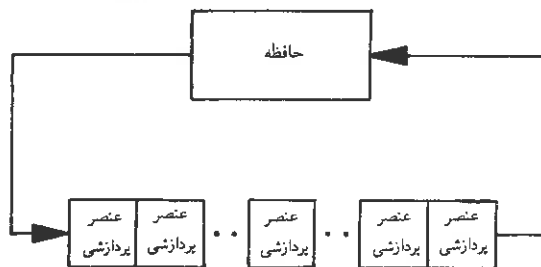
شکل ۲- ب طرز قرار گرفتن ثابتها در عنصر پردازشی

شده و سپس نتیجه برای سلول بعدی فرستاده می‌شود. هر سلول تنها با سلولهای مجاور خود در تماس است بنابراین بین سلولها ارتباط محلی برقرار است. همچنین، بر روی هر قسمت از داده‌ها عملیات محاسباتی انجام شده و ارتباط با خارج تنها در سلولهای فردی انجام می‌گیرد. بنابراین آرایه فوق دارای تمامی شرایط جهت طراحی مدارات مجتمع در مقیاس بسیار بزرگ است. برای آشنایی بیشتر با نحوه عمل آرایه تپنده از مثال ضرب ماتریسی که قبلاً اشاره شده است استفاده می‌کنیم.

شکل ۱- الف نحوه ارتباط حافظه را با یک واحد محاسباتی یا یک کامپیوتر نشان می‌دهد. در صورتی که این واحد عهده‌دار انجام یک عمل ضرب ماتریس باشد، کلیه $X(I)$ ها باید به دفعات از حافظه خوانده شوند (با فرض اینکه خود واحد محاسباتی دارای حافظه محدودی است). از طرف دیگر در طرح یک آرایه تپنده، $X(I)$ تنها یک بار از حافظه خوانده می‌شود و بارها توسط آرایه‌ای از پردازشگرها مورد استفاده قرار می‌گیرد (شکل ۱- ب). شکل ۲- الف نمای یک سلول پردازشگر را که در طراحی آرایه تپنده‌ای که به منظور انجام عمل ضرب فوق مورد استفاده قرار می‌گیرد نشان می‌دهد. شکل ۲- ب ثابتهای^{۱۱} این سلول را نشان می‌دهد. این سلول پردازشگر ساده، تنها دارای سه ثابت R_x ، R_y ، R_a و یک واحد عملیاتی^{۱۲} است. R_x و R_y به ترتیب دارای مقادیر $X(I)$ ، $A(I, J)$ و $Y(I)$ هستند. واحد عملیاتی مقدار فعلی R_y را با حاصل ضرب R_x و R_a جمع کرده، نتیجه را دوباره در R_y قرار می‌دهد. این عمل بوسیله دستور زیر نشان داده می‌شود.



شکل ۱- الف نحوه ارتباط پردازنده با حافظه اصلی در کامپیوترهای متداول



شکل ۱- ب کامپیوتر مبتنی بر آرایه تپنده

فوریه سریع^۶ یا محاسبه تبدیل فوریه گسسته یک رشته جهت افزایش سرعت محاسباتی می‌توان استفاده کرد. پردازش موازی مورد بررسی ما در سطح انجام دستورات ماشین است. بدیهی است که پردازش موازی در سطوح بالاتر (مثلاً سطح برنامه که در آن دو برنامه به طور همزمان اجرا می‌شوند) و نیز در سطوح پایینتر (مثلاً اجرای همزمان قسمتهایی از یک دستور اسمبلی) امکان‌پذیر است.

آرایه‌های تپنده

آرایه‌های تپنده برای اولین بار در سال ۱۹۷۸ توسط کانگ^۸ و دستیاران وی در دانشگاه کارنگی-ملون^۹ جهت محاسبات ماتریس پیشنهاد شد. از آن زمان تاکنون این آرایه در بسیاری از الگوریتمهای پیچیده مورد استفاده قرار گرفته است.

آرایه تپنده مجموعه‌ای از عناصر پردازشی^{۱۰} (یا سلول) است که هر یک از این سلولها توانایی انجام دستور ساده‌ای را دارند. این سلولها بسته به نوع فعالیت مورد نظر به یکدیگر وصل شده، تشکیل یک شبکه می‌دهند. آرایه تپنده معمولاً از طریق سلولهای فردی با دنیای خارج در ارتباط است. ساختار این آرایه به دلیل شباهت عمل این آرایه با نحوه تلمبه شدن خون در بدن تپنده نام گرفته است. خون به وسیله انقباض قلب و شراین در بدن تلمبه می‌شود. در یک آرایه تپنده نیز داده‌ها به طور منظم از یک سلول به درون سلول دیگر فرستاده می‌شوند. در داخل هر سلول تغییرات مورد نظر بر روی داده انجام

Fast Fourier Transform-FFT^۷H. T. Kung^۸Carnegie-Mellon^۹Processing Element-PE^{۱۰}register^{۱۱}
functional unit^{۱۲}

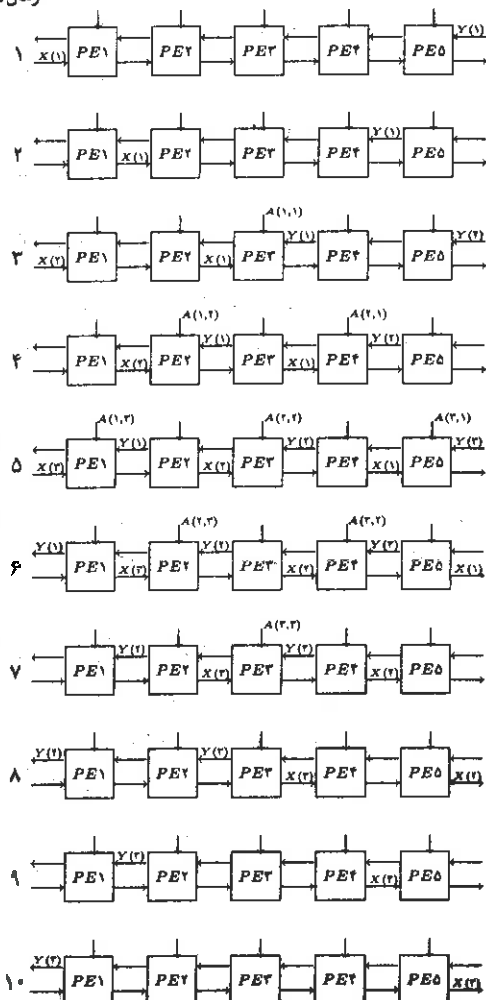
سوم و $X(3)$ و $Y(3)$ در پالس پنجم.

- قرار گرفتن $A(I, J)$ و $X(I)$ و $Y(I)$ بر روی ورودیهای هر سلول به معنای قرار گرفتن این مقادیر در ثبتهای R_x و R_a و R_y است.
- با هر پالس زمان سنج $X(I)$ بطرف راست و $Y(I)$ به طرف چپ حرکت می‌کند.

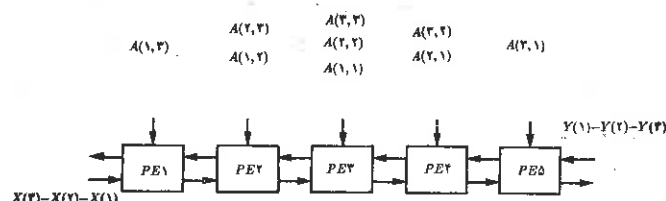
برای محاسبه این حاصل ضرب، مراحل زیر انجام می‌شوند:

- ۱- $X(1)$ وارد سلول ۱ و $Y(1)$ وارد سلول ۵ می‌شود.
- ۲- $X(1)$ و $Y(1)$ بدون تغییر وارد سلولهای ۲ و ۴ می‌شوند (در سلول ۵ عمل انجام شده $0 * 0 = 0$ است).
- ۳- $A(1, 1)$ و $Y(1)$ و $X(1)$ وارد سلول ۳ می‌شوند. در پایان این مرحله $Y(1) = A(1, 1) * X(1)$.
- ۴- $A(1, 2)$ و $X(2)$ و مقدار جدید $Y(1)$ وارد سلول ۲ می‌شوند. در همین زمان $A(1, 2)$ ، $X(2)$ و $Y(2)$ به سلول ۴ می‌رسند.

زمان سنج



شکل ۴- ترتیب مراحل ضرب ماتریس در بردار



شکل ۳- پیکربندی آرایه تپنده برای ضرب ماتریسی

$$R_y := R_y + R_x * R_a$$

به همراه هر پالس زمان سنج، مقدار جدید $Y(I)$ در خروجی سلول ظاهر شده و جهت جمع با حاصل ضرب $A(I, J) * X(I)$ داخل ثبت R_y سلول بعدی می‌شود. در تمامی مراحل محاسبه مقدار $A(I, J)$ و $X(I)$ تغییر نمی‌کنند. مقادیر $A(I, J)$ تنها یک بار مورد استفاده قرار می‌گیرند (به معادله ۱ توجه کنید) و بنابراین پس از استفاده کنار گذاشته می‌شوند. تمامی $X(I)$ ها در داخل ثبت R_x سلولها قرار می‌گیرند. محتویات ثبتهای R_x و R_y با هر پالس زمان سنج به سلولهای مجاور منتقل می‌شوند. مقدار Y از داخل چندین سلول عبور کرده و در هر سلول حاصل ضرب $X(I) * A(I, J)$ به آن اضافه شده و در نهایت به صورت معادله زیر در می‌آید:

$$Y(I) = A(I, 1) * X(1) + A(I, 2) * X(2) + \dots + A(I, N) * X(N)$$

باید این نکته را متذکر شد که X و Y در داخل سلولها خلاف جهت یکدیگر حرکت می‌کنند.

عملکرد آرایه تپنده

شکل ۳ پیکربندی^{۱۳} یک آرایه تپنده را برای محاسبه ضرب ماتریسی در حالت $N = 3$ نشان می‌دهد. در این شکل ورودیهای سیستم در زمانهای مختلف به صورت زنجیر نشان داده شده‌اند. در ابتدا $X(1)$ و $Y(1)$ همزمان با هم و با یک پالس زمان سنج به ترتیب وارد سلولهای ۱ و ۵ می‌شوند. به همین صورت $A(1, 2)$ و $A(2, 1)$ در پالس چهارم وارد سلولهای ۲ و ۴ می‌شوند (۱-۱ به معنی صفر است). در شکل ۴ اعمالی را که جهت به دست آوردن نتیجه ضرب ماتریسی لازم است مشاهده می‌کنید. کلاً ده پالس زمان سنج مورد نیاز است تا $Y(3)$ (آخرین جواب) آماده شود. قبل از ذکر یک مثال و جهت درک بیشتر آن نکات زیر را در نظر بگیرید:

- در ابتدا ثبتهای R_x ، R_a ، R_y تمامی سلولها و $Y(1)$ ، $Y(2)$ و $Y(3)$ صفرند.
- در زمانهای فرد $X(I)$ و $Y(I)$ داخل آرایه تپنده می‌شوند یعنی $X(1)$ و $Y(1)$ در پالس اول، $X(2)$ و $Y(2)$ در پالس

۲- نداشت^{۱۵} الگوریتم بر روی آرایه. دیدیم که الگوریتمهای مختلف جهت افزایش کارایی احتمالاً به آرایه‌های تپنده‌ای با ساختارهای متفاوت نیاز دارند. انتخاب و طراحی یک آرایه تپنده با ساختار مناسب نیاز به یک طراح با تجربه و با اطلاعات وسیع دارد. در حال حاضر تلاشهایی در دست اقدام است تا این عمل توسط نرم‌افزار انجام گیرد. این کار منجر به کاهش زمان طراحی و در نتیجه کاهش هزینه خواهد شد. محققان تاکنون انواع پیکربندیها را برای آرایه تپنده مشخص کرده‌اند (شکل ۵).

با وجود سهولت پیاده‌سازی مدارهای مجتمع در مقیاس بسیار بزرگ، آرایه‌های تپنده دارای نقایصی نیز هستند که نیاز به بررسی بیشتری دارند.

۱- فقدان برنامه‌پذیری. در تمام این مقاله آرایه تپنده به‌عنوان پردازشگرهایی با مقاصد خاص^{۱۶} در نظر گرفته شده‌اند با این وجود در بعضی موارد اجرای چند الگوریتم مختلف (جهت حل یک یا چند مسئله) بر روی یک آرایه اقتصادی‌تر است. بنابراین فقدان برنامه‌پذیری یک نقص برای آرایه تپنده محسوب می‌شود. یک راه حل، استفاده از آرایه‌های تپنده با قابلیت پیکربندی مجدد^{۱۷} است. به‌طور مثال ماتریسی از پردازشگرها را می‌توان توسط یک سری کلید قابل کنترل به یکدیگر متصل کرد. این کلیدها را می‌توان به‌گونه‌ای قطع و وصل کرد تا پیکربندی مورد نیاز (درخت دوشاخه‌ای^{۱۸})، شبکه‌های شش ضلعی^{۱۹} یا آرایه خطی) به‌دست آید. روش دیگر استفاده از یک آرایه تپنده با پیکربندی ثابت به همراه ابزاری نرم‌افزاری جهت تبدیل الگوریتمهای مختلف به الگوریتم مناسب این آرایه است. این روش از جهاتی شبیه به روش کامپیوترهای موازی همه منظوره^{۲۰} است. چیزی که این کامپیوترها را در رده پردازشگرهای تپنده قرار می‌دهد عمل انتقال تلمبه مانند داده‌ها از یک پردازشگر به دیگری است.

در پایان این مرحله

$$Y(1) = A(1, 1) * X(1) + A(1, 2) * X(2)$$

$$Y(2) = A(2, 1) * X(1) +$$

۵- $A(1, 3), A(2, 2), A(3, 1)$ به ترتیب وارد سلولهای ۱، ۲، ۳ و ۵ می‌شوند. در پایان این مرحله

$$Y(1) = A(1, 1) * X(1) + A(1, 2) * X(2) +$$

$$A(1, 3) * X(3)$$

$$Y(2) = A(2, 1) * X(1) + A(2, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1)$$

۶- در این مرحله $A(2, 3)$ و $A(3, 2)$ به ترتیب وارد سلولهای ۲ و ۴ می‌شوند. در پایان این مرحله

$$Y(2) = A(2, 1) * X(1) + A(2, 2) * X(2) +$$

$$A(2, 3) * X(3)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

$$Y(3) = A(3, 1) * X(1) + A(3, 2) * X(2)$$

نتایج عملی

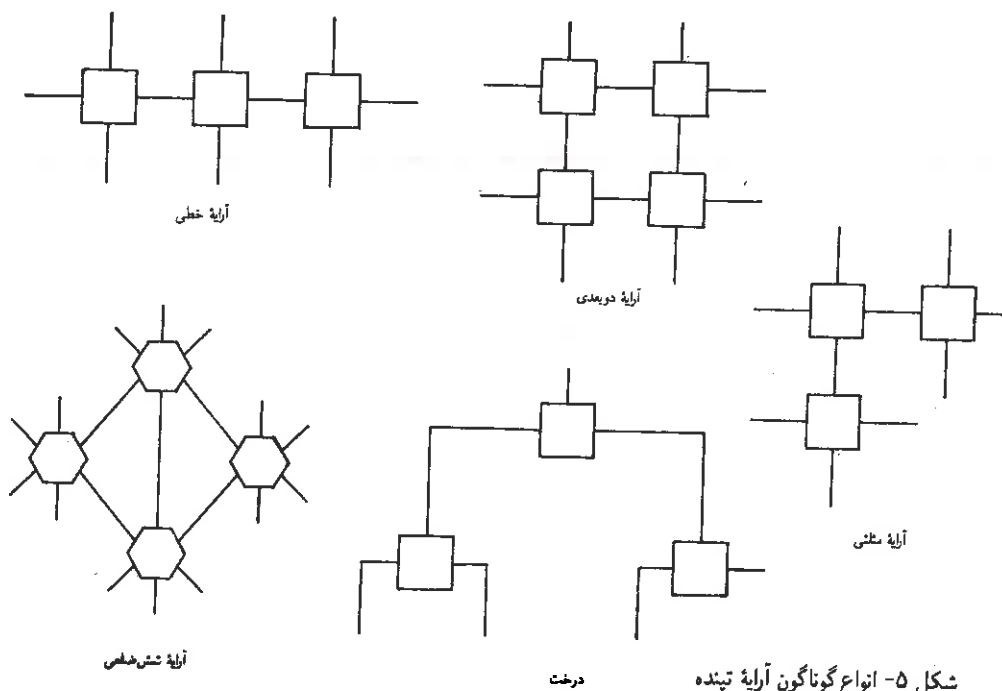
در صورتی که بخواهیم از آرایه‌های تپنده برای حل مسائل عملی استفاده کنیم با مشکلات پیاده‌سازی مهمی مواجه می‌شویم که درباره مهم‌ترین آنها بحث می‌شود.

۱- حل مسائل بزرگتر از طول آرایه. دیدیم که اندازه آرایه تپنده وابسته به اندازه آرایه مسئله ماست. مسلماً آرایه‌ای که برای ضرب دو ماتریس 10×10 در نظر گرفته شده است، مستقیماً برای ضرب ماتریس 50×50 قابل استفاده نیست. البته ساده‌ترین راه حل ساخت یک آرایه بزرگتر است که چنین امری به دلیل مشکلات تکنولوژیکی همیشه امکان‌پذیر نیست. راه بهتر شکستن^{۱۴} مسائل بزرگ به مسائل کوچک است به‌گونه‌ای که مناسب با آرایه موجود باشد. در حال حاضر تلاشهایی جهت تهیه ابزارهای نرم‌افزاری که این عمل را برای برنامه‌نویس انجام دهد در دست اقدام است.

برای ایجاد سیستمی با قابلیت اطمینان بالا باید از مکانیزمهای سخت‌افزاری بسیار پیچیده‌ای استفاده کرد.

۲- تأخیر زمان‌سنج. آرایه‌های تپنده توسط یک زمان‌سنج همگام شده‌اند. در آرایه‌های بسیار بزرگ سیمهای حامل پالس زمان‌سنج به نقاط مختلف دارای طولهای متفاوتند که این باعث ایجاد تاخیرهای متفاوتی بر روی سیگنال زمان‌سنج در نقاط مختلف می‌شود. این

mapping	۱۵
special purpose	۱۶
reconfigurable	۱۷
binary tree	۱۸
hexagonal mesh	۱۹
general purpose	۲۰



شکل ۵- انواع گوناگون آرایه تپنده

سیلیکون (که مدارهایی مجتمع از آن ساخته می شود) قرار گرفته و به همدیگر متصل می شوند. چنین روشی به سرعت عملیاتی بیشتر و توان مصرفی پایین منجر می شود. این تکنیک مجتمع سازی در مقیاس ویفر^{۲۶} نام داشته و در طراحی کامپیوترهای موازی و سایر سیستمهای با چگالی قطعات بالا کاربرد وسیعی دارد.

سیستمهای موجود

در دانشگاهها و آزمایشگاهها، تعداد زیادی از سیستمهای همه منظوره و با الگوریتم ویژه^{۲۷} که اساس ساختمانشان تپنده است، ساخته شده اند. دو نمونه از سیستمهای تپنده همه منظوره، ماتریس-۲۸۱ از شرکت ساکس پای^{۲۹} و وارپ^{۳۰} از دانشگاه کارنگی ملون هستند. گزارشات حاکی از آن است که ماتریس-۱ دارای حداکثر سرعتی برابر ۱۰۰۰ میلیون دستوراعشاری در ثانیه است.

نتیجه گیری

آرایه های تپنده، ساختارهایی جهت پردازش موازی بوده و جهت پیاده سازی بر روی تراشه ها بسیار مناسبند. این آرایه ها هنگام انجام محاسبات زمان گیر در سخت افزار و نیز جهت ساخت کامپیوترهای سریع همه منظوره به کار می روند.

پدیده تأخیر زمان سنج^{۲۱} نام دارد. یکی از علل این پدیده، خاصیت خازنی خطوط حامل پالس زمان سنج است که با افزایش طول خط تغییر می کند. از آنجائی که آرایه تپنده بدون همگام سازی داخلی قادر به کار نیست، این امر می تواند باعث بروز مشکلات جدی در کار این آرایه ها شود. یک راه حل پیشنهادی استفاده از آرایه های تپنده موجی^{۲۲} است. در چنین آرایه ای زمان سنج داخلی وجود نداشته و در عوض تمامی محاسبات مبتنی بر داده ها^{۲۳} است. یعنی پردازنده تنها در صورت آماده شدن تمامی داده ها دستورات را اجرا می کند.

۳- قابلیت اعتماد. تراشه هایی که تعداد زیادی پردازشگر دارند از اطمینان کافی برخوردار نیستند. بنا به اشکالات تولید، خرابی قطعات، عوامل محیطی و عوامل دیگر این احتمال وجود دارد که پردازشگرها از کار افتاده یا به تناوب دچار اشکال شوند. برای ایجاد سیستمی با قابلیت اطمینان بالا باید از روشهای سخت افزاری بسیار پیچیده ای استفاده کرد. چنین طراحیهای مطمئن در بخش تشخیص و تحمل خرابیها^{۲۴} مورد مطالعه واقع می شوند.

۴- طراحی آرایه های تپنده بزرگ. تعداد پردازشگرهای موجود بر روی یک تراشه به دلیل محدودیتهای تولید مدارهای مجتمع دارای محدودیتند. اگر یک آرایه به فضایی بیشتر از حد معمول نیاز داشته باشد، در آن صورت چندین مدار مجتمع جهت تولید یک آرایه بزرگتر روی مدار چاپی قرار گرفته به همدیگر متصل می شوند. روش دیگری نیز وجود دارد. در این روش چندین تراشه روی یک لایه نازک^{۲۵}

wafer scale integration	۲۶
algorithm specific	۲۷
Matrix-I	۲۸
Saxpy	۲۹
Warp	۳۰

clock skew	۲۱
wavefront	۲۲
data driven	۲۳
fault tolerant	۲۴
wafer	۲۵

مقاله

فوتونیک

نوشته آنتونی دوماریا*
ترجمه و تلخیص از حسن کاتوزیان

نقش والدین، و شاخه‌های فوتونیک فرزندان آن درخت هستند (شکل ۲). رشد فوتونیک تازه شروع شده است و بلوغ آن در نیمه دوم قرن بیست و یکم خواهد بود.

تاریخچه الکترونیک

فریدیناند براون^۵ به سال ۱۸۷۴ در آلمان موفق به کشف خاصیت یکسوکنندگی در اتصال بین فلز و نیمه هادی شد که همین اکتشاف منجر به ساخت دیودهای آشکارساز شد و نقشی عمده در امر تحقیقات مربوط به کاربرد امواج رادیویی در مخابرات ایفا کرد.

توماس ادیسون هنگام کار روی لامپهای رشته‌ای به سال ۱۸۸۳ متوجه شد، زمانی که صفحه با بار مثبت نزدیک رشته فیلامان گرم شده قرار دارد، درون لامپ خلأ جریان الکتریکی عبور می‌کند. این کشف تا زمان اختراع نخستین دیود لامپی توسط فلمینگ به سال ۱۹۰۴ که جریان متناوب الکتریکی را به جریان مستقیم تبدیل می‌کرد بلااستفاده ماند. اختراع دیود لامپی با اختراع لامپ خلأ تریود به سال ۱۹۰۶ (یعنی نخستین تقویت‌کننده الکترونیک) توسط لی دو فودست^۶ دنبال شد.

نوسانگر الکترونیک آرمسترانگ نیز در ادامه، نسل اول دستگاههای مولد امواج الکترومغناطیس را پایه‌گذاری کرد و به دنبال آن صنعت الکترونیک سریعاً رو به توسعه نهاد و لامپ خلأ به عنوان قلب صنایع الکترونیک در ساخت رادیو، تلویزیون، رادار، مدارهای کنترل، مخابرات و پردازش اطلاعات درآمد و تا زمان اختراع ترانزیستور به سال ۱۹۴۸ حاکم بلامنابع در این عرصه بود. به سال ۱۹۴۸ جان باردین^۷، والتر براتین^۸ و ویلیام شاکلی^۹ از آزمایشگاه بل، اختراع ترانزیستور را اعلام کردند. ابعاد کوچک، بازدهی زیاد، تولید در مقیاس وسیع، هزینه ساخت کمتر، نیاز به ولتاژ کار پائینتر و ضریب اطمینان کاری بالاتر از مزایای عمده ترانزیستور در مقایسه با لامپهای خلأ بود که منجر به تسلط سریع آن در عرصه صنایع الکترونیک شد.

پیدایش علم الکترونیک در قرن بیستم زائیده عرصه الکترومکانیک در قرن نوزدهم بوده است و می‌توان گفت که زائیده «فوتونیک»^۱ در قرن بیست و یکم خواهد بود.

صنعت الکترونیک و رشد آن مطمئناً تا حد زیادی مدیون آرمسترانگ^۲ است که نخستین نوسانگر یا اسیلاتور الکترونیک را به سال ۱۹۱۲ ابداع کرد. اولین نوسانگر نوری (اپتیکی) یا همان لیزر هم توسط تد هیمن^۳ تحت عنوان لیزر یا قوت ۴۸ سال بعد اختراع شد. تقریباً ۸۰ سال طول کشید تا بازار الکترونیک به حد امروزی رشد و بلوغ برسد حال آنکه حدود ۳۰ سال از ساخته شدن نخستین لیزر می‌گذرد. اگر سرعت رشد تکنولوژیک در عرصه فوتونیک را مشابه الکترونیک فرض کنیم، می‌توان پیش‌بینی کرد که فوتونیک در اواسط قرن بیست و یکم به حد بلوغ و رشد کامل برسد (شکل ۱). البته این پیش‌بینی خوشبینانه است چرا که هزینه پائین ساخت دستگاههای الکترونیک، این تکنولوژی را بر دنیای کنونی حاکم نموده است. نتیجتاً زمانی فوتونیک را می‌توان با موفقیت مورد استفاده قرار داد که اولاً از آن در مواردی استفاده کنیم که الکترونیک قادر به رقابت با آن نباشد و ثانیاً در کاربردهایی از آن بهره گیریم که به سادگی قابل ارتباط یافتن با دستگاههای الکترونیک شود.

سه عرصه الکترومکانیک، الکترونیک و فوتونیک، ریشه در معادلات معروف ماکسول دارند که توسط جیمز کلرک ماکسول^۴ در سال ۱۸۶۴ گردآوری و ارائه شده‌اند. صناعی که براساس قوانین الکترومغناطیس و الکترومکانیک کار می‌کنند، از قبیل موتورهای، ژنراتورها، مغناطیس دائم و خطوط انتقال نیرو؛ در اواخر قرن نوزدهم رشدی عمده داشتند و در نیمه نخست قرن بیستم به تکامل رسیدند. سه ربع آخر قرن بیستم شاهد رشد سریع حوزه الکترونیک بوده است. در شجره‌نامه تکنولوژیهای استوار بر معادلات ماکسول، تکنولوژیهایی که برپایه الکتریسته و مغناطیس پایه‌ریزی شده‌اند، نقش اجداد را به عهده دارند. تکنولوژیهای مبتنی بر نرم‌افزار و سخت‌افزار الکترونیک، در

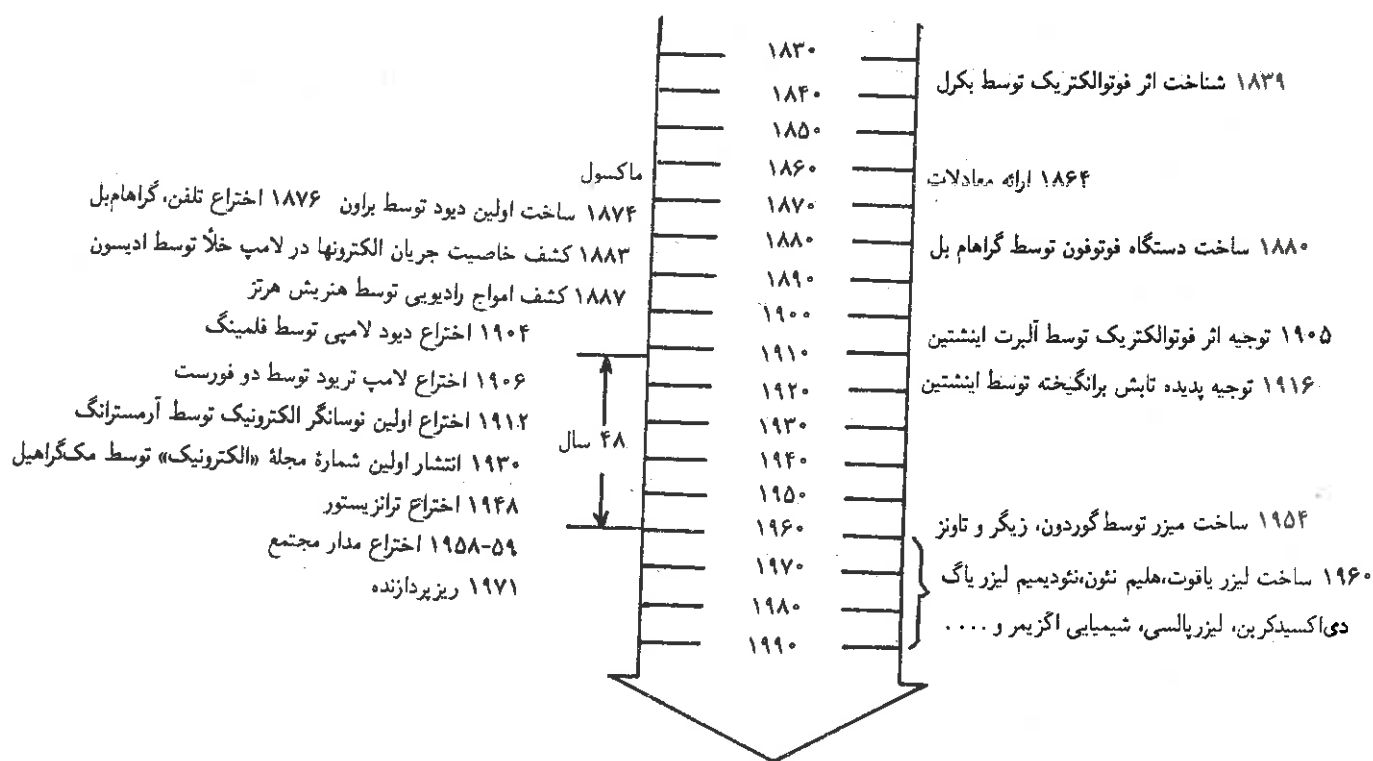
* Anthony J. DeMaria, "Maxwell's Children Lighth the Way", *IEEE Circuits and Devices*, March 1991.

۵ Ferdinand Braun
۶ Lee DeForest
۷ John Bardeen
۸ Walter Brattain
۹ William Shockley

۱ Photonics
۲ E. H. Armstrong
۳ Maiman Ted
۴ James Clerk Maxwell

فوتونیک

الکترونیک



شکل ۱- رویدادهای مهم تاریخی در فوتونیک و الکترونیک

در سال برای هر فرد ساکن کره زمین هستند. بهای یک مدار مجتمع با یک میلیون ترانزیستور در حال حاضر معادل یک ترانزیستور منفرد در روزهای اول اختراع آن است. پیشرفت در ساخت مدارات مجتمع یعنی پیشرفت در ساخت کامپیوترهای مدرنتر که به نوبه خود ابزار مناسب و دقیقی را جهت طراحی مدارهای مجتمع متراکمتر و پیچیده تر در اختیار می گذارد. این روند مداوم در حال تکرار است. همین منجر به رشد سریع و انفجارگونه در عرصه الکترونیک شده است.

تاریخچه فوتونیک

الکساندر ادmond بکریل^{۱۲} به سال ۱۸۳۹ در فرانسه آشکار سازهای نوری را برای نخستین بار کشف کرد. یعنی نخستین ابزار ایتو-الکترونیک که ضمن آن با تابش نور به محل اتصال الکترولیت، ولتاژ در دو سر آن پدید آمد. این اثر در آن زمان شناخته نشد تا آن که آلبرت اینشتین آن را به سال ۱۹۰۵ تحت عنوان «اثر فوتوالکتریک»^{۱۳} توجیه کرد. هر

در سالهای ۱۹۵۸ و ۱۹۵۹ مدارهای مجتمع یا آی-سی توسط جک کیلبی^{۱۰} از شرکت تگزاس اینسترومنتز و داجرت فوئیس^{۱۱} از فیرچایلد اختراع شد. ساخت تراشه مدار مجتمع این امکان را داد تا کل یک سیستم الکترونیک را در ابعاد بسیار کوچکتر روی یک تکه سیلیسیم گرد هم آوریم و این نقطه آغاز انقلاب پردازش اطلاعات امروزی شد. توسعه ساخت ریزپردازنده ها از سال ۱۹۷۱ توسط شرکت اینتل آغاز شد و به تولد ماشین حسابهای جیبی و همینطور کامپیوترهای شخصی امروزی انجامید. اگر اختراع لامپ تریود به سال ۱۹۰۶ را آغاز انقلاب الکترونیک در نظر بگیریم، سن و سال الکترونیک را حتی باید بیش از ۸۰ سال دانست و شواهد و آمار نشان می دهد که روند کاهش هزینه ساخت و افزایش تراکم مدارات الکترونیک همچنان رشد سریع و مداومی را نشان می دهد.

صنعت نیمه هادیها طی ۳۰ ساله اخیر رشد نمایی شگفت انگیزی داشته است. از ارقامی معادل ۳۰ میلیارد دلار در این عرصه سخن گفته می شود و صنایع فوق قادر به تولید معادل ۲۵۰۰۰۰ ترانزیستور

^{۱۲} Alexandre Edmond Becquerel
^{۱۳} photoelectric effect

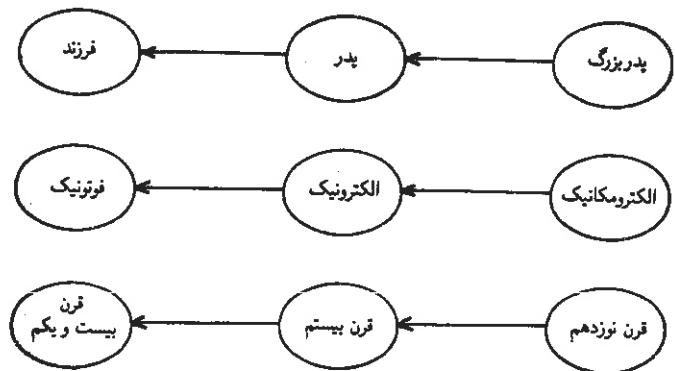
^{۱۰} Jack Kilby
^{۱۱} Robert Noyce

به سال ۱۹۵۴ شد. این دستگاه نخستین نوسانگر کوانتم الکترونیک به حساب می‌آید. لیزر یاقوت به سال ۱۹۶۰ توسط هچمن ساخته شد که نخستین نوسانگر امواج الکترومغناطیس در باند نوری بود. اختراع لیزر از همان ابتدا مورد توجه زیادی قرار گرفت و همین‌طور عامه را به هیجان آورد. اهل فن نیز بدین موضوع پی بردند که لیزر امکان و تکنولوژی لازم جهت انتقال صدا و تصویر در باند نوری و نه رادیویی را در اختیار می‌گذارد. در کمتر از ۲۰ سال لیزرهایی به شرح زیر به سرعت تکامل یافتند: لیزر هلیوم-نئون، آرگون، یون آرگون، دی‌اکسیدکربن، لیزر شیمیایی، اگزایمر، لیزر الکترون آزاد، لیزر یاک-۲۰ و همین‌طور انواع دیوهای لیزری نیمه هادی. همچنین پیشرفتهای وسیع در عرصه ساخت تقویت‌کننده‌های نوری به دست آمد. تولید پالسهای لیزری با عرض پالسی پیکوثانیه، استفاده از فیبرهای نوری، لیزرهای سوئیچ‌کننده کیو^{۲۱}، تمام‌نگاری^{۲۲} اپتیک غیرخطی، مدارهای مجتمع اپتیکی و... توسعه دستگاههای لیزری عامل پیشرفت عمده در عرصه فوتونیک بعد از دهه شصت میلادی شمرده می‌شود و درست همان نقشی را ایفا کرد که لامپهای خلأ در ابتدای عرصه الکترونیک بعد از سال ۱۹۰۶ ایفا کرد.

به دنبال اختراع لیزر واژه‌های جدیدی نظیر کوانتم الکترونیک، الکترواپتیک، لیدار^{۲۳} و پردازش نوری تصاویر در فرهنگ لغات فنی پیدا شد. اما خیلی زود معلوم شد که نیاز به یک واژه فراگیر است که همچون چتری همه این عناوین و اسامی را زیر سایه خود گرد هم آورد و دربرگیرد. واژه‌ای مشابه الکترونیک که شامل همه دستگاههای مربوط بدان می‌شد. از آنجائی که در همه این زمینه‌های تحقیقاتی از نور استفاده می‌شود و در واقع جریانی از فوتونها هستند، لذا واژه «فوتونیک» با افزودن پسوند «ایک» به انتهای کلمه فوتون شکل گرفت. فوتونیک یعنی تکنولوژی لازم برای تولید، آشکارسازی و هدایت امواج الکترومغناطیس در باند نوری در زمینه‌های مختلف کاربردی همچون تولید انرژی، مخابرات و پردازش اطلاعات.

رشد همگام الکترونیک و فوتونیک

در حال حاضر پیشرفت و توسعه در عرصه الکترونیک و فوتونیک را می‌توان به موازات هم مشاهده کرد. لیزرهای حالت جامد، لیزرهای گازی و سایر انواع لیزر همان نقشی را در فوتونیک ایفا می‌کنند که لامپ خلأ در دستگاههای الکترونیک انجام می‌داد و دارای خواص مشابه هستند. البته لیزر -مخصوصاً انواع لیزر گازی- نقاط ضعف لامپها یعنی وزن، حجم و مصرف زیاد را نیز دارد. لیزر نیمه هادی و ترانزیستور هم نقشی مشابه را در عرصه فوتونیک و الکترونیک بازی می‌کنند. حجم کم، قدرت مصرفی مورد نیاز اندک، قابلیت تولید در



شکل ۲- شجره نامه سه تکنولوژی اصلی که بر پایه معادلات ماکسول به پیش رفته‌اند.

چند علت این پدیده در قرن نوزدهم شناخته نشده بود ولی بارها توسط محققین مورد استفاده قرار گرفت. الکساندر گراهام بل به سال ۱۸۸۰ در دستگاه «فونوفون»^{۱۴} از اثر فتوالکتریک بهره گرفت. اینشتین فرمول‌بندیهای خود مربوط به فرآیندهای تابش خودبخود^{۱۵} و تابش برانگیخته^{۱۶} و ارتباط آن با فرآیند جذب نوری^{۱۷} در ساختمان اتمی و مولکولی را به سال ۱۹۱۶ ارائه کرد. پس از آن بسیاری از دانشمندان به هنگام مطالعه روی طیف گازها متوجه شدند که پدیده تجمع معکوس در سیستمهای مولکولی و اتمی و به عبارت دیگر افزایش نسبت تعداد اتمهای برانگیخته به تعداد غیر برانگیخته می‌تواند منجر به فرآیند تابش برانگیخته شود که همان منشأ نور لیزر است ولی در آن زمان این مهم نادیده گرفته شد.

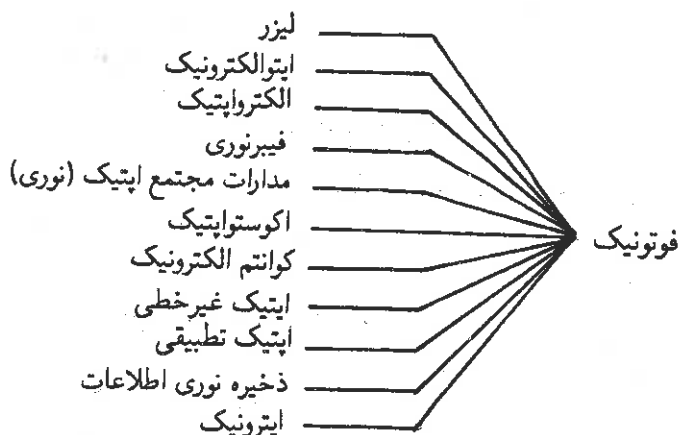
اما سؤال اینجاست که چرا دانشمندان در فاصله طولانی بین سالهای ۱۹۱۶ تا ۱۹۶۴ از خود این پرسش را نکردند که چطور می‌توان از پدیده تابش برانگیخته استفاده کرد و نخستین نوسانگر نوری را ساخت؟ محتمل‌ترین پاسخ آن است که محققانی که به بررسی خواص طیف گازها و بروز پدیده تجمع معکوس می‌پرداختند یا عموماً فیزیکدان بودند و یا در جریان تحقیقات الکترونیک توسط مهندسين قرار نداشتند. طی جنگ دوم جهانی بود که بین فیزیکدانان و مهندسين برق یک پل ارتباطی برقرار شد. اهمیت رادارهای میکروویو طی جنگ دوم باعث شد تا بسیاری از دانشمندان روی ساخت تقویت‌کننده‌ها و نوسانگرهای میکروویو به تحقیق پردازند. یکی از این متخصصین قانون^{۱۸} بود که تحقیقاتش در ساخت اینگونه نوسانگرها منجر به اختراع هیتر^{۱۹}

YAG	۲۰
Q-switching	۲۱
holography	۲۲
lidar	۲۳

photophone	۱۴
spontaneous emission	۱۵
stimulated emission	۱۶
radiation absorption	۱۷
C. H. Townes	۱۸
maser	۱۹

حاضر لیزرهای نیمه هادی عمدتاً در حوزه باند مادون قرمز و قرمز کار می‌کنند و می‌توان انتظار داشت که در آینده از موادی استفاده شود که امکان ساخت لیزر در سایر رنگهای طیف مرئی را فراهم کند.

سلاحهای لیزری، کنترل واکنش جوش^{۲۴} هسته‌ای در نیروگاههای اتمی به کمک لیزر و لیزرهای نیمه هادی سه زمینه عمده‌ای هستند که تکنولوژی لیزر طی ۲۵ سال اخیر در حال پیشرفت در آنهاست. دو مورد اول هنوز راهی طولانی در پیش دارند و به‌خاطر اهمیتی که در امور امنیتی و اقتصادی دارند فعلاً با حمایت دولتها، تحقیقات در آن زمینه‌ها در حال انجام است. اما لیزرهای نیمه هادی هم اینک نیز با سرمایه‌گذاریهای عمده از سوی واحدهای تحقیقاتی بخش خصوصی کشورها حمایت می‌شوند و این زمینه موفقترین حوزه کاربردی لیزر، تاکنون بوده است. سازگاری و مشارکت دستگاههای فوتونیک نیمه هادی با موادهای الکترونیک نیمه هادی منجر به رشد بسیار سریع این عرصه در امر مخابرات، پردازش سیگنالها و ذخیره اطلاعات، محاسبات کامپیوتری نوری و غیره شده که به‌عنوان زیرسیستمهایی در



شکل ۳- واژه فوتونیک نشانگر شمار وسیعی از تکنولوژیهای مرتبط در عرصه اپتیک و نور است.

fusion^{۲۴}

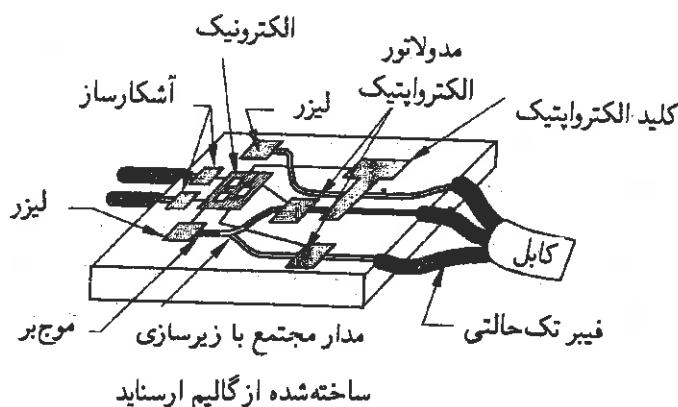
مقیاس وسیع و هزینه کمتر و ضریب اطمینان بالاتر و قابلیت ساختن شدن به‌صورت مدار مجتمع همه و همه از مزایای لیزرهای نیمه هادی به‌حساب می‌آیند. به‌علاوه به‌راحتی قابل ارتباط یافتن با ترانزیستورها هستند. نتیجتاً بازار فروش آنها به‌سرعت در حال رشد است.

در الکترونیک از موج برهایی چون کابلهای هم محور، خطوط مایکرواستریپ و غیره برای انتقال موج استفاده می‌شده حال آنکه در عرصه فوتونیک فیبرهای نوری نقش هدایت امواج نوری را با حداقل تلفات به‌عهده دارند که در مخابرات نوری مورد استفاده فراوانی یافته‌اند. تکنولوژی ساخت مدارات مجتمع اپتیکی امروزه مورد توجه بسیار قرار دارد و امید می‌رود که نقش مشابهی را در عرصه فوتونیک ایفا کند.

مواد نیمه هادی همچون گالیم آرسناید و آلومینیم گالیم آرسناید دارای خواص منحصر به فردی در مقایسه با نیمه هادیهای چون سیلیسیم و ژرمانیم هستند. چرا که قادر به تابش نور از خود هستند و به‌علاوه از نظر الکترواپتیکی و خواص پیزوالکتریک بسیار فعالند و می‌توان از آنها برای ساخت ترانزیستور نیز استفاده کرد. در نتیجه امکان ساخت یک‌جای مدارهای الکترونیک و مدارهای فوتونیک به‌صورت مجتمع وجود دارد.

میکروالکترونیک و فوتونیک

دیودهای نوری ساخته شده از گالیم آرسناید جهت استفاده در صفحات نمایش رقمی از اواخر دهه ۵۰ میلادی وارد بازار شد. در سال ۱۹۶۲ لیزر گالیم آرسناید که در حوزه نزدیک باند مادون قرمز کار می‌کرد توسط جنرال الکتریک و آی‌بی‌ام و مرکز تحقیقات لینکلن عرضه شد. گزارشات این مراکز نشان داد که دستگاههای ساخته شده از مواد نیمه هادی نقش بسیار عمده‌ای در فوتونیک خواهد داشت. در حال



شکل ۴- طرحی از آی‌سی‌های اپتیکی که متخصصین روی ساخت آنها کار می‌کنند. علام نوری توسط فیبرنوری وارد تراشه‌های مزبور می‌شود و از طریق موج‌برهای نوری درون آی‌سی به‌سوی آشکارسازهای نوری هدایت می‌شود و آنگاه تبدیل به علام الکتریکی شده و قابلیت پردازش رقمی را پیدا می‌کند و سپس توسط لیزرهای نیمه‌هادی مجدداً تبدیل به علام نوری می‌شوند. علام نوری تولید شده نیز به‌نوبه خود به‌کمک موج‌برهای نوری به محل اتصال فیبرهای نوری کوپل می‌شود. تاکنون شمار اندکی از اینگونه آی‌سی‌ها ساخته شده‌اند اما توجه زیادی را به‌خود معطوف ساخته‌اند.

دستگاههای الکترونیک به کار گرفته می شوند.

این وابستگی متقابل میان فوتونیک و الکترونیک اثرات تربیتی و آموزشی هم دارد. برنامه های آموزشی تربیتی دانشگاهها به سمتی میل می کند که درسهای مربوط به علم اپتیک را به دروس مورد نیاز مقاطع کارشناسی و کارشناسی ارشد و یا بالاتر منتقل سازند. به عنوان یک نتیجه مهم می توان انتظار داشت که مرکز ثقل تربیت کادر متخصص در عرصه اپتیک و فوتونیک به تدریج از دانشکده های فیزیک به دانشکده های مهندسی برق منتقل شود.

چشم انداز آینده

فوتونیک هنوز در ابتدای راه است و آینده امیدبخشی را برای آن می توان پیش بینی کرد. نتایج کاربردی آن را در صنعت و مشاغل می توان تا قرن بیست و یکم مشاهده کرد. فوتونیک و الکترونیک مکمل یکدیگرند و نه رقیب هم؛ در واقع این دو بسیار به هم وابسته اند و بنابر این زمانی موفقیت در زمینه های کاربردی به دست می آید که بتوان بین این دو عرصه پل ارتباطی برقرار کرد و از تاثیرات رشد سریع الکترونیک امروزی در عرصه فوتونیک بهره گرفت.

جدیدترین مسئله فوتونیک در دنیای کنونی کمبود متخصصین در این زمینه است که بتوانند زمینه های تکنولوژیک جدیدی را در پیشرفت این رشته فراهم آورند. به وجود آمدن رشته تحصیلی تحت عنوان مهندسی فوتونیک می تواند به این کار کمک کند. به طوری که در ادامه کار می توان انقلاب فوتونیک را پی آمد انقلاب الکترونیک دانست همانطور که انقلاب الکترونیک پی آمد انقلاب صنعتی بود.

حال این سؤال مطرح است که فوتونیک تا چه حدی قادر است گسترش یابد و اصولاً می تواند پا فراتر از بازار الکترونیک بگذارد؟ حداقل برای مدتهای طولانی پاسخ به این سؤال منفی است، زیرا الکترونیک به حد بلوغ و تکامل نسبی رسیده است. الکترونیک حداقل نیم قرن جلوتر است و به علاوه دارای شالوده و زیر بنای محکم نظری است و بعید به نظر می رسد بتوان آن را یکباره به کناری نهاد و به تکنولوژی تازه تری روی آورد.

سرگرمی

راست و دروغ

در یک جزیره دور افتاده، سه قبیله «راستگویان»، «دروغگویان» و «یک در میانها» زندگی می کنند. همانطور که از نامهای قبایل بر می آید، افراد قبیله اول همیشه راست می گویند، افراد قبیله دوم همیشه دروغ می گویند، و افراد قبیله سوم هرگاه در پاسخ به یک سؤال راست بگویند، در پاسخ به سؤال بعدی دروغ خواهند گفت و بر عکس. جهانگردی به این جزیره وارد می شود و به سه نفر به نامهای «آبان»، «آذر»، و «بهمن» بر می خورد.

از آبان می پرسد: «تو از کدام قبیله هستی؟».

آبان پاسخ می دهد: «از قبیله راستگویان».

از آذر می پرسد: «تو از کدام قبیله هستی؟».

آذر پاسخ می دهد: «از قبیله دروغگویان».

مجدداً از آذر می پرسد: «آیا آبان در پاسخ به سؤال من راست گفت؟».

آذر پاسخ می دهد: «بله».

از بهمین می پرسد: «تو از کدام قبیله هستی؟».

بهمین پاسخ می دهد: «از قبیله راستگویان».

از پاسخهای بالا، جهانگرد چه نتایجی می تواند بگیرد؟

ترتیب

اطلاعات زیر درباره پنج مقدار عددی A، B، C، D و E در دست هستند.

۱- A با C برابر است اگر و تنها اگر E با B برابر نباشد.

۲- تنها اگر C همان قدر از B کمتر باشد که B از A کمتر است، A بزرگتر از D خواهد بود.

۳- C کمتر از A و بیشتر از D است.

ترتیب مقادیر عددی بالا را به دست آورید.

گردآوری از بهروز پرهامی



گزارش

سمینار آموزش و انفورماتیک

بابلیان، عضو هیأت علمی دانشگاه تربیت مدرس و یکی از نویسندگان کتاب «اصول کامپیوتر و انفورماتیک» که از سال تحصیلی جاری در رشته ریاضی-فیزیک در دبیرستانها تدریس می‌گردد، شرکت داشتند. در این میزگرد، ابتدا توسط مسئولان شرکت‌کننده به چگونگی فعالیتهای انجام شده در زمینه مورد بحث در ایران پرداخته شد و پیشرفتهای به عمل آمده و کمبودها و مشکلات موجود مورد بحث قرار گرفتند. سپس به سؤالات حاضران در مورد مطالب عنوان شده در سخنرانیها و سخنان مسئولان پاسخ داده شد. اداره این میزگرد برعهده آقای ابطحی بود.

جلسه بعد از ظهر روز سه‌شنبه پنجم آذرماه به مبحث «آموزش دانشگاهی» اختصاص داشت و از ساعت ۱۳/۳۰ دقیقه با سخنرانی آقای ابراهیم نقیب‌زاده مشایخ عضو هیأت علمی مرکز تحقیقات فیزیک نظری و ریاضیات و رئیس انجمن انفورماتیک ایران در مورد «آموزش دانشگاهی انفورماتیک در ایران» آغاز گردید. ایشان با ذکر اینکه ارائه برنامه‌های آموزش دانشگاهی جهت تربیت متخصصان کامپیوتر در ایران از سال ۱۳۴۹ در دانشگاه صنعتی شریف تهران آغاز گردید و در حال حاضر ۱۲ دانشگاه کشور در زمینه ارائه دوره‌های کاردانی، کارشناسی سخت‌افزار و نرم‌افزار و کارشناسی ارشد سخت‌افزار و نرم‌افزار فعالیت دارند، برنامه‌های آموزش دانشگاهی در رشته کامپیوتر در ایران را در دو مقطع قبل از انقلاب فرهنگی و پس از آن مورد بررسی قرار داد. آقای مشایخ ضمن مروری بر برنامه‌های آموزشی از آغاز تاکنون و ارائه آماری از پذیرفته‌شدگان به دوره‌های مختلف آموزش کامپیوتر در دانشگاههای ایران و فارغ‌التحصیلان این رشته پس از انقلاب فرهنگی، بی‌هدف بودن و عدم توجه به جنبه‌های کاربردی را از اهم معایب برنامه آموزش فعلی دانست و تفکیک این برنامه به دو گرایش سخت‌افزار و نرم‌افزار را غیرضروری و نادرست خواند. ناطق در انتها پیشنهاداتی جهت بهبود برنامه آموزش فعلی مطرح ساخت.

سخنران بعدی، آقای دکتر محسن رستمی مدیر امور نرم‌افزار شرکت برنا و سرپرست دوره کارشناسی ارشد علوم کامپیوتر در دانشگاه آزاد اسلامی بودند. موضوع سخنرانی ایشان، «آموزش دانشگاهی علوم کامپیوتر در جهان» بود. آقای دکتر رستمی با بیان اینکه زمینه‌های نیاز به تکنولوژی کامپیوتر در کشورهای پیشرفته، عمده‌ترین محور تعیین‌کننده چهارچوب محتوای دوره‌های دانشگاهی است و بر این مبنا مرتباً محتوای دوره و دروس مورد تجدید نظر قرار می‌گیرند، نقش استراتژیک تکنولوژی نرم‌افزار را طی سالهای اخیر به عنوان عامل اصلی در گرایش فوق‌العاده به سوی دو موضوع مهندسی نرم‌افزار و سیستمهای اطلاعاتی دانستند. ایشان آموزش دانشگاهی کامپیوتر در جهان را با تأکید بر زمینه‌های

سمینار آموزش و انفورماتیک در تاریخ پنجم و ششم آذر ماه ۱۳۷۰ در محل آمفی تئاتر دانشکده علوم دانشگاه تهران برگزار گردید. برطبق برنامه تنظیم شده قبلی، سمینار در ساعت ۸/۳۰ دقیقه صبح روز سه‌شنبه پنجم آذر ماه با سخنان رئیس انجمن گشایش یافت. ایشان طی سخنان کوتاهی ضمن خوشامدگویی به حاضران، به فعالیتهای انجام گرفته در ارتباط با برگزاری سمینار و چگونگی انتخاب موضوعات مختلف آن اشاره کردند.

پس از پایان مراسم افتتاحیه، برنامه سخنرانیها با مبحث آموزش پیش‌دانشگاهی انفورماتیک آغاز شد. اولین سخنران سمینار، آقای سید ابراهیم ابطحی، عضو هیأت علمی دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف بودند که تحت عنوان، «آموزش پیش‌دانشگاهی انفورماتیک در ایران» به سخنرانی پرداختند. ایشان در سخنان خود با بیان اینکه آموزش پیش‌دانشگاهی انفورماتیک با سابقه‌ای پانزده ساله در ایران با توجه به عمر کمتر از نیم قرن آغاز فعالیتهای کامپیوتری در ایران، فعالیتهای قابل توجهی را در برداشته است به این نکته که ابعاد این آموزش تاکنون به شکل جدی و گسترده مطرح نگشته است، اشاره نمودند. آقای ابطحی در سخنرانی خود به تفصیل فعالیتهای مختلفی را که در زمینه آموزش پیش‌دانشگاهی انفورماتیک در ایران صورت گرفته است، از اولین فعالیتهای که شامل تدریس کتابی درباره کامپیوتر در رشته بازرگانی در دبیرستانها بوده تا آخرین فعالیت در دست انجام که آموزش دبستانی کامپیوتر است، مورد بررسی قرار دادند.

سخنران بعدی خانم فاطمه فقیهی، کارشناس ارشد سازمان پژوهش و برنامه‌ریزی آموزشی وزارت آموزش و پرورش بودند که درباره «آموزش پیش‌دانشگاهی انفورماتیک در جهان» سخنرانی نمودند. ایشان با ذکر این که دهه اخیر، دوران اوج‌گیری گسترش آموزش پیش‌دانشگاهی انفورماتیک در جهان بوده است، در سخنان خود، قلمروهای آموزش کامپیوتر در آموزش پیش‌دانشگاهی را با تأکید بر دو مورد انجام شده در کشور ژاپن و ایالات متحده آمریکا مورد بررسی قرار دادند. خانم فقیهی در ادامه سخنان خود به بررسی آثار مثبت و منفی حضور کامپیوتر در آموزش پیش‌دانشگاهی انفورماتیک در جهان پرداختند.

پس از یک استراحت کوتاه و پذیرایی از شرکت‌کنندگان، برنامه صبح روز اول سمینار از ساعت ۱۰/۴۵ دقیقه تا ۱۲/۱۵ دقیقه با تشکیل میزگردی در باره آموزش پیش‌دانشگاهی ادامه یافت. در این میزگرد، خانم فقیهی و آقای ابطحی سخنرانان این جلسه به همراه آقایان مهندس پورسیف مدیرکل مرکز تحقیقات و برنامه‌ریزی کتب درسی وزارت آموزش و پرورش، مهندس برات قنبری معاون مرکز تحقیقات مخابرات و دکتر

به کارگیری روشهای انفورماتیکی در طراحی دانستند.

سخنران بعدی آقای دکتر محمد صنعتی مدیر عامل شرکت نرم افزاری سینا بودند. ایشان تحت عنوان «آموزش یک زبان برنامه سازی توسط کامپیوتر» به بیان تجربه خود در تولید یک نرم افزار آموزش دهنده زبان پاسکال در کشور امریکا پرداختند. آقای دکتر صنعتی خصوصیات نرم افزار مربوطه و روش به کارگیری آن، نتایج تحقیقات یک تیم روانشناسی بر روی دو کلاس آموزش زبان پاسکال که یکی به شیوه سنتی و دیگری با استفاده از نرم افزار مربوطه اداره شده اند را ذکر کردند.

پس از یک استراحت کوتاه، میزگرد «آموزش به کمک کامپیوتر» از ساعت ۱۰/۳۰ دقیقه صبح روز ۶ آذر با شرکت خانم میرحسینی و آقای دکتر صنعتی سخنرانان جلسه صبح و آقایان محمد میرزاعبداللهی رئیس مرکز تحقیقات کامپیوتر ایران، دکتر شاپور اعتماد عضو هیأت علمی انجمن حکمت و فلسفه ایران و نقیب زاده مشایخ رئیس انجمن انفورماتیک ایران برگزار گردید. در این میزگرد ابتدا جنبه های مختلف آموزش به کمک کامپیوتر از قبیل یادگیری به کمک کامپیوتر و مدیریت آموزشی به کمک کامپیوتر مطرح گردید و سپس به سؤالات مختلف حاضران که بیشتر در مورد مسائل مختلف ناشی از جایگزینی کامپیوتر به جای معلم و مزایای آموزش سنتی از لحاظ روابط بین معلم و شاگرد بود، پاسخ داده شد. شرکت کنندگان در میزگرد تأکید داشتند که هدف از آموزش به کمک کامپیوتر، حذف نقش معلم نیست.

آخرین جلسه سمینار از ساعت ۱۳/۱۵ دقیقه بعد از ظهر چهارشنبه ششم آذر ماه آغاز گشت. موضوع جلسه، «آموزش عمومی انفورماتیک» بود. اولین سخنران آقای داریوش جوان تبریزی، مدیر امور دانشگاهها در شرکت پویا و نایب رئیس انجمن انفورماتیک ایران بودند که تحت عنوان «اولین درس کامپیوتر» به سخنرانی پرداختند. ایشان با ذکر اینکه کامپیوتر، ماشینی است که دارای قابلیت «شعورپذیری» بوده و در نتیجه در عین ماشین بودن، می تواند عملکردی انسان گونه داشته باشد، عنوان نمودند که شناخت بنیادی و زیربناساز از این ابزار شعورپذیر از یک طرف بایستی شامل کلیه مشخصه های اصولی «ابزارشناسی» و از طرف دیگر مولد تمامی امکانات اطلاعاتی و اجرایی مربوط به ویژگیهای غیر ماشینی و انسان گرایانه آن باشد. بنابراین، اولین درس کامپیوتر علاوه بر ایجاد و برپاسازی یک «شناخت عملیاتی یکپارچه» در آموزش گیرنده که از دیدگاه ابزارشناسی محض لازم است، باید به اهداف تکمیلی و جنبی دیگری نیز دست یابد. آقای جوان در سخنرانی خود به تشریح روش ابداعی خود برای دستیابی به اهداف فوق در زمینه آموزش اولیه کامپیوتر پرداختند.

سخنرانی بعدی از ساعت ۱۴ تحت عنوان «آموزش غیرآکادمیک» توسط آقای محمد حسن محوری مشاور امور انفورماتیک و عضو هیأت اجرایی انجمن انفورماتیک ایران ارائه گردید. ایشان با ذکر اینکه آموزش غیرآکادمیک، جایگاه خاصی در صحنه تولید و کاربرد تکنولوژی کامپیوتر دارد، به یکی از زمینه های عمده آن تحت عنوان آموزش

نیاز به تکنولوژی کامپیوتر در این کشورها مورد بررسی قرار دادند. در این زمینه، مروری بر تاریخچه و شکل گیری دوره های دانشگاهی و ساختار و محتوای متداول در دوره های کارشناسی، کارشناسی ارشد و دکترای علوم کامپیوتر و به ویژه برنامه های پیشنهادی یونسکو، انجمن ماشینهای محاسب امریکا^۱ و دانشگاه کارنگی ملون، از اهم مطالب عنوان شده بود.

پس از یک استراحت کوتاه و پذیرایی از حاضران، جلسه بعد از ظهر روز اول سمینار با برگزاری میزگردی درباره آموزش دانشگاهی ادامه یافت. شرکت کنندگان در این میزگرد را آقایان دکتر رستمی و نقیب زاده مشایخ سخنرانان جلسه بعد از ظهر و آقایان دکتر کریم فائز رئیس کمیته تخصصی برنامه ریزی مهندسی کامپیوتر در وزارت فرهنگ و آموزش عالی، دکتر امیر حسین جهانگیر عضو هیأت علمی و رئیس دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف تهران و علی پارسا مهندس ارشد نرم افزار در شرکت داده سیستمهای ایران و عضو هیأت اجرایی انجمن انفورماتیک ایران تشکیل می دادند. در این میزگرد ابتدا آقای دکتر فائز به تفصیل در مورد فعالیتهای کمیته تخصصی برنامه ریزی مهندسی کامپیوتر و اهداف دوره دانشگاهی مربوطه و مشکلات موجود صحبت کردند و سپس شرکت کنندگان در میزگرد به بیان نقطه نظرات خود و پاسخگویی به سؤالات مطرح شده پرداختند. عمده ترین مشکل دوره دانشگاهی فعلی، عدم کفایت فارغ التحصیلان جهت رفع نیازهای بازار کار و به ویژه بخش صنعت و نداشتن هدفی مشخص عنوان شده که قسمت عمده بحثهای میزگرد آموزش دانشگاهی را به خود اختصاص داد. مسئول کمیته تخصصی برنامه ریزی مهندسی کامپیوتر در وزارت فرهنگ و آموزش عالی که در میزگرد حضور داشتند نیز ضمن دفاع از برنامه فعلی و اعلام اینکه هنوز فارغ التحصیلان برنامه جدید از دانشگاهها خارج نشده اند، قضاوت در مورد آن را فعلاً زود دانستند.

بدین ترتیب، برنامه روز اول سمینار در ساعت ۱۷/۳۰ دقیقه خاتمه یافت. برنامه روز دوم سمینار از ساعت ۹ صبح روز چهارشنبه ۶ آذر آغاز گشت. موضوع جلسه صبح به «آموزش به کمک کامپیوتر» اختصاص داشت. اولین سخنران، خانم زهرا میرحسینی عضو هیأت علمی دانشکده مهندسی کامپیوتر در دانشگاه علم و صنعت و کارشناس ارشد نرم افزار در بانک مرکزی جمهوری اسلامی ایران بودند که تحت عنوان «نقش انفورماتیک در یادگیری» به سخنرانی پرداختند. در این سخنرانی، مشخصات و ویژگیهای سیستمهای یادگیری و چگونگی طراحی آنها که با رشد دانش انفورماتیک به عنوان شاخه ای از علوم سبیرنتیک مطرح شده است، ارائه گردید. خانم میرحسینی با ذکر اینکه آنچه متخصصان روانشناسی تفکر منطقی و جانبی می نامند و از دیدگاه تعلیم و تربیت تفکر همگرا و واگرا نام دارد، امروزه بر اساس شیوه های انفورماتیکی به شکل روشهای الگوریتمی و مکاشفای قابل آموزش دادن است؛ «یادگیری» را به عنوان یک سیستم پویا که مبتنی بر یک مدل ارزشیابی و یک مدل ترمیم است، تا حد زیادی مدیون

پس از یک استراحت کوتاه و پذیرایی از حاضران، میزگرد پایانی بحث آموزش عمومی با شرکت آقایان دکتر اعتماد، جوان و محوری سخنرانان جلسه و آقایان دکتر محسن صدیقی، مشکانی عضو هیأت علمی دانشکده مهندسی کامپیوتر دانشگاه شهید بهشتی و نقیبزاده مشایخ رئیس انجمن انفورماتیک ایران برگزار گردید. شرکت کنندگان در این میزگرد ضمن بیان نارسائیهای موجود در آموزش عمومی کامپیوتر در کشور، در مورد شرایط و زمینههای مورد نیاز این آموزش به بحث پرداختند و در پایان به سؤالات حاضران در این زمینه پاسخ گفتند.

با پایان این میزگرد، برنامه سمینار دو روزه آموزش و انفورماتیک پایان پذیرفت و در انتها، رئیس انجمن طی سخنان کوتاهی با ابراز تشکر از حاضران و آرزوی اینکه مطالب مطرح شده در طول سمینار مورد استفاده آنان قرار گرفته باشد، خاتمه سمینار را اعلام نمود.

غیرآکادمیک در بعد تولید، پشتیبانی و ارتقاء دانش فنی به منظور همگامی با روند تکنولوژی کامپیوتر پرداختند. آقای محوری ضمن مروری بر چگونگی امر آموزش در دیگر کشورها، وضعیت نامساعد آموزش غیرآکادمیک در کشور را مورد بررسی قرار دادند. این بررسی شامل تاریخچه، شرایط فعلی، زمینههای مورد نیاز و چهارچوب مورد انتظار از آموزش غیرآکادمیک کامپیوتر در کشور بود.

آخرین سخنرانی سمینار از ساعت ۱۴/۴۵ دقیقه با عنوان «تکنولوژی اطلاعاتی و بازآموزی» توسط آقای دکتر شاپور اعتماد عضو هیأت علمی انجمن حکمت و فلسفه ایران ایراد شد. آقای دکتر اعتماد در سخنرانی خود ضمن دسته بندی انواع تکنولوژی، نسبتهای گوناگونی را که انسان با آنان برقرار می کند مورد بررسی قرار دادند. ایشان در ادامه سخنان خود با توجه به تعریفی که از ماهیت تکنولوژی ارائه نمودند، در مورد اشکال کاربرد و آموزش متناسب با آن سخن گفتند.

در حاشیه برگزاری سمینار

محل برگزاری سمینار، سالن آمفی تئاتر دانشکده علوم دانشگاه تهران بود که از سوی مدیریت محترم دانشکده علوم به رایگان در اختیار انجمن قرار داده شده بود. انجمن انفورماتیک ایران بدینوسیله مراتب قدردانی عمیق خود را از آقای دکتر قدی ریاست محترم دانشکده علوم دانشگاه تهران، آقای دکتر صالحی معاونت محترم آموزشی دانشکده، آقای حاج امینی معاونت محترم امور اداری و مالی دانشکده به خاطر پذیرش درخواست انجمن و نیز آقایان کلانکی، طهماسبی و سایر مسئولان کارکنان گرامی دانشکده که با نهایت صمیمیت برای برگزاری سمینار همکاری نمودند، ابراز می دارد.

هیأت برگزار کننده سمینار بدینوسیله از همکاری صمیمانه آقای بهزاد طالبپور عضو گرامی انجمن و سایر دانشجویان و درستان عزیز که چه در خلال مراحل پیش از برگزاری سمینار از قبیل ثبت نام، انجام امور اداری، آماده سازی محل برگزاری و چه در طول برگزاری سمینار همچون بازوان پرتوانی به انجمن یاری رساندند، عمیقاً قدردانی و سپاسگزاری می نماید.

• هیأت برگزار کننده سمینار از سخنرانان و مسئولان محترمی که دعوت انجمن را برای شرکت در سخنرانیها و میزگردهای سمینار پذیرفتند، صمیمانه قدردانی و سپاسگزاری می نماید.

• متأسفانه علیرغم برنامه پر محتوای سمینار و تلاش زیادی که برای جلب سخنرانان و شرکت کنندگان در میزگردها به عمل آمده بود، استقبال زیادی از جلسات سمینار به عمل نیامد و اکثراً نیمی از سالن خالی بود.

• شورای عالی انفورماتیک کشور، پرداخت هزینه های اجرایی سمینار را برعهده گرفت. انجمن انفورماتیک ایران بدینوسیله از مسئولان محترم شورا قدردانی می نماید.

• خلاصه سخنرانیها و شرح حال سخنرانان از طرف هیأت برگزار کننده سمینار تهیه و تکثیر شده بود و در روز شروع سمینار در اختیار شرکت کنندگان قرار داده شد.

• طبق برنامه از پیش تعیین شده سمینار، قرار بود خانم آرمیتا کامفر کارشناس نرم افزار شرکت داده پردازی ایران در روز چهارشنبه ۶ آذرماه تحت عنوان آموزش به کمک کامپیوتر سخنرانی نمایند که متأسفانه ایشان در فاصله کمی قبل از شروع سمینار، عدم آمادگی خود جهت ایراد سخنرانی را به اطلاع برگزار کنندگان سمینار رساندند! بدین خاطر در برنامه صبح روز دوم سمینار اختلالاتی پیش آمد که با جابه جایی سخنرانی خانم میرحسینی و موافقت آقای دکتر صنعتی که در جلسه حضور داشتند برای ایراد سخنرانی در مورد نرم افزار آموزشی Paslab، رفع و رجوع گردید! هیأت برگزار کننده سمینار بدینوسیله از همکاری خانم میرحسینی و به ویژه آقای دکتر صنعتی صمیمانه سپاسگزاری می نماید.

• از آقای حسین غفاری از مسئولان آموزش شرکت داده پردازی ایران دعوت به عمل آمده بود تا در میزگرد پایانی بحث آموزش عمومی شرکت نمایند و ایشان نیز موافقت خود را اعلام نموده بودند ولی در صبح روز دوم سمینار عدم شرکت خود را در سمینار به اطلاع هیأت برگزار کننده سمینار رساندند!

• از دبیر شورای عالی انفورماتیک دعوت به عمل آمده بود تا در میزگرد پایانی بحث آموزش عمومی شرکت کنند، ایشان طی نامه ای از مسئولان انجمن خواستند که تا در جلسه ای حضور یابند و در مورد اهداف انجمن از برگزاری این سمینار به ایشان و دیگر مسئولان شورا توضیح دهند! نایب رئیس انجمن در آن جلسه حضور یافت و توضیحات لازم را ارائه نمود ولی با این وجود، دبیر شورا یا نماینده های از ایشان در میزگرد حضور نیافت. البته دبیر کمیته آموزش شورای عالی انفورماتیک در تمام جلسات سمینار حضور داشتند.

• کلیه سخنرانیها و بحثهای مطرح شده در میزگردها توسط یکی از اعضای علاقه مند انجمن بر روی نوار ضبط شد و در حال پیاده شدن است. امید است مجموعه مقالات سمینار به زودی در دسترس علاقه مندان قرار گیرد.

گزارش

کامپیوترهای چند رسانه‌ای*

اگر در مورد آینده کامپیوترهای شخصی یک مسئله باشد که همه در آن اتفاق نظر دارند، ویژگی چند رسانه‌ای آنها است. بدین مفهوم که برای ایجاد ارتباط با انسان فقط به استفاده از متنهای سبز، نارنجی و یا نمودارهای ستونی ساده اکتفا نخواهد شد. کامپیوترهای فردا صدای انسان را شبیه‌سازی کرده، نتیجهٔ تجارب علمی را به صورت تصاویر زنده نمایش میدهند و حتی می‌توانند موسیقی جاز را بهتر پخش کنند.

کامپیوترهای چند رسانه‌ای با شبیه‌سازی همه چیز از «همشهری کین» و «مود ایندیگو» گرفته تا حلقه‌های DNA و تصادف ماشینها، پنجره‌ها را به روی دنیای جدیدی از اطلاعات و سرگرمیها می‌گشایند. صنایع راکد کامپیوترهای شخصی اکنون به دنبال بالا بردن چاشنی نمایشی هستند تا بدین ترتیب استفاده از کامپیوترها را ساده‌تر کرده و بازار وسیع مصرف را جذب خود کنند.

به دلیل مشکل بودن استفاده از کامپیوتر و یا محدودیت در دسترسی به آن تنها ۲۵ درصد مردم آمریکا از کامپیوتر بهره می‌گیرند. تکنولوژی همه کاره کردن کامپیوترها متعهد شده حتی بیسوادان نیز بتوانند در عصر اطلاعات از کامپیوتر استفاده کنند. در شهر تولارا، کالیفرنیا، کشاورزان به عنوان کسب خدمات اجتماعی برای تهیه کامپیوترهایی که با لمس نمودن صفحه نمایش فرمان می‌برند ثبت نام کرده‌اند. این کامپیوترها چگونگی درو کردن محصول را با تصاویر ویدیویی به زبانهای انگلیسی، اسپانیولی و چهار زبان آسیای جنوب شرقی آموزش می‌دهند.

آقای پیتربلاک، مدیر کمپانی Xiphias، تولید کننده دیسکهای لیزری چندمنظوره در ایالت کالیفرنیا می‌گوید «بازار کامپیوتر فعلی، مانند صنعت سینما قبل از ورود صدا به آن است». مشکل این است که هیچکس کاملاً در مورد چگونگی همه کاره بودن این ماشینهای تازه وارد مطمئن نیست و این موجب عدم وجود استاندارد برای نرم‌افزار و متعاقباً عدم تشویق به تولید برنامه‌های از پیش ساخته خواهد شد. آیا کامپیوترهای چند رسانه‌ای ترکیبی از کامپیوترهای شخصی آی‌بی‌ام و مکینتاش ابل خواهند بود یا اینکه کامپیوترهایی کاملاً جدید که برای کاربردهای چندرسانه‌ای طراحی شده‌اند؟ و یا فقط ممکن است نوعی تلویزیون توسعه یافته برای معتادین به تماشای تلویزیون باشند. اینکه این کامپیوترها و دیسکهای چند رسانه‌ای‌شان در چه محلی به فروش می‌رسند، فروشگاههای کامپیوتری یا وسایل الکترونیک و یا در عمده‌فروشیها معلوم نیست.

John W. Verity, "Multimedia Computing", Business Week, Aug. 12, 1991.

در واقع کسی معنی چند رسانه‌ای بودن این کامپیوترها را نمی‌داند، فقط بعضی استنباطهای مبهم مانند ارتباط تلویزیون با کامپیوتر در مورد آنها شنیده می‌شود. آقای نیک آرنت، رئیس کمپانی تحقیقاتی کامپیوترهای چندرسانه‌ای واقع در سانتاکلارا، کالیفرنیا، می‌گوید «کامپیوتر همه کاره مثل فیلم است، زمانی می‌توان آن را فهمید که آن را دید یا از آن شنید». اما این مسئله مانع از پیش بینی حجم توسعه این تکنولوژی در بازار آینده سینما اسکوپ نمی‌شود. شرکت تحقیقاتی Inteco واقع در نورواک، کانتیکات، پیش بینی کرده است که درآمد ۴/۸ میلیارد دلاری کامپیوترهای همه کاره در سال ۱۹۹۱ به ۲۲ میلیارد دلار در سال ۱۹۹۵ خواهد رسید.

اغلب سیستمهای شگفت‌آور چند رسانه‌ای فعلی بر مبنی سفارش برای مقاصد کارآموزی در صنایع، آموزش و کیوسکهای عمومی اطلاعات تولید شده‌اند. برای مثال دانشجویان در دانشگاه کرنل، فیزیولوژی را به وسیله کامپیوترهای مکینتاش همراه با دیسکهای نوری حاوی متون، طرحهای دقیق و تصاویر متحرک فرا می‌گیرند. همچنین دانش‌آموزان دبیرستان با استفاده از کامپیوترهای رومیزی آی‌بی‌ام به همراه سیستم ویدیو دیسک، شعر «اولیس» اثر تنیسان شاعر انگلیسی را مطالعه می‌کنند. آنها می‌توانند تفسیر و تحلیلهای ثبت شده در ویدیو را به جلو و عقب برده، به علاوه مبحثی را درباره ارتباط فن شاعری و آوازه‌ای رپ را مورد مطالعه قرار دهند.

استفاده از تکنولوژی به منظور تفسیر سیلی از اطلاعات موجود برای استفاده‌کنندگان هوشمند، دانشمندان، تحلیلگران و مهندسين امری اجتناب ناپذیر و ضروری است. محققین سراسر گیتی در آینده قادر خواهند بود توسط شبکه‌ای از کامپیوترهای رومیزی کنفرانس داده و همزمان تصاویر یکدیگر را مشاهده و از تخته سیاه الکترونیکی به طور مشترک استفاده کنند و فیلمهایی را که از داده‌های آزمایشگاهی تولید شده ملاحظه کنند.

بنابراین گرایش بازار آینده به سمت این خصوصیات است. تقریباً کلیه منابع اطلاعاتی در خانه از قبیل خطوط تلویزیونی، دستگاههای ویدیو، دیسکهای حاوی اطلاعات، کتابها، روزنامه‌ها، رادیو، دیسکهای لیزری و حتی تلفن هدف کامپیوترهای همه کاره است.

در نتیجه این تحولات، بین تولید کنندگان ژاپنی و آمریکایی جدال تاریخی در حال شکل‌گیری است. با وجودیکه صنایع آمریکا در زمینه نرم افزار و کامپیوتر پیشگام‌اند اما غولهای الکترونیک ژاپن در همگرا نمودن کامپیوتر و سایر وسایل الکترونیکی مصرفی و سرگرم‌کننده در موقعیت برتری هستند. برای مثال، شرکت سونی و صنایع الکترونیک ماتسوشیتا پیشگام در تولید وسایل الکترونیکی سرگرم‌کننده و همچنین مالک شرکتهای بزرگ آمریکایی هستند. این شرکتهای بسیاری از مواد اولیه لازم برای تولید فیلم و پخش صدا را تولید میکنند. بسیاری از ناظران صنایع معتقدند یکی از دلایل عجله شرکت آی‌بی‌ام برای همکاری با شرکت ابل موقعیت حاضر ژاپنهاست و با این نزدیکی سعی در پیش‌گیری از نزدیکی بیشتر و خطر ساز ابل با شرکت سونی

گزارش

دارد.

به جای صفحه کلید*

در دهه ۶۰ خانم گلوریا استاینم که از اعضای نهضت زنان آمریکا بود به زنانی که دنبال کار می‌گشتند پیشنهاد می‌کرد که نگویید ماشین نویسی بلدید چرا که در آن صورت شما را به کار دفتری می‌گمارند. پیشرفتهای اخیر در صنعت کامپیوتر این سؤال را مطرح می‌کند که تا کی ما باید وابسته به صفحه کلید باشیم. سه تحول عمده در پیش است: کامپیوتر دستخطی^۱، کامپیوتر صدا شناس^۲ و کامپیوتر پوشیدنی^۳.

کامپیوتر دستخطی چیز جدیدی نیست. به جای استفاده از صفحه کلید برای وارد کردن اطلاعات این سیستمها به کاربران اجازه می‌دهند که اطلاعات را از طریق نوشتن روی یک صفحه صاف که روی میز قرار می‌گیرد، وارد کنند. شرکت «گرید سیستم» یکی از شعبات شرکت تندی در کالیفرنیا کامپیوترهایی می‌سازد که دردست جا می‌گیرند و در حقیقت دفتر یادداشت الکترونیکی هستند. برای مثال شرکت «یونایتد پارسل سرویس» که موسسه انتقال امانات و بسته‌هاست به جای استفاده از کاغذ که قبلاً به کار می‌برده از کامپیوترهای این شرکت استفاده می‌کند تا بر حمل و تحویل بسته‌ها نظارت کند. بنابراین وقتی یکی از کارکنان برای وارد کردن اطلاعات روی صفحه نمایش کامپیوتر از قلم الکترونیکی استفاده می‌کند بلافاصله این اطلاعات به کامپیوتر مرکزی منتقل می‌شود. این شرکت در نظر دارد بدین وسیله مقدار زیادی از کارهای نوشتنی روزمره خود را کاهش دهد.

این هفته در نمایشگاهی که در شهر لاس‌وگاس تشکیل می‌شود و معمولاً صد هزار نفر را به خود جلب می‌کند، قرار است چند شرکت، کامپیوتر دستخطی عرضه کنند. یکی از آنها که تازه وارد میدان شده است به کاربر اجازه می‌دهد که بین قلم الکترونیکی و صفحه کلید یکی را انتخاب کند. کاربر می‌تواند کامپیوتر را با خود به هر جایی ببرد بدون آنکه صفحه کلید را نیز همراه داشته باشد. چه نیازی به این کار است؟ یکی از کاربردهای آن هنگام یادداشت برداشتن در جلسات اداری است. آوردن کامپیوتر در جلسات و درآوردن صدای صفحه کلید هنگام سخنرانی رئیس، دور از آداب معاشرت است.

بسته به نوع کامپیوتر، متن دستنویس را می‌توان به همان صورت نوشته شده، ذخیره کرد و یا اینکه بعد از تبدیل به شکل تایپ شده، آن را ضبط کرد. تبدیل متن به شکل تایپ شده خیلی مشکل است و زیاد قابل اعتماد نیست. اشکال در این است که کامپیوتر به آسانی نمی‌تواند دستخط را بخواند.

با ایجاد استاندارد مناسب، مانند آنچه در سیستمهای ویدیویی ملاحظه می‌شود، احتمال پیشرفت و توسعه کامپیوترهای همه کاره خانگی بسیار زیاد است. علاوه بر همکاری آی‌بی‌ام و اپل شرکت مایکروسافت و تندی نیز در صدد تهیه استاندارد برای تولید کامپیوترهای همه کاره هستند. به عنوان مثال، مایکروسافت قابلیت‌های ضبط و پخش صدا را توسط دیسکهای لیزری به نرم افزار خود موسوم به «ویندوز» اضافه می‌کند. بخش وسایل الکترونیکی مصرفی فلیپس هلد نیز با کمک سونی و ماتسوشیتا سعی در تولید دیسکهای چند رسانه‌ای با قالب خودش دارد.

با عرضه کامپیوتر همه‌کاره تاندی و شروع برنامه فروش فلیپس در پاییز امسال شاهد ایجاد بازار وسیعی خواهیم بود. سیستمهای همه کاره فلیپس شامل دیسکهای لیزری با قابلیت ذخیره سازی متون، پخش صدا و تصویر است که توسط نرم‌افزار هماهنگ و قابل استفاده می‌شوند. یک نمونه از کاربرد این گونه سیستمها، دایره‌المعارفهای همه کاره خواهد بود. کمپانی بین المللی کمودور در حال فروش CDTV است. این سیستم ۱۰۰۰ دلار، کامپیوتری است که به کلیه تلویزیونها قابل نصب بوده و شبیه به یک سیستم ویدیو است. می‌توان آن را از راه دور کنترل کرد. یکی از اولین دیسکهای این سیستم «اشعار ترسناک برای بچه‌های شیطان» است که به وسیله تصاویر و پخش صدا به زبانهای انگلیسی و اسپانیایی کودکان را آموزش داده و آنها را به هیجان می‌آورد. دیوید رازن، مدیر بازاریابی بین المللی کمودور این سیستمها را «اسب تروا» می‌نامد که کامپیوترها را به خانه می‌برد.

شرکتهای آی‌بی‌ام، اپل، مایکروسافت، سونی و سایرین با دقت ناظر فعالیتهای اخیر هستند و به نظر می‌رسد هر کدام کامپیوترهای همه‌کاره خود را تا پایان سال به بازار عرضه خواهند کرد. این شرکتها میزان فروش شرکتهای کمودور و فلیپس را به دقت تعقیب می‌کنند. با توان بالقوه بالایی که در بازار برای کامپیوترهای چند رسانه‌ای ملاحظه می‌شود، شرکتها سعی دارند تا از تکرار اتفاقی که برای سیستمهای بتامکس در مقابل VHS افتاد جلوگیری کنند و به این ترتیب سهم بیشتری از بازار را به خود اختصاص دهند.

ترجمه سعید داعی‌حق



Michael Rogers, "Speak Nicely to Your PC", *
Newsweek, October 28, 1991.

pen-based computing ۱
voice recognition ۲
wearable computer ۳



را انجام می‌دهد. عیناً مثل اینکه با صفحه کلید به آن فرمان داده شده باشد.

آخرین راه خلاصی از دست صفحه کلید، کامپیوتر پوشیدنی است. شرکت‌های ژاپنی در تلاش هستند از نمونه‌های اولیه پافراتر گذارند. اساس کار ساده است. واحد مرکزی پردازش را که متصل به یک عینک مخصوص است درون کیسه‌ای قرار می‌دهید و بعد دور کمر خود می‌بندید. این عینک دارای یک صفحه نمایش از کریستال مایع است که معادل همان صفحه کامپیوتر است که به صورت شیشه‌هایی در جلوی چشم‌ها قرار دارد (در ضمن می‌توان هنگام انجام کارهایی مانند پارک کردن ماشین، با خاموش کردن دستگاه، عدسیها را تبدیل به شیشه عینک معمولی کرد). این عینک‌ها به میکروفونهای کوچکی متصل هستند که روی شقیقه‌ها قرار می‌گیرند و برای تشخیص صدا به‌کار می‌روند.

البته هنوز زود است که در فروشگاه‌ها دنبال این وسایل بگردید. تمام این پیشرفتهای یعنی کامپیوترهای دستخطی و تشخیص صدا و پوشیدنی ما را وادار می‌کنند که مثل کامپیوتر عمل کنیم. کامپیوترهای دستخطی موقعی که کاربر ساده و خوب بنویسد بهترین کارایی را خواهند داشت. در ژاپن پیشرفت بیشتری حاصل شده است چرا که به دانش آموزان مدارس یاد می‌دهند که حروف را با دقت و یکنواخت بنویسند و شروع و خاتمه در یک خط باشد. کامپیوترهای تشخیص صدا موقعی که شخص با صدای یکنواخت صحبت کند بهترین اثر را خواهند داشت ولی اگر با صدای گرفته و یا هیجان صحبت کنید کامپیوتر نخواهد فهمید که چه می‌گویید. اما درمورد کامپیوترهای پوشیدنی، شما خود کامپیوتر هستید اگر عینک‌تان یا دستگاهی را که به دور کمرتان بسته‌اید از دست بدهید، دیگر کامپیوتری ندارید.

ترجمه هما فاضل‌نیا

برای مثال تشخیص حروف خمیده مانند o,e یا c بسیار دشوار است. در مقابل، کار سیستمهای دسته اول که می‌توانند مطالب را عیناً ضبط کنند و دوباره نمایش دهند، بسیار ساده است. یکی دیگر از دستگاههایی که می‌توان به جای صفحه کلید استفاده کرد، دستگاه تشخیص صداست. این دستگاه کامپیوتر را به یک دستگاه نامه‌نگار بسیار پیشرفته تبدیل می‌کند، با ظهور دیسکهای فشرده مانع اساسی تکنولوژیک مرتفع شده، چرا که صوت را چه به صورت گفتار، آواز یا موسیقی می‌توان ضبط کرد و به شکل رقمی تبدیل نمود. این دستاورد هنوز در مرحله آزمایشگاهی است و نمی‌توان آن را به صورت قابل دسترسی و با قیمت مناسب در اختیار کارمندان یک اداره قرار داد. مشکل زمانی است که می‌خواهیم علائم رقمی را به شکل نوشتاری آنها تبدیل کنیم. مهندسين در تلاش هستند تا نرم افزاری طراحی کنند تا تفاوت بین گونه‌های مختلف صداها را تشخیص دهد. تبدیل کردن صدا برای کامپیوتر خیلی مشکلتر از خواندن متن دستنویس است. حتی افراد خوشبین می‌گویند حداقل ۵ سال طول خواهد کشید تا کامپیوتری که بتواند دیکته بنویسد تکمیل شود.

ولی پیشرفتهای جدید دیگری نیز حاصل شده است. دانشگاه کارنگی ملون در ایالت پنسیلوانیا توانسته است سیستمی بسازد که می‌تواند ۵۰۰۰ کلمه را تشخیص دهد. آی‌بی‌ام نیز سیستم مشابهی را در آزمایشگاههای خودش دارد ولی هنوز آنها را به صورت تجارتي عرضه نکرده است. برای کسانی که عجله دارند چند نوع محصول به بازار آمده است که یکی از آنها بسته نرم‌افزاری شرکت «آرتیکولت سیستم» برای کامپیوترهای مکینتاش است (بسیاری از مکینتاشهای جدید مجهز به میکروفون هستند). هر چند که این نرم‌افزار محدودتر از پروژهای کارنگی ملون یا آی‌بی‌ام است ولی به شخص اجازه می‌دهد که به کامپیوتر بگوید تصحیح کن، حذف کن، یا ضبط کن و کامپیوتر کار مناسب

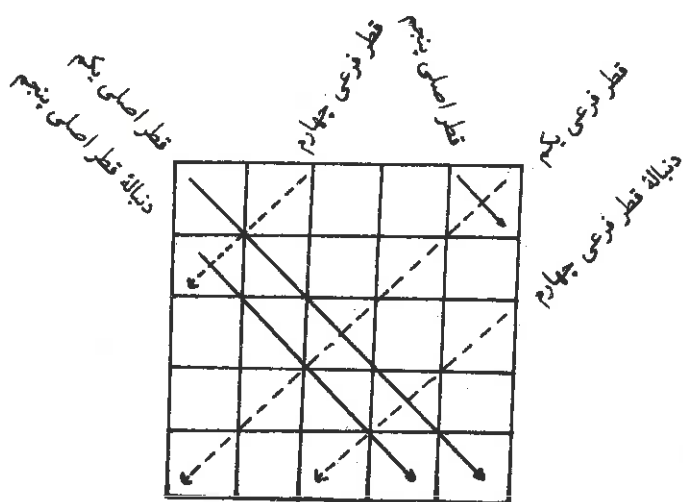
گزارش

المپیاد ملی انفورماتیک

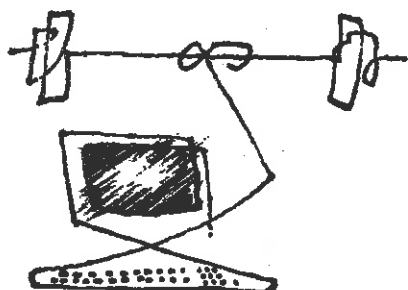
الف. مجموعه $S = \{1, 2, \dots, N\}$ مفروض است. الگوریتمی بنویسید که یک ماتریس $N \times N$ را در نظر بگیرد و برای هر درایه ماتریس A در صورتی که آن درایه متعلق به مجموعه S باشد درایه متناظر را در ماتریس D مساوی یک و در غیر این صورت مساوی صفر قرار دهد.

ب. الگوریتمی بنویسید که دو بردار N تایی IQ و JQ را در نظر بگیرد و در صورتی که در سطر iam ماتریس A عضو تکراری وجود داشته باشد مولفه iam بردار IQ را مساوی یک و در غیر این صورت مساوی صفر قرار دهد. مشابهاً چنانچه در ستون iam ماتریس A عضو تکراری وجود داشته باشد مولفه iam بردار JQ را مساوی یک و در غیر این صورت مساوی صفر قرار دهد.

ج. الگوریتمی بنویسید که حاصل ضرب قطری $A * B$ را به دست آورده در یک ماتریس $N \times N$ دیگر، C ، قرار دهد. ضرب قطری دو ماتریس، $A * B$ ، شبیه ضرب معمولی آن دو می باشد. با این تفاوت که به جای ضرب شدن سطر iam ماتریس اول در ستون iam ماتریس دوم، قطر اصلی iam ماتریس اول در قطر فرعی iam ماتریس دوم ضرب می شود. برای مثال قطره های اصلی و فرعی یک ماتریس 5×5 در شکل زیر نشان داده شده است:



د. با استفاده از الگوریتم های فوق، الگوریتمی بنویسید که در صورتی که اعضای دو ماتریس A و B متعلق به مجموعه S باشند و هیچ سطر و ستون آنها عدد تکراری وجود نداشته باشد، حاصل ضرب قطری $A * B$ را محاسبه نماید.



اولین مرحله المپیاد ملی انفورماتیک روز جمعه ۱۵ آذر ۷۰ در سراسر کشور برگزار شد. تعداد شرکت کنندگان در این مرحله حدود ۲۰۰۰ نفر بود که از بین دانش آموزان سالهای سوم و چهارم دبیرستان در رشته ریاضی-فیزیک داوطلب شده بودند. برگزیدگان مرحله اول (حدود ۱۵۰ نفر) به مرحله دوم راه می یابند که در دهه فجر در آبادان برگزار خواهد شد و از بین آنان اعضای تیم شرکت کننده در المپیاد جهانی برگزیده می شوند.

با توجه به اینکه درس مبانی کامپیوتر و انفورماتیک در دبیرستانها هنوز مراحل آغازی خود را می گذراند، تعداد شرکت کنندگان در این مرحله قابل توجه است و المپیاد انفورماتیک می تواند اشتیاق نوجوانان به فراگیری مبانی کامپیوتر و انفورماتیک را هدایت کرده و در یک مسیر آموزشی سنجیده قرار دهد. سئوالات مرحله اول به قرار زیر است. برای پاسخ به هر سؤال که چهار قسمت دارد، ۴۵ دقیقه وقت تعیین شده بود که در دو نوبت صبح و بعدازظهر، داوطلبان به پاسخگویی پرداختند.

در مسائل زیر الگوریتم های خواسته شده در هر قسمت را می توانید با هر یک از روشهای گام به گام، نمودار گردش یا زبان برنامه نویسی (پسیک یا پاسکال) بیان کنید. از نظر ارزشیابی هیچیک بر دیگری مزیتی ندارد و فقط درستی یا نادرستی راه حل مد نظر قرار می گیرد.

مسئله ۱.

الف. الگوریتمی بنویسید که عدد طبیعی N را از ورودی بگیرد و تعداد زیر مجموعه های ۴ عضوی یک مجموعه N عضوی را پیدا کرده، در متغیر K ذخیره نماید.

ب. الگوریتمی بنویسید که اعداد طبیعی M و N را از ورودی بگیرد و تعداد زیر مجموعه های M عضوی یک مجموعه N عضوی را پیدا کرده، در متغیر L ذخیره نماید.

ج. الگوریتمی بنویسید که عدد طبیعی N را از ورودی بگیرد و زیر مجموعه های ۴ عضوی مجموعه $\{1, 2, \dots, N\}$ را تولید کرده، در یک ماتریس $K \times 4$ ذخیره نماید. K تعداد این زیر مجموعه هاست که از قسمت «الف» به دست می آید.

د. الگوریتمی بنویسید که اعداد طبیعی M و N را از ورودی بگیرد و زیر مجموعه های M عضوی مجموعه $\{1, 2, \dots, N\}$ را تولید کرده، در یک ماتریس $L \times M$ ذخیره نماید. L تعداد این زیر مجموعه هاست که از قسمت «ب» به دست می آید.

مسئله ۲. فرض کنید A, B ماتریس های $N \times N$ هستند که در هر درایه آنها یک عدد صحیح قرار داده شده است.

واژه‌نامه

تغییرات و اضافات واژه‌نامه مقدماتی

کمیتۀ بازبینی واژه‌نامه در نشست‌های متعدد خود، بازبینی واژه‌هایی را که با حرف K و L شروع می‌شوند به پایان رسانده است. در زیر، نتیجه این بازبینی و تغییرات و اضافاتی که توسط کمیتۀ 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 کمیتۀ بازبینی واژه‌نامه به نشانی انجمن تماس حاصل نمایند. کار بازبینی واژه‌نامه مقدماتی ادامه دارد و نتایج حاصله، حرف به حرف به اطلاع اعضای گرامی انجمن و خوانندگان ماهنامه خواهد رسید. کمیتۀ بازبینی واژه‌نامه امیدوار است که از نظرات سازنده صاحب‌نظران و علاقه‌مندان برخوردار شود. قبل از ارائه فهرست واژه‌ها، یادآوری نکات زیر را لازم می‌دانیم:

- ۱- در یافتن معادل فارسی برای واژه‌ها، غالباً جنبۀ «اسمی» واژه‌ها در نظر گرفته شده است.
- ۲- واژه‌های غیرکامپیوتری واژه‌نامه 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.

K

L

Karnaugh map	نقشه کارنو، جدول کارنو
K-ary	کلتایی
Kernel	هسته
Key	کلید
Key Change	تبدیل کلید
Key Compression	فشرده سازی کلید
Key Dependency	وابستگی کلید
Keyboard	صفحه کلید
Keyboard entry	ورودی صفحه کلیدی، ورودی از طریق صفحه کلید
Keyboard inquiry	پرسش از طریق صفحه کلید
Keyboard lockout	قفل کردن صفحه کلید
Keyboard printer	صفحه کلید با چاپگر
Keyswitch	کلید دستی
Keyword	کلید واژه
Kilocycle	هزار چرخه
Knot	گره کور
Know-how	معرفت عملی، شناخت عملی
Knowledge	معرفت، شناخت، دانش
Knowledge acquisition	کسب معرفت
Knowledge base	پایگاه معرفت
Knowledge engineering	مهندسی معرفت
Knowledge representation	بازنمایی معرفت
Knowledge-based system	سیستم معرفت پایه، سیستم مبتنی بر معرفت

LABEL	برچسب
LABELED FILE	پرونده برچسب دار
LABELED TREE	درخت برچسب دار
LADDER NETWORK	شبکه نردبانی
LAG	پس افت، تأخیر
LANGUAGE	زبان
LANGUAGE PROCESSOR	پردازشگر زبان، زبان پرداز
LANGUAGE-BASED PROTECTION	حفاظت بر پایه زبان، حفاظت زبانی
LARGE-SCALE INTEGRATION (LSI)	مجموع سازی در مقیاس بزرگ
LASER PRINTER	چاپگر لیزری
LAST-IN-FIRST-OUT	بر عکس ترتیب ورود
LATE BINDING	مقید سازی تأخیری
LATENCY TIME	زمان کمون، زمان نهان
LATERAL	جانبی
LATTICE	مشبک، توری
LATTICE NETWORK	شبکه توری
LAYER	لایه
LAYERING	لایه بندی
LAYOUT	طرح بندی
LEAF	پرگ
LEARNABILITY	قابلیت یادگیری
LEARNING MACHINE	ماشین یادگیرنده
LEARNING RULES	قواعد یادگیری

LEARNING SYSTEM	سیستم یادگیر
LEASED LINE	خط واگذار شده، خط استیجاری
LEAST FREQUENTLY USED	با کمترین تکرار استفاده
LEAST RECENTLY USED	با کمترین استفاده در گذشته
LEAST SIGNIFICANT	کم معنی‌ترین، کم اهمیت‌ترین
LEAST SQUARES METHOD	روش کمترین مجنورات
LEDGER	معین
LEFT DERIVATIVE	مشتق چپ
LEFT RECURSIVE	بازگشتی از چپ
LEFT SHIFT	تغییر مکان به چپ
LEFT-JUSTIFIED	هم تراز شده از چپ
LEFT-TO-RIGHT PARSE	تجزیه از چپ به راست
LENGTH	طول
LETTER-QUALITY PRINTER	چاپگر با کیفیت خوب
LEVEL	سطح
LEXICAL	واژگانی
LEXICAL AMBIGUITY	ابهام واژگانی
LEXICAL ANALYSIS	تحلیل واژگانی
LEXICOGRAPHIC ORDERING	به ترتیب واژگانی
LIBRARY	کتابخانه

LIBRARY CALL	فراخوانی از کتابخانه	LINKAGE LOADER	بارکننده پیوندی	LOCK OUT	تحریم
LIBRARY PROGRAM	برنامه کتابخانه‌ای	LINKED LIST	لیست پیوندی	LOG	ثبت وقایع
LIBRARY ROUTINE	روال کتابخانه‌ای	LINKING LOADER	بارکننده پیوندی	LOG-IN	ورود به سیستم، برقراری ارتباط
LIBRARY SUBROUTINE	زیرروال کتابخانه‌ای	LIQUID-CRYSTAL DISPLAY	صفحه نمایش با کریستال مایع	LOG-OFF	خروج از سیستم، قطع ارتباط
LIFE CYCLE	چرخه عمر، دوره عمر	LIST	لیست	LOG-OUT	خروج از سیستم، قطع ارتباط
LIGHT GUN	تفنگ نوری	LIST PROCESSING	پردازش لیستها، لیست پردازش	LOGIC	منطق
LIGHT PEN	قلم نوری	LIST STRUCTURE	ساختار لیست	LOGIC-ARITHMETIC UNIT	واحد حساب و منطق
LIGHT-EMITTER	گسیلگر نور، ساطع کننده نور	LISTING	لیست	LOGICAL	منطقی
LIMIT	حد	LITERAL	نویسه لفظی	LOGICAL ADDRESS	نشانی منطقی
LIMITED ROUND ROBIN	با گردش نوبت محدود	LITERAL MEANING	معنی لفظی	LOGICAL RECORD	رکورد منطقی
LINE	خط، سطر	LITERAL OPERAND	عملوند لفظی	LONG-DISTANCE	راه دور
LINE EDITOR	ویرایشگر خطی	LIVEWARE	زیست افزار	LONG-TERM MEMORY	حافظه بلند مدت
LINE PRINTER	چاپگر سطری	LOAD	بار کردن	LOOKAHEAD	پیش بینی
LINE SPACING	فاصله گذاری بین سطرها، فاصله سطرها	LOAD MODULE	واحد بار شوند، واحد بار	LOOK-ASIDE BUFFER	میانگیر کناری
LINE-FEED CHARACTER	نویسه تعویض سطر	LOAD-AND-GO	بار و اجرا	LOOK-UP TABLE	جدول مراجعه
LINEAR	خطی	LOADER	بار کننده	LOOP	حلقه
LINEAR CODE	برنامه خطی، برنامه بدون حلقه	LOCAL	محلی، موضعی	LOOP BODY	بدنه حلقه
LINEAR PROGRAMMING	برنامه ریزی خطی	LOCAL AREA NETWORK	شبکه محلی	LOOP INVARIANT	ثابت حلقه
LINEAR SEARCH	جستجوی خطی	LOCAL VARIABLE	متغیر محلی	LOOSELY COUPLED	با اتصال ضعیف
LINGUISTICS	زبان شناسی	LOCALITY	محلی بودن	LOW-LEVEL	سطح پائین
LINK	پیوند	LOCATE	محل یابی، مکان یابی	LOW-LEVEL SCHEDULER	زمانبند سطح پائین
LINKABLE	پیوند پذیر	LOCATION	محل، مکان	LOW-ORDER	پائین رتبه
LINKAGE EDITOR	ویراستار پیوندی	LOCATION COUNTER	محل شمار، مکان شمار	LOWER BOUND	کران پائین
		LOCK	قفل کردن، قفل		

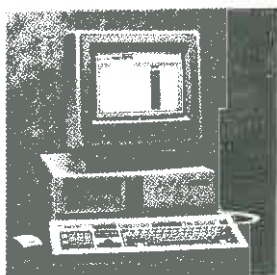
سخنرانی علمی اسفند ماه

منطق غیر صوری و نظریه پیچیدگی

سخنران : آقای دکتر ضیاء موحد



زمان : چهارشنبه ۱۴ اسفند ساعت ۴/۵ بعد از ظهر
مکان : گروه ریاضی دانشکده علوم دانشگاه تهران



EVEREX

EVER for EXcellence®

کامپیوتر EVEREX ساخت آمریکا سازگار با IBM . با تضمین و پشتیبانی کامل
دارای رتبه بهترین کامپیوتر از مجله Info World در سال ۱۹۹۱ .

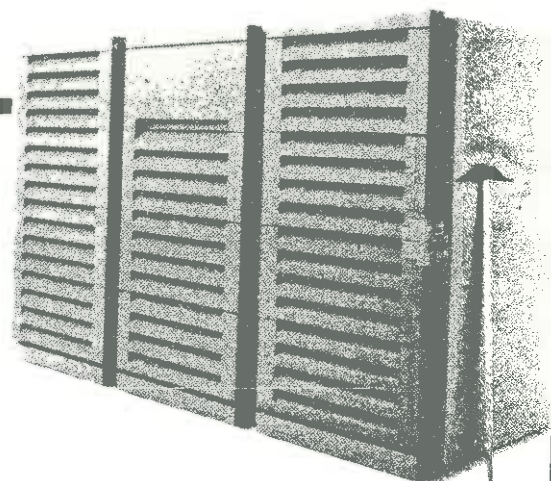


نگاره

شرکت کامپیوتری نگاره نماینده انحصاری کامپیوترهای EVEREX در ایران
تهران - خیابان استاد مطهری - خیابان جم (فجر) - خیابان تابنده (شهید لطفی) - کوچه بهرامی - پلاک ۷
تلفن : ۸۲۸۶۲۸ - ۴۴۰۸۴۵۰ تلکس : ۲۱۲۴۴۱ فاکس : ۸۳۵۹۹۷ صندوق پستی : ۱۵۸۷۵/۱۴۱۴ تهران

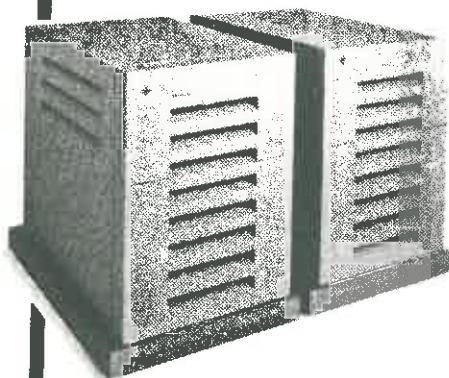


sun microsystems



بدعت گزار در صنعت انفورماتیک جهان

- تکنولوژی و طراحی برتر
- سیستم عامل UNIX بر اساس پردازنده RISC/SPARC
- پیشرو در شبکه بندی OPEN SYSTEMS
- پیشرو در سیستم های CLIENT/SERVER
- پیشرو در سیستم های کامپیوتری به لحاظ PRICE/PERFORMANCE
- پردازنده مرکزی SPARC با طراحی RISC با سرعت 110 MIPS و امکان پردازش موازی (PARALLEL PROCESSING)
- پیشرو در سیستم های توزیعی (DISTRIBUTED)
- OPEN LOOK واسطه گرافیکی کاربران (GRAPHIC USER INTERFACE) برای استفاده ساده از کامپیوتر
- بیش از ۳۸۰۰ بسته نرم افزاری برای کاربردهای علمی / مهندسی / ...
- و اداری / مالی



Sun تولید کننده سیستم های کامپیوتری سوپر مینی :

microsystems

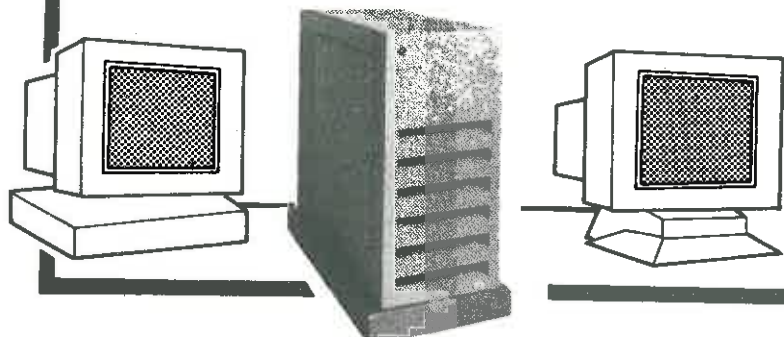
- ۱ - ایستگاه کار (WORKSTATION) برای کاربردهای ENGINEERING
 - ۲ - SERVER برای محیط های CLIENT/SERVER , ELECTRONIC PUBLISHING, MANUFACTURING, SCIENTIFIC و ...
 - ۳ - چند استفاده کننده (MULTIUSER) برای کاربردهای آموزشی ، اداری ، DATA BASE SERVER WORKGROUP COMPUTING و
- مالی و با قابلیت اتصال ۳۶ تا ۱۳۰ ترمینال



شرکت نگاره عرضه کننده و پشتیبان سیستم های کامپیوتری کمپانی
SUN MICROSYSTEMS در ایران .

تهران - خیابان استاد مطهری - خیابان حم (فجر) - خیابان ناصبه
(شهید لطفی) - کوچه بهرامی - پلاک ۷

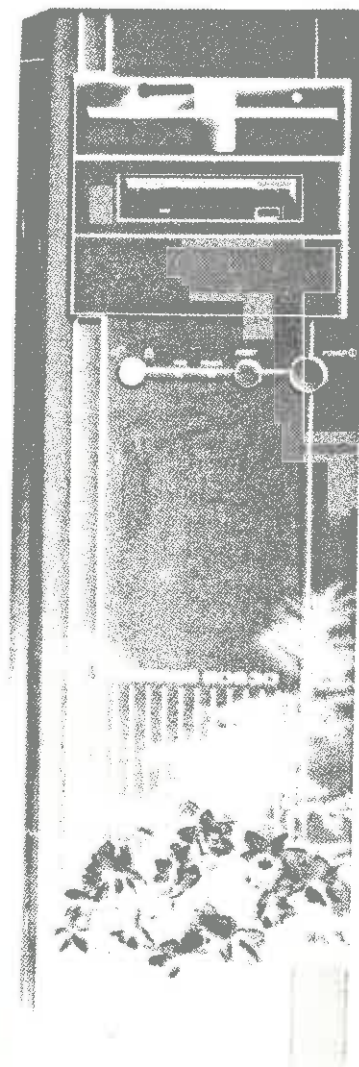
تلفن: ۸۳۸۶۲۸ - ۴۴۰۸۴۵۰ تلکس: ۲۱۲۴۴۱
فاکس: ۸۳۵۹۹۷ صندوق پستی: ۱۵۸۷۵/۱۴۱۴





پویا

طراحی و ساخت سیستمهای کامپیوتری



خیابان قائم مقام فراهانی، کوچه بیستم، شماره ۶۲، صندوق پستی: ۵۷۴، ۱۴۳۳۵، تهران

تلفن: ۸۶۶۶۷۹-۸۶۳۳۶۷-۸۶۱۷۴۲-۸۲۹۲۳۵-۸۲۸۳۲۰

فاکس: ۸۶۱۸۵۸

