

Flash informatique

p/a Service informatique central, C.P. 121, CH-1015 Lausanne, ☎ 021/693 22 47 - N° 7 DU 29 SEPTEMBRE 1992

Le Service Informatique du Département d'Informatique: le SIDI

PAR ROLAND WUILLEMIN, SIDI

Cet article présente les activités du Service Informatique du DI et les moyens informatiques collectifs mis à disposition des étudiants pendant leurs études. Les étudiants proviennent des sections d'Informatique, de Systèmes de Communication et de plusieurs autres sections pour les cours d'informatique de base. Afin de rester assez général, seules les grandes lignes d'activités seront abordées, le but de cet article n'étant pas de présenter tous les travaux réalisés par l'ensemble des laboratoires du département ni leurs propres équipements.

Les activités du SIDI sont essentiellement:

- Maintenir tous les équipements en fonctionnement et à un bon niveau de performances
- Faire la gestion des comptes des utilisateurs
- Acheter et installer les logiciels et leurs révisions
- Élaborer les budgets et les cahiers des charges pour l'achat des nouveaux équipements
- Assurer la coordination des services informatiques de base entre les différents ordinateurs
- Participer aux différentes commissions de coordination des moyens informatiques.

Les équipements collectifs sont :

- deux salles de Macintosh avec un serveur;
- une salle de terminaux reliés au VAX 8600;
- une salle de stations reliées à deux serveurs SUN.

Le SIDI comprend cinq personnes, quatre responsables pour les différents systèmes et une secrétaire à temps partiel. Un des membres est délégué à la gestion des laboratoires banalisés du LIT et du LSL. Avec cet effectif, il doit assurer une permanence pendant les périodes de vacances ou de service militaire.

LES SALLES MACINTOSH CO 020 ET CO 023

Ces deux salles ont été ouvertes dans le bâtiment de la Coupole en octobre 1990.

La configuration des Macintosh est:

- système d'exploitation: Mac OS 6.0.7;

- mémoire RAM: 5 MBytes au moins;
- disque interne: 40 MBytes;
- serveur: Macintosh II Cx, RAM de 8 MBytes, disque interne de 80 MBytes.

L'utilisation de ces deux salles se divise essentiellement en 2 activités:

>>> Suite en page 8

Dans ce numéro

Le Service Informatique du Département d'Informatique ROLAND WUILLEMIN	1
Editorial MARIE-CHRISTINE SAWLEY	2
SIC-INFO	2
Recyclage du matériel infor- matique PATRICE PITTORI	3
	
Le contrôle du Tokamak TCV J.-B. LISTER, Ph. MARMILLOD, S. BERNEL, P.-F. ISOZ & B. MARLETAZ	5
LabVILEW JEAN-JACQUES DUMONT	8
Le point sur les modems JACQUES VIRCHAUX	11
C++, version 3.0 FRANCIS LAPIQUE	13
CROSSCUTS MARCO TOMASSINI	16
Cours pour les utilisateurs du Nec SX-3 du CSCS MARCO TOMASSINI	16
Formation JOSIANE SCALFO	18
Calendrier	20

éditorial

Le Flash Informatique vous propose dans cette édition automnale quelques articles de fond et d'information, reflétant l'actualité de certaines activités informatiques à l'Ecole.

Tout d'abord, Roland Willemin vous fait mieux connaître les équipements et services proposés par le SIDI (Service informatique du DI qu'il dirige), et vous donne en primeur une idée de l'évolution prévue à court et moyen terme. Evolution axée sur un concept d'informatique décentralisée et distribuée, dont l'intégration dans le cadre de l'Ecole compose une palette de services de base et plus spécialisés. Cet article contribue à faire connaître les fonctions trop méconnues à l'Ecole de Délégué départemental à l'informatique et de Service informatique de département. Après le DMA et le DI, nous sommes sûrs que d'autres départements emboîteront le pas.

Dans le domaine de l'instrumentation, le CRPP nous explique ses choix en matière de logiciel de contrôle du TCV (Tokamak à configuration variable). Pour ceux qui ont choisi la plate-forme Macintosh pour l'instrumentation et le pilotage de leurs équipements de mesure, Jean-Jacques Dumont propose le point sur LabVIEW et les convie à une journée de présentation et d'atelier consacrée à ce produit. Notre collègue Jean-Jacques envisage de constituer un ensemble de logiciels et d'applications de base pour ce public, à l'instar de ce qui se fait pour la plate-forme PAO et bureautique bien implantée à l'EPFL. Les intéressés trouveront certainement des articles sur ce sujet dans les numéros à venir.

Recyclage: préoccupation et phénomène de société. Patrice Pittori vous fait partager ses connaissances et interrogations en matière de recyclage du matériel informatique, et se propose de faire le point dans les prochains numéros sur les mesures pratiques prises à l'EPFL: sujet important et complexe, à suivre donc...

Le point sur les modems, par Jacques Virchaux: voilà ce qui ne manquera pas d'intéresser ceux, et ils sont nombreux, qui se trouvent parfois confrontés aux problèmes de normes et de performances de ces équipements dont l'offre a fait un bond considérable ces derniers temps.

A destination de ceux qui se sont laissés tenter par la programmation orientée objet, Francis Lapique propose une revue approfondie des points saillants de la nouvelle version du langage C++, la troisième dans la lignée.

Une naissance au SIC! Dès le mois d'octobre, un nouveau service sera offert à l'Ecole: une salle de cours, équipée de 9 stations Unix Sun IPX sera mise sur pied; rendez-vous dans les pages *Formation* si vous souhaitez en savoir plus. Vous y trouverez comme toujours le programme complet des cours MAC et PC offerts au personnel de l'Ecole. Parmi les annonces du SIC l'arrêt de *sicsun*, le point sur les problèmes de *nestor*, méritent tout particulièrement votre attention si vous êtes utilisateur de ces

serveurs centraux.

Un journal comme le nôtre n'est vivant que par la matière que vous (lecteurs et rédacteurs potentiels) y apportez. Si vous souhaitez faire partie de la rédaction du Flash Informatique en 93, si vous avez des idées sur les sujets ou les orientations que vous aimeriez voir apparaître dans ces pages, nous vous invitons à nous rejoindre lors d'une prochaine réunion du Comité.

A tous et à toutes, bonne lecture et bonne rentrée.

Marie-Christine SAWLEY

Rédacteur en chef du Flash Informatique

sic - info

SERVICE DE LA PAT: UN PETIT RAPPEL

En cette période de rentrée automnale, nous souhaitons refaire le point sur les différents moyens de contacter la PAT, Permanence d'aide technique du Service informatique central.

Comment joindre le service de PAT?

- par messagerie électronique: pat@sic.epfl.ch, ou sur QuickMail
- par téléphone: (693) 2209. Depuis le 21 août dernier, l'Ecole s'est dotée d'un central téléphonique digital dont vous avez certainement déjà testé les nouvelles possibilités. Le numéro de la Permanence est doté d'une boîte vocale. N'hésitez pas à nous laisser votre message, la boîte est régulièrement relevée. Un rappel: les personnes nous appelant depuis le site de l'EPFL ont la possibilité de déclencher le bip en composant désormais le 181 avant le numéro interne. Les heures de

➤ ➤ ➤ suite en page 17

Flash Informatique

Les articles de ce journal ne reflètent que l'opinion de leurs auteurs. Toute reproduction, même partielle, n'est autorisée qu'avec l'accord de la rédaction et des auteurs.

Rédacteur en chef : Marie-Christine Sawley
Comité de rédaction : Jean-Daniel Bonjour, Jean-Michel Chenaïs, Nathalie Colombi, Milan Crvcnin, Jacqueline Dousson, Pierre-André Haldy, Véronique Jost, Pierre-Jean Paris, François Roulet & Jacques Virchaux
Composition : Appoline Raposo de Barbosa

Impression :



Tirage : 4000 exemplaires
Adresse: SIC-SA EPFL 1015 - Lausanne
☎ 021/693 22 42 & 22 47
Prochaine parution : 27 octobre 1992
Délai de rédaction : 9 octobre - 12h00

Recyclage du matériel informatique

PAR PATRICE PITTORI, SIC-ETAT-MAJOR

LA RÉCUPÉRATION, LE RECYCLAGE ET LA PROTECTION DE L'ENVIRONNEMENT

Trois parmi les grands sujets de préoccupation de notre fin de siècle! Le grand public est relativement bien informé de ce qui se fait actuellement pour ses biens de consommation quotidiens: verre, aluminium, papier, PET, huiles, matériel électroménager, voitures, etc.

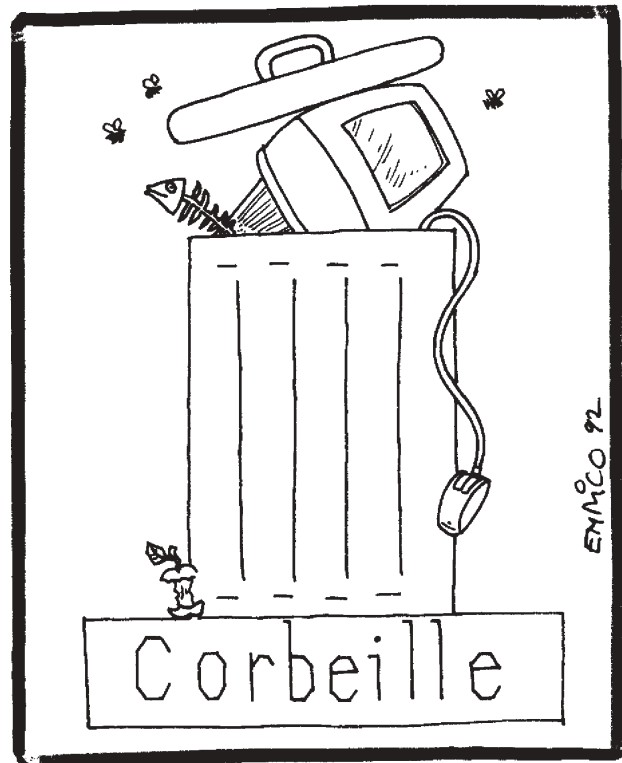
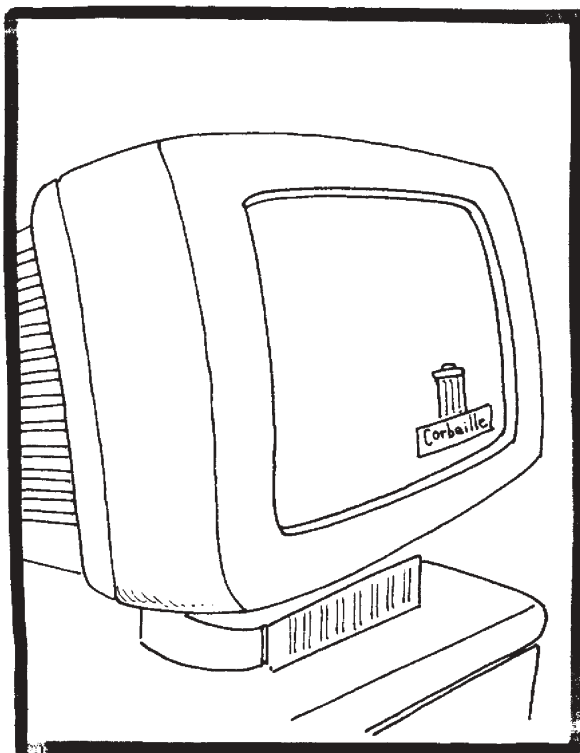
Mais sait-il (ou se demande-t-il) ce que deviennent ces machines qui ont fait leur intrusion dans sa vie professionnelle et privée ces trois dernières décennies et qui représentent un volume de matières industrielles toujours plus monstrueux? Jugez-en vous-mêmes:

- selon les estimations officielles, en Suisse la montagne de déchets électroniques, qui représente actuelle-

LA RÉCUPÉRATION

Cette méthode d'élimination *archaïque* consistant en simple ferrailage et incinération des déchets non-ferreux ne peut plus servir de base à un recyclage judicieux: en effet, la combustion des tubes cathodiques d'écrans, des circuits imprimés contenant des métaux lourds et des plastiques renfermant parfois du brome, de la dioxine et du furane, dégagent des substances hautement toxiques, dont les effets nocifs pour l'environnement sont aujourd'hui parfaitement connus - rappelez-vous Seveso de triste mémoire!

Comme la plupart des composants électroniques, ces produits usagés contiennent une certaine proportion de métaux lourds et d'additifs dont l'élimination non appro-



ment quelque 4'000 tonnes par an, grossira d'ici l'an 2000 pour atteindre 40'000 tonnes (10 fois plus!).

En outre, une part importante de ces déchets est dirigée vers les installations d'incinération des ordures!

L'industrie informatique connaît de telles poussées innovatrices que les nouveaux produits se suivent à intervalles toujours plus courts et sont toujours plus performants et avantageux. Par conséquent, les systèmes informatiques sont plus vite démodés de nos jours qu'il y a quelques années seulement et le volume des appareils usagés ne cesse d'augmenter.

priée peut nuire à l'environnement:

- les déchets électroniques contiennent jusqu'à 23 métaux précieux ou composites, ce nombre étant fonction de l'âge de l'ordinateur. Ainsi, à partir d'une tonne de déchets de plaques classiques (boards), il est possible de recycler entre 100 et 200 kg de cuivre, 1 kg d'or, 100 à 300 grammes de palladium et 40 grammes de platine.

Ces chiffres expliquent pourquoi les utilisateurs de matériel informatique pouvaient *vendre* leurs installations usagées à des ferrailleurs. Malheureusement, ceux-ci agissent souvent plus par souci économique

qu'écologique et les considérations de protection de l'environnement sont reléguées à l'arrière-plan par les perspectives de gain. Or, recycler ne signifie pas prélever les métaux précieux, puis incinérer ou remettre le reste des déchets à la décharge. Il s'agit davantage de **réintroduire** les matières premières dans le cycle de production, avec la même qualité et sans porter préjudice à l'environnement. Un constructeur informatique conscient de ses responsabilités ne peut donc confier l'élimination des déchets à un partenaire animé par des motivations discutables.

La récupération du matériel informatique joue dans ce contexte un rôle croissant pour les entreprises et les administrations utilisatrices de ce matériel. Dans un esprit de responsabilité vis-à-vis de l'environnement, de plus en plus de clients des constructeurs informatiques s'interrogent avec raison, parfois même avant l'achat d'un système déjà, sur le devenir de celui-ci à la fin de son cycle de vie.

Par bonheur, les utilisateurs ne sont pas les seuls à se poser des questions et à s'enquérir sur les débouchés possibles pour leur *vieux* matériel. En effet, les grands constructeurs et leurs filiales ont décidé depuis quelques années de passer à l'action, en mettant sur pied un service de récupération organisé comprenant le transport du domicile du client à l'atelier de démontage et la livraison au ferrailleur des matières, qui ne peuvent pas être *recyclées* telles quelles dans la chaîne de production des nouveaux produits ou comme pièces détachées de réserve destinées à d'anciens équipements encore opérationnels en clientèle. Actuellement, cet intérêt pour la protection de l'environnement amène même certains fabricants de matériel électronique, constructeurs et sous-traitants, à concevoir à la base des produits protégeant eux-mêmes l'environnement, grâce à un **recyclage** facile et le moins polluant possible.

LE RECYCLAGE INFORMATIQUE - UNE NÉCESSITÉ POUR L'ENVIRONNEMENT

En collaboration étroite avec leurs propres usines de production et leurs centres logistiques, ainsi qu'avec des entreprises spécialisées dans la récupération et des conseillers en écologie, les principaux constructeurs de la planète ont analysé tous les problèmes de recyclage cruciaux et mis en place un concept de récupération orienté vers l'avenir (!). Des entreprises de traitement domiciliées en Suisse alémanique et en Suisse romande traitent les appareils pour en récupérer les matériaux à la composition aussi homogène que possible et les réintroduire dans le cycle des matériaux.

Une technologie de recyclage de pointe permet aujourd'hui déjà de recycler plus de 80% des matériaux d'un ordinateur; dans un proche avenir on parle même de 90%! Le fer et les métaux lourds non-ferreux sont soumis au processus de récupération ordinaire des mitrailles. Les divers types de synthétiques permettent d'en extraire des matières brutes secondaires à partir desquelles sont produits des matériaux industriels, tels que parois insonorisantes, des récipients ou des revêtements de sols.

Les tubes cathodiques sont récupérés, leurs pièces métalliques triées, la couche toxique en est laminée puis neutralisée selon des processus mécano-chimiques. Le verre du tube d'écran est ensuite introduit dans le cycle de récupération du verre.

PLANIFIER PLUTÔT QU'ÉLIMINER

A l'exemple d'une grande firme allemande de voitures, certains constructeurs informatiques tels que DEC, IBM et SIEMENS-NIXDORF, pour ne citer que les principaux, ont développé une ingénierie de recyclage spéciale, afin d'assurer la prise en compte d'aspects écologiques importants dans le développement des nouveaux produits, ainsi que de leur emballage.

Les matériaux sont utilisés avec parcimonie et les substances présentant un risque pour l'environnement sont évitées dans la mesure du possible. En outre, la consommation d'énergie et les émissions sont réduites au minimum. L'objectif essentiel de cette politique est de diminuer la production de déchets spéciaux.

L'allongement de la durée de vie d'un produit sert également la protection de l'environnement. Dans cet esprit, les services de logistique des constructeurs, lors de la reprise de vieux appareils, vérifient d'abord si une amélioration technique ne permettrait pas de réutiliser ceux-ci et donc d'allonger leur cycle de vie. Si cette tentative ne s'avère pas concluante, les techniciens procèdent au démontage des pièces et des modules réutilisables et, après vérification, les remontent dans d'autres machines (par exemple en réparation).

CONTRÔLE DE RÉCUPÉRATION - CONTRÔLE ET PROTECTION DES DONNÉES

Le contrôle de récupération mis en place par les constructeurs veille en particulier à ce que les entreprises externes mandatées mènent à bien le recyclage selon les directives qu'elles émettent et en conformité avec les dispositions légales. Ces firmes travaillant sous contrat font l'objet de contrôles périodiques de la part des constructeurs et d'organes professionnels, tels que la Fondation suisse de l'élimination des déchets et la Société suisse pour le traitement des déchets spéciaux.

LOGISTIQUE DE RÉCUPÉRATION

La logistique de récupération doit, elle aussi, réaliser l'équilibre entre les mesures requises et souhaitables du point de vue écologique et les réalités techniques et économiques. Certains constructeurs intègrent les coûts de la récupération à la prestation. Cette pratique s'harmonise avec les projets actuels d'amendement de la loi fédérale sur la protection de l'environnement et réduit les frais administratifs de façon importante. Les charges liées à la récupération sont ainsi prises en compte à la base, c'est à dire dès le début du cycle de vie du produit.

EXEMPLE DE PRESTATIONS OFFERTES PAR UN CONSTRUCTEUR

La société Digital Equipment, fournisseur de l'EPFL, est l'un des constructeurs dont la filiale suisse assure un service complet à ses clients au niveau de la récupération des matériels usagés, dont les clients veulent se débarrasser lors de l'acquisition de nouveaux équipements. Ce service est facturé à un tarif préférentiel aux clients.

Matériel d'emballage

Toute livraison comprend entre autres le déballage des produits livrés et la récupération immédiate du matériel d'emballage.

Matériel de consommation

Le petit matériel et les fournitures, telles que les rubans encreurs, les cartouches de toner des imprimantes laser, les disques et disquettes, etc. peuvent être retournés au service de récupération de ce constructeur, pour être recyclés. Les clients ne payent que les frais d'envoi.

Pièces défectueuses

Les pièces et les composants défectueux retournés au fournisseur sont réparés autant que possible. Les éléments qui ne peuvent pas être réparés sont systématiquement recyclés.

Matériel de reprise

Lorsqu'ils sont remplacés par de nouveaux produits du même constructeur, les ordinateurs et périphériques usagés peuvent être récupérés à la demande du client. Dans le cas où le client ne remplace pas son matériel usagé par un équipement de la même marque, le constructeur facturera les frais de récupération et de recyclage.

QUELLES SONT LES DIRECTIVES DE LA CONFÉDÉRATION EN LA MATIÈRE ?

L'Office Fédéral de l'Environnement et les EPF ont participé activement au projet de création des organes spécialisés dont il a été question plus haut. Mais il faut reconnaître qu'en Suisse, contrairement à ce qui se passe dans d'autres pays européens, l'Association pour l'élimination des appareils électroniques anticipe les travaux du législateur. Ainsi, les entreprises ont pu, en dehors de tous délais ou contraintes législatives, élaborer une solution sur mesure intégrant tous les impératifs écologiques.

En ce qui concerne les EPF et l'EPFL en particulier, nous aurons l'occasion de vous parler des mesures pratiques en vigueur et de la procédure à suivre dans un prochain numéro du Flash Informatique. ■

Le contrôle du Tokamak TCV

PAR J.B. LISTER, Ph. MARMILLOD, S. BERNEL, P.-F. ISOZ & B. MARLETAZ, CENTRE DE RECHERCHES EN PHYSIQUE DES PLASMAS

D'autres articles de ce journal ont déjà abordé le sujet du Tokamak "Acquisition et commande du Tokamak TCV" (1/90) et "Un multiplicateur matriciel hybride pour le contrôle du plasma dans le Tokamak TCV" (8/90). L'article qui suit décrit le système de contrôle des équipements du TCV.

LES EXIGENCES DU SYSTÈME DE CONTRÔLE DU TCV

Le système de contrôle de l'équipement du TCV (TCV-CS) doit fournir les fonctionnalités suivantes:

- une interface homme-machine concernant tous les aspects opérationnels,
- un contrôle automatique de l'équipement,
- l'enregistrement de l'état actuel dudit équipement,
- une interface avec d'autres logiciels,
- le contrôle du cycle de tir.

Le TCV-CS agit donc au niveau du comportement cyclique des opérations de tir du tokamak et à celui du contrôle continu de tout l'appareillage. Une partie de ce dernier est autonome et ne requiert que de très simples opérations de contrôle, alors que d'autres nécessitent un

suivi *bit à bit* au niveau de l'opérateur. Certains équipements -comme les alarmes ou l'état de préparation d'un tir- doivent être fréquemment contrôlés (c'est-à-dire au moins une fois par seconde), tandis que d'autres ne le sont que quelques fois par heure.

Lors de l'implémentation du TCV-CS, une première décision a été de n'accéder à l'équipement que via la communication BITBUS. Une seconde fut d'implémenter les fonctions de contrôle et celles des acquisitions de données sur un unique système informatique, ce qui évite tout problème de compatibilité entre elles. Sur d'autres installations, elle sont souvent séparées, mais il nous a semblé plus prudent qu'elles ne le soient pas. Le choix du système d'acquisition de données s'est porté sur MDSplus et a conditionné celui du système de supervision, VSYSTEM, de sorte que toute la gestion du TCV utilise le même software X-window et tourne sur un cluster de stations VAX de Digital.

VSYSTEM fournit un ensemble d'outils de supervision qui permettent à l'utilisateur de créer (i) une base de données partagée sur plusieurs machines en temps réel, (ii) une visualisation des équipements via X-window, (iii) des séquences de contrôle grâce à un langage de

programmation qui lui est propre, et (iv) des outils d'archivage de l'évolution de l'équipement.

La couche d'accès de bas niveau vers l'équipement, les sorties à l'écran, les séquences et la base de données sont générées par l'utilisateur le plus familiarisé avec l'installation concernée, puisque, pour des raisons de performance, l'essentiel de ces parties de software sont spécifiques à cette dernière.

LA COUCHE D'ACCÈS VERS LES ÉQUIPEMENTS

La couche d'accès à l'équipement (CAE) doit fournir une communication entre les bases de données du TCV-CS, qui représentent le TCV à un niveau abstrait, et la collection d'appareils hétérogènes qui le constituent réellement, à savoir:

- le turbogénérateur de 340 MJ,
- des redresseurs de courant de 250 MVA,
- enclenchement de la puissance électrique,
- contrôle des accès
- refroidissement des bobines
- chauffage de la chambre à vide
- contrôle des pompes à vide
- diagnostics
- Gyrotron 39 GHz
- etc.

Certains équipements sont des appareils munis de leur propre protocole RS232. Les autres sont construits autour des noeuds BITBUS sur lesquels nous avons un contrôle de leur programmation. Un type de nos cartes BITBUS offre des ports RS232. Un «répéteur» software transparent de ce noeud BITBUS particulier retransmet le message à un RS232 quelconque. La CAE ne nécessite donc qu'un service de message au BITBUS au niveau le plus bas, bien qu'il existe malheureusement une large palette de liens RS232.

La CAE est structurée en quatre couches distinctes:

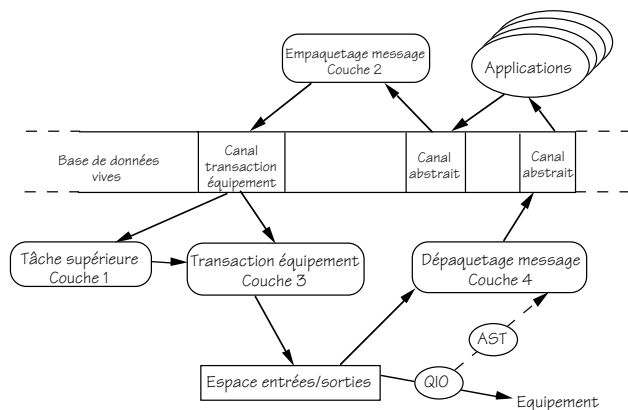


Figure 1 — Schéma de principe des couches d'accès aux équipements

1. la couche supérieure qui sait initier une communication avec des appareils,
2. la couche d'empaquetage des données qui crée des messages pour l'appareillage à partir de plusieurs canaux des bases de données,
3. la couche de transaction avec les appareils, qui envoie

- les messages —empaquetés dans la couche 2— aux appareils concernés, et
4. la couche de dépaquetage, qui dépaquette les messages des appareils en canaux de la base de données, soit la fonction opposée à la couche 2.

Pour l'empaquetage, comme pour le dépaquetage, il existe bien sûr une variété sans fin de messages différents imaginables; et en implémentant quatre algorithmes génériques seulement nous avons couvert tous les cas rencontrés sur TCV.

Les quatre couches brièvement décrites ci-dessus nous donnent la flexibilité nécessaire au contrôle de tous nos instruments. Pour y ajouter de nouvelles installations, il suffira de simplement définir de nouvelles structures à l'intérieur de l'ensemble des bases de données, sans pour autant devoir ajouter un cinquième type de empaquetage/dépaquetage au niveau des couches inférieures.

LE CONTRÔLE DES ÉQUIPEMENTS

La CAE permet aux bases de données de communiquer en temps réel avec les équipements et vice-versa. En d'autres termes, une variation de l'un entraîne immédiatement celle de l'autre. Cela étant acquis, le contrôle des instruments physiques devient équivalent à celui des valeurs de la base de données, ce qui est simple et constitue le travail de VSYSTEM. Ainsi la tendance à augmenter la complexité du système de contrôle avec l'addition de nouveaux appareils est évitée.

Une première forme «manuelle» de contrôle se présente à l'opérateur sous la forme d'une fenêtre générée par les outils graphiques de VSYSTEM. La figure 2, par exemple, donne d'une part un compte-rendu facilement lisible de l'état actuel de l'installation de refroidissement des bobines du TCV. D'autre part, elle offre à l'opérateur un certain nombre de boutons sur lesquels ce dernier peut «cliquer» pour indiquer un choix (ici celui du débit de l'eau ou une demande pour une information plus détaillée). Chacun de ces boutons modifie un canal dans la base de donnée, modification qui se répercute au niveau de la séquence de contrôle automatique, puis à celui des appareils.

Cependant, un système comme celui du refroidissement des bobines doit essentiellement pouvoir fonctionner sans surveillance. C'est à ce niveau que la séquence de contrôle automatique intervient. Elle pourrait se trouver soit au niveau local des noeuds BITBUS, soit au niveau global de l'opérateur. En l'occurrence, l'absence de protocole d'esclave à esclave interdit l'automatisation de sous-systèmes complexes qui incluent plusieurs noeuds BITBUS. C'est donc au niveau du superviseur que tournent les séquences d'automatisation du fonctionnement des appareils, où existent des langages évolués plus commodes à utiliser pour celui qui produit lesdites séquences. Le risque de perdre la communication entre le niveau local et celui de l'opérateur existe, bien sûr. Et c'est pourquoi un système d'*hommes morts* a été conçu qui interrompt toutes les opérations en cours au niveau local s'il ne reçoit pas régulièrement un message standard de la part du niveau global.

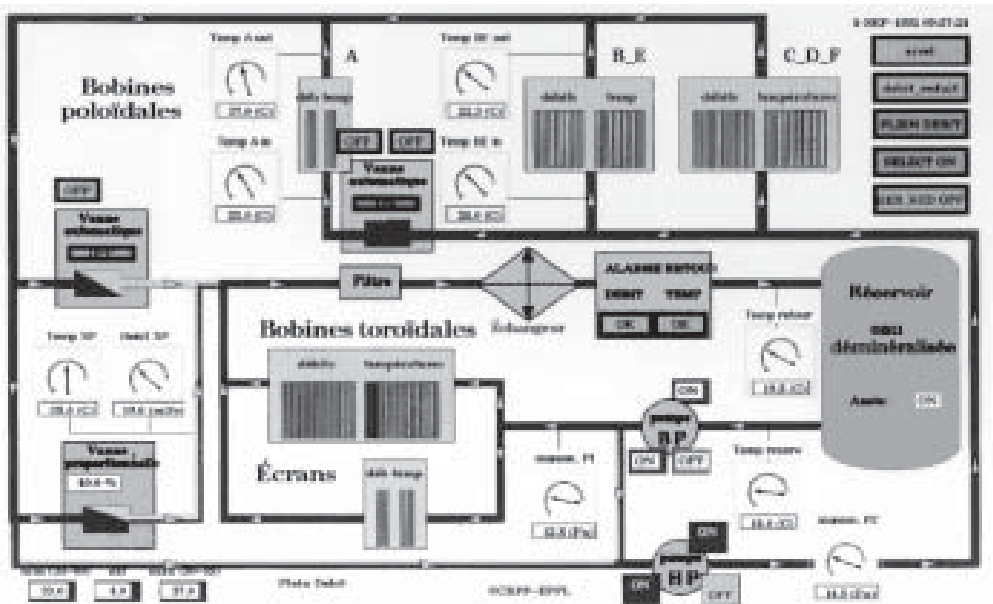


Figure 2 — Synoptique du système de refroidissement

Il ne manque plus alors que quelques canaux dans les bases de données pour contrôler le mode de fonctionnement des séquences elles-mêmes ou pour déterminer les paramètres des boucles de contrôle, par exemple.

Les systèmes asservis au contrôle automatique sont les suivants:

- la sécurité des accès à l'enceintes du tokamak,
- le refroidissement des bobines,
- le chauffage de la chambre à vide,
- etc.

La génération et la visualisation des séquences d'automatisation a nécessité une certaine attention. Nous avons dès le départ imposé un vocabulaire et des schémas standards pour l'écriture des «protocoles d'opération», afin de conserver une certaine homogénéité dans l'ensemble du système de contrôle malgré la multiplicité des

Elles seront adaptées au fur et à mesure que les particularités de chaque sous-système seront mieux comprises et maîtrisées.

CONCLUSION

Les logiciels actuellement proposés sur le marché évoluent trop vite et sont trop performants pour qu'un groupe de recherche comme le nôtre puisse imaginer en développer un qui soit comparable et surtout mieux adapté à nos besoins. Nous avons donc acheté un système de supervision pour le TCV-CS et n'avons eu qu'à produire la couche d'accès aux équipements, les bases de données, les séquences d'automatisation et les sorties à l'écran. Une telle politique nous a permis de concevoir le contrôle complet de l'installation avec globalement moins

d'un physicien employé à plein temps. Par ailleurs, l'évolution constante des produits achetés nous permet de toujours rester «à la page».

L'avantage majeur d'une telle approche est donc la possibilité qui est offerte à l'utilisateur de définir facilement sa propre interface de visualisation et de contrôle de l'équipement dans un environnement capable de s'adapter rapidement aux divers changements à venir. ■

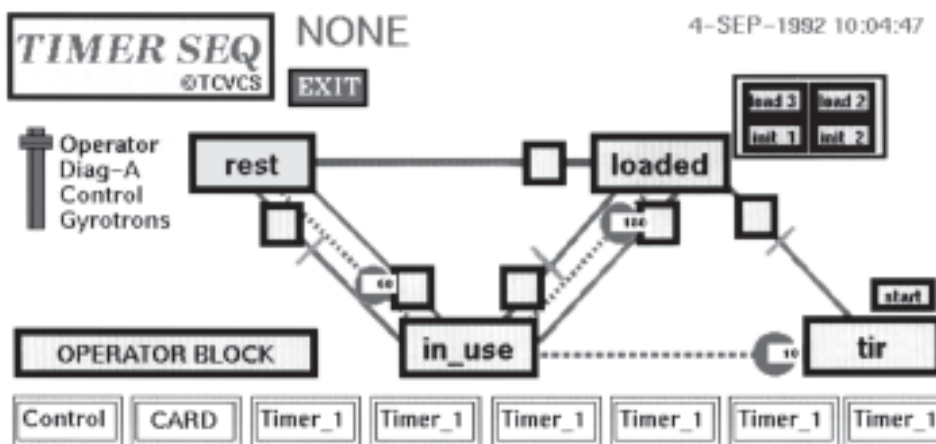


Figure 3 — Visualisation de l'état des séquences du système

LabVIEW

PAR JEAN-JACQUES DUMONT, SIC-LOGICIELS

Il fut un temps où les PDP-11 régnaient sur ce domaine.

Depuis quelques mois, il semble bien que les Macintosh habillés du logiciel LabVIEW, de la société National Instruments, se taillent la part du lion dans les labos de l'EPFL dès qu'il s'agit d'interagir avec les instruments de test, de contrôle ou de mesure. Ce logiciel permet en effet de combiner acquisition, traitement et présentation des données, évitant ainsi le stockage intermédiaire de celles-ci; une caractéristique bien utile pour une analyse en temps réel par exemple.

Le projet LabVIEW a en fait débuté en 1983 déjà, avec l'élaboration d'un concept d'instrument virtuel; ce que l'on appellerait peut-être aujourd'hui, en utilisant les termes à la mode, une API orientée-objet. Les instruments en question étaient du type basé-message, comme les oscilloscopes GPIB, avec une évolution vers le type basé-registre, comme les numériseurs VXI.

Le résultat fut l'apparition en 1986 de LabVIEW 1 sur les Macintosh, cet engin inhabituel ayant été choisi essentiellement en fonction de son interface graphique révolutionnaire.

Depuis, il y eut LabVIEW 2 en 1990, toujours sur Macintosh; un produit complètement réécrit en fonc-

tion de l'évolution des technologies et des expériences faites avec la version 1; celui-là même qui fait actuellement l'objet d'une large distribution sur le site de l'EPFL. Le produit inclut désormais divers modules pour le traitement de signaux, le filtrage digital, l'analyse numérique, l'algèbre matricielle et l'analyse statistique, couplés avec les modules de contrôle des appareils et d'acquisition des données. Ces dernières transitent donc directement, sans problème de reformattage entre les modules. Le produit offre également des possibilités de programmation graphique simplifiant grandement la construction d'applications.

La nouvelle de cette rentrée 92, c'est l'arrivée—attendue—de LabVIEW sur PC/Windows, dans une version dénommée 2.5 aussi proche que possible de la version Macintosh. Cette version remplace avantageusement l'ancien LabWindows qui ne tournait que sur DOS, une plateforme désormais éculée pour des applications où le graphique joue un si grand rôle.

La seconde nouvelle - oh surprise - c'est que ce même LabVIEW 2.5 tourne également sur les stations Sun! Sur Unix, si, si!

Ce portage est en fait le premier résultat d'une alliance stratégique entre Sun Microsystems, Tektronix et

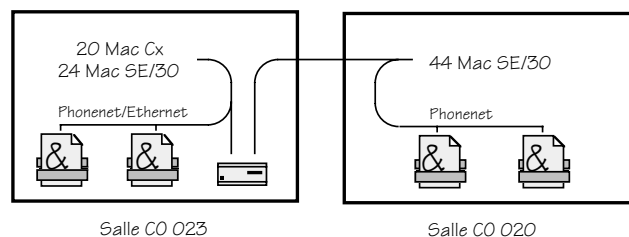
National Instruments visant à créer un environnement intégré pour les tests et mesures basé sur les systèmes ouverts et les standards industriels (GPIB, VXI). Tiens, ne serait-ce pas là justement le créneau favori d'un autre grand constructeur, dont les initiales seraient H et P?

Vous êtes curieux de voir ce que cela peut donner? Pas de problème: une présentation est prévue au SIC le 11 novembre prochain (matin, horaire à définir). La démo sera suivie d'un mini-workshop (après-midi) ayant pour but essentiel de tisser un réseau de relations entre les utilisateurs actuels et futurs de LabVIEW à l'EPFL - et pourquoi pas à l'UNIL, dans la foulée... Avis donc à toutes les personnes désireuses de participer plus ou moins activement à cette première initiative, puis éventuellement à l'élaboration d'un groupe plus structuré. Pour plus de renseignements sur ces opérations, prière de contacter le sous-signé, e-mail dumont@sic ou téléphone 2224 (avec boîte vocale, s'il vous plaît!). ■

NDR: sur le même sujet, nous vous rappelons l'article *Labview, un système d'acquisition prêt à porter* de Monsieur A. Fasoli, dans le FI 4/92.

LE SERVICE INFORMATIQUE DU DÉPARTEMENT D'INFORMATIQUE: LE SIDI (suite de la page 1)

- Les exercices de programmation liés aux cours donnés aux étudiants de première année en Informatique, Mathématiques, Électricité, Microtechnique, Mécanique et Physique. Le langage enseigné est le PASCAL (compilateur THINK PASCAL).
- Les exercices de programmation liés aux cours donnés aux étudiants du deuxième semestre en Mécanique et Physique. Le langage enseigné est le FORTRAN (compilateur MAC-FORTRAN II pour MPW).



Le serveur est utilisé pour la gestion des salles (distribution des systèmes et des logiciels) et pour la distribution des exercices: les étudiants y trouvent les données des exercices à effectuer, puis ils y déposent les programmes qu'ils ont écrits.

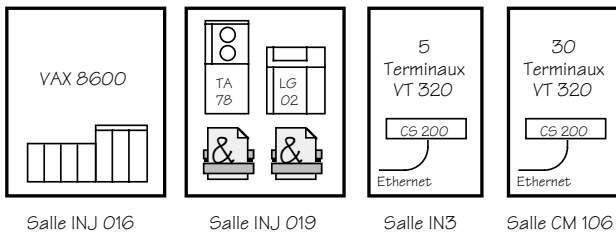
LE VAX 8600 ET LA SALLE DE TERMINAUX

Ce VAX a été transféré du DMA au DI dès la date de création de ce dernier en janvier 1988.

Sa configuration est:

- *Système d'exploitation:* VMS 5.5;
- *Mémoire RAM:* 48 MBytes;
- *Disques système et applications:* 2 totalisant 1 GBytes;
- *Disques pour les utilisateurs:* 4 totalisant 1,8 GBytes.

L'accès au VAX pour les étudiants se fait à l'aide des terminaux situés dans les salles CM 106 et IN3.

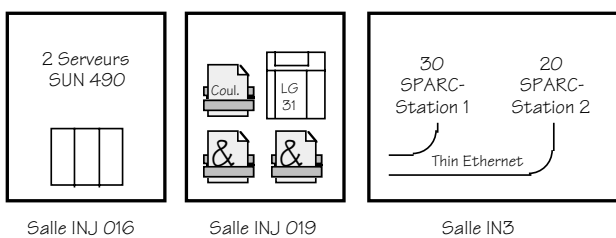


L'utilisation de cet ordinateur se divise essentiellement en 3 activités :

- Les cours de programmation donnés en deuxième année d'Informatique ou les cours donnés par des professeurs du département aux étudiants d'autres sections, ainsi que les projets de semestre. Pour ce faire, les logiciels utilisés sont les langages de programmation PASCAL, ADA, NEWTON et les gestionnaire de bases de données INGRES, ORACLE.
- En tant qu'ordinateur central de département, il assure une grande partie des services informatiques de base (notamment le courrier électronique VMS-MAIL) à l'usage des étudiants et des collaborateurs du DI.
- L'exécution en «batch» de plusieurs programmes de statistique, d'analyse et de recherche opérationnelle par le DMA.

A cause de ses performances maintenant très modestes (il date de 1985) et du manque de possibilités graphiques (seuls les collaborateurs ayant des stations de travail peuvent profiter de l'interface DEC Windows/Motif), l'avenir de cet ordinateur est limité. Il sera remplacé prochainement pour les besoins de l'enseignement par une salle de stations graphiques.

LA SALLE DE STATIONS SUN



A son ouverture en novembre 1989, la salle IN3 comprenait 30 stations de travail. En septembre 1991, son équipement était augmenté de 20 stations supplémentaires. A cette occasion, les deux serveurs SPARCserver 4/390, ont été modifiés pour devenir des SPARCserver 4/490.

La configuration actuelle est :

- *Système d'exploitation:* SUN OS 4.1.3;
- *Mémoire RAM dans les stations:* 16 MBytes;
- *Disque interne dans les stations:* 200 MBytes;
- *Mémoire RAM dans chaque serveur:* 64 MBytes;
- *Disques système et applications:* 2 totalisant 2 GBytes;
- *Disques pour les utilisateurs:* 4 totalisant 4 GBytes;
- Deux robots Exabyte 8 mm de 10 cassettes chacun;

- Une unité de 10 disques optiques réinscriptibles.

L'utilisation de cette salle est en priorité pour les étudiants de troisième et de quatrième années en Informatique et en Systèmes de Communication qui y suivent des cours et qui y font leurs projets de semestre et de diplôme.

Les logiciels utilisés sont:

- *Les langages de programmation:* ADA, C, C++, Fortran, Pascal;
- *Les éditeurs et traitements de texte:* EMACS, Frame-Maker;
- *Les outils graphiques:* Uniras, Interview, Mathematica.

La croissance du nombre d'utilisateurs, donc la charge des processeurs et du réseau interne de la salle, implique chaque année une augmentation de la puissance pour conserver un niveau de performance acceptable. Cette année, deux cartes réseaux et deux cartes d'accélération NFS ont été mises en service. L'année prochaine, les processeurs seront changés pour des superSPARC, ainsi les serveurs deviendront des SS690.

LES SERVICES INFORMATIQUES DE BASE

Les services informatiques de base assurés par les ordinateurs du département sont:

- Le service de courrier électronique:
 - sur VAX, il est assuré par VMS-MAIL, sur SUN par SMTP-MAIL et sur Macintosh par QUICK-MAIL. Le passage de l'un à l'autre est fait à l'aide des passerelles de l'École;
 - de plus, sur SUN, pour les collaborateurs du département, l'introduction d'une adresse unique qui ne dépende plus de l'ordinateur sur lequel ils travaillent est en cours. Ainsi la syntaxe classique «username@nodename.epfl.ch» est remplacée par «firstname.name@di.epfl.ch». L'adresse ne contient plus le nom du noeud, mais celui du département.
- Le service de NEWS sur SUN qui remplit le rôle de serveur pour le département.
- Le service de téléconférence. Entre utilisateurs VAX-VMS, il est assuré par PHONE, entre utilisateurs SUN-UNIX par TALK ou IRC.
- «Time Server» : le VAX et les deux SUN sont synchronisés entre eux et sur une source externe pour créer le serveur de temps du département.
- «Domain Name Server» : un SUN maintient pour le département la liste de tous les noeuds connus sur le réseau de l'École. Un noeud désigne un ordinateur, une passerelle ou un autre dispositif de télécommunication.
- Le service de fichiers qui permet le transfert des fichiers entre les trois mondes VAX, SUN et Macintosh utilise FTP (File Transfert Protocole).

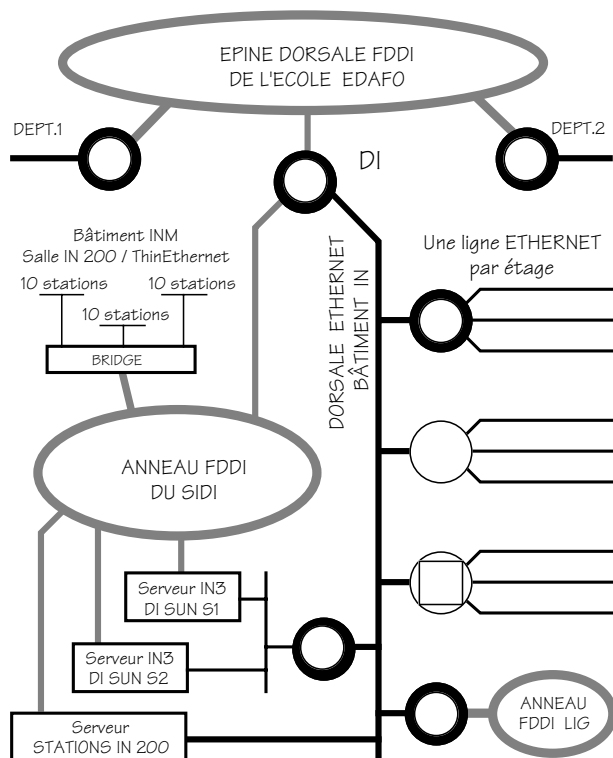
Le SIDI assure aussi la gestion d'un serveur de fichiers Macintosh appelé ARTEMIS à l'attention de tous les collaborateurs. Son disque de 1 GBytes est divisé en plusieurs dossiers : un pour chaque laboratoire.

Dans un proche futur, les services de base implantés sur VAX et sur SUN sont destinés à évoluer vers un

environnement distribué de type DCE (Distributed Computing Environnement) et DME (Distributed Management Environnement). Ce sera le but du serveur SUSP (Systèmes Universels à Services Partagés).

L'ÉVOLUTION FUTURE

Le Département d'Informatique a deux importants projets en vue : le premier est le remplacement du VAX par une nouvelle salle de stations de travail IN 200, le second est l'acquisition d'un serveur SUSP de département dans le but d'intégrer le DI au concept d'informatique décentralisée mis en place par l'École.



La salle de stations IN 200

Cette salle sera située dans le nouveau bâtiment INM dans lequel les cours débiteront cet automne. Pour remplacer le VAX, il est prévu de faire l'acquisition d'environ 30 stations graphiques avec écrans couleurs et d'un serveur de stations. Le système d'exploitation de cet équipement sera conforme à POSIX avec une conséquence importante : il n'y aura plus d'équipement collectif destiné à l'enseignement fonctionnant sous VMS.

A l'heure actuelle, un appel d'offres est en cours. Le planning prévoit la livraison pour la fin de cette année et la mise en service pour le printemps 1993.

Tous les départements sont connectés à l'épine dorsale FDDI de l'École appelée EDAFO. A l'intérieur du département, le réseau comprend, pour chaque étage, une ligne ETHERNET reliée à l'épine dorsale du bâtiment. A cause de la distance entre la salle IN 200 et de l'augmentation du nombre des stations, donc de l'augmentation des transferts de données, le réseau local du SIDI sera un anneau à fibre optique FDDI à 100 Mbits/seconde. Cet

anneau reliera les 3 serveurs et la nouvelle salle. Pour le réseau de la salle, il y aura 3 sous-réseaux Thin Ethernet afin d'éviter la saturation lors des périodes de pointe (telles que les projets de semestre ou les diplômes).

Le serveur de département SUSP

L'acquisition de ce serveur est prévue pour 1993. Il aura deux rôles principaux pour le département :

- Le rôle de serveur de fichiers distribués avec accès homogène depuis toutes les machines clientes distribuées. Il assurera aussi les tâches d'archivage et de sauvegarde des données.
- Le rôle de coordination des services informatiques de base. A ce titre, en plus des nouveaux services prévus, il regroupera à un niveau hiérarchique supérieur ce qui a été installé sur les ordinateurs actuels.

Pour une description détaillée de ces services, on lira l'article de J.J. Dumont «Qu'entend-on en 1992 par services informatiques de base» paru dans le Flash Informatique No 1/92.

CONCLUSION

Le nombre important des ordinateurs mis à disposition des étudiants, la diversité des systèmes et des logiciels peuvent rendre la gestion des équipements difficile. Mais l'application des 3 principes suivants facilite la tâche :

- Utilisation d'outils de gestion puissants
- Évolution continue du matériel et des logiciels avant que des situations critiques ne se produisent
- Formation polyvalente des collaborateurs du service pour qu'ils puissent s'aider, voire se remplacer. ■

Offre d'emploi

Le Centre de Conception de Circuits Intégrés du département d'électricité (LEG/C31) cherche un

Ingénieur électricien ou informaticien

pour participer aux recherches en cours dans le domaine de la modélisation de circuits et systèmes en langage VHDL. Le projet, qui a démarré il y a deux ans, a pour but le développement d'un environnement logiciel intelligent permettant la génération automatique de modèles VHDL à partir de spécifications de haut niveau (machines d'état, chronogrammes, etc...).

Le candidat doit posséder de bonnes connaissances informatiques (Unix, X Window, RPC, C++). Une connaissance de base du langage VHDL est souhaitée.

Entrée en fonction: octobre ou à convenir.

Renseignements et offre de candidature:

Alain Vachoux, ☎ 693 46 06

EPFL-Ecublens, DE-LEG/C31, CH 1015 Lausanne

Le point sur les modems

PAR JACQUES VIRCHAUX, SIC-TÉLÉINFORMATIQUE

Il n'y a pas si longtemps, acheter et utiliser un modem 1200 ou 2400 bps (bits par seconde) était plutôt facile. La plupart de ceux-ci étant *Hayes-compatible*, l'emploi était trivial.

Aujourd'hui, les modems sont non seulement plus rapides mais comportent encore des possibilités de correction d'erreur et de compression des données. Si les performances augmentent, il en va de même de la complexité d'utilisation.

LA MODULATION

Le modem (*MODulator/DEModulator*) convertit les données numériques en signaux analogiques pour les transférer sur une ligne téléphonique. Plusieurs protocoles sont utilisés, définissant la méthode de codage et le transfert des données. En Europe, les protocoles utilisés sont décrits dans la série «V» des recommandations du CCITT.

Les modems à 2400 bps supportent généralement les protocoles suivants:

Bell 103	300 bps, standard US
V.21	300 bps, FSK (<i>Frequency Shift Keying</i>)
Bell 212A	1200 bps, standard US
V.22	1200 bps, DPSK (<i>Differential Phase Shift Keying</i>)
V.22bis	2400 bps et 1200 bps, QAM (<i>Quadrature Amplitude Modulation</i>)
V.23	1200/75 bps, FSK (<i>Frequency Shift Keying</i>)

Si le choix peut très souvent être fait entre le standard US (Bell) et CCITT, il arrive cependant que ces protocoles ne soient pas tous disponibles sur des modèles de bas de gamme. Récemment, certains de ces modems offrent parfois la correction d'erreur, voire la compression des données.

Les modems à *haute vitesse* utilisent des protocoles qui ont été définis plus récemment et qui utilisent la modulation dite en *treillis* TCM (*Trellis Coded Modulation*), permettant d'augmenter encore la vitesse, sans toutefois augmenter la fréquence sur la ligne téléphonique:

V.32	9600 et 4800 bps
V.32bis	14400, 12000, 9600, 7200 et 4800 bps

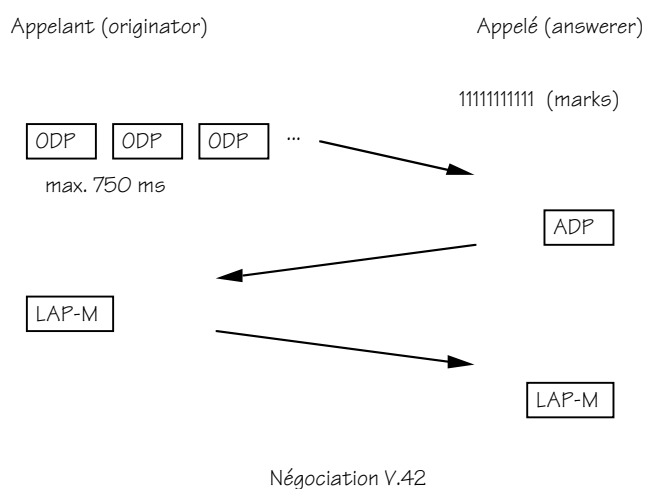
La plupart de ces modems supportent également le V.22bis (2400 et 1200 bps) en plus pour conserver la plus grande compatibilité.

Deux modems ne peuvent communiquer ensemble que s'ils utilisent le même protocole de modulation. C'est le modem appelé (*answerer*) qui fait les tentatives de reconnaissance depuis la vitesse la plus élevée. Le modem appelant (*originator*) se contente de faire la détection appropriée.

LA CORRECTION D'ERREUR

Les premiers modems ont utilisé exclusivement le protocole propriétaire MNP-4 (*Microcom Networking Protocol*). Le protocole V.42 inclut actuellement deux possibilités : le LAP-M (*Link Access Protocol for Modems*) et le MNP-4 en alternative.

L'étape la plus délicate c'est l'établissement de la communication. Etant donné que chaque modem possède ses propres particularités, l'exemple cité ci-après est celui qui est en usage sur CARPE.



Pour la négociation du V.42, le modem appelant envoie une suite de caractères appelée ODP (*Originator Detection Pattern*) et attend pendant 750 ms une réponse de type ADP (*Answerer Detection Pattern*) de la part du modem appelé. La réponse ADP peut indiquer que le modem appelé supporte la correction d'erreur ou non. Si la correction d'erreur est supportée, le modem appelant négocie le LAP-M (une procédure qui ne sera pas décrite ici). Si la négociation échoue, le modem appelant envoie une séquence de «polling» MNP (protocole alternatif). S'il n'a pas de réponse, le mode normal (sans correction) est alors utilisé.

Le modem appelé envoie des bits «1» (*marks*) et examine ce qu'il reçoit. Si l'ODP est détecté, il envoie l'ADP et attend la négociation LAP-M. Si c'est le «polling» MNP qui est détecté, alors il répond. Si aucun protocole n'est détecté, le mode normal est utilisé.

Tous les modems ne réagissent pas de la même manière, certains n'ayant que le MNP, et la description précise du fonctionnement manque. Il n'est donc pas aisé de trouver rapidement les paramètres corrects de la configuration pour établir un tel dialogue. Certains modems ont même la possibilité de renégocier dynamiquement une vitesse inférieure ou supérieure selon le taux d'erreur, mais ce n'est pas le cas sur CARPE.

LA COMPRESSION DES DONNÉES

Le protocole de compression est soit le V.42bis, soit le MNP-5, dépendant du protocole de correction d'erreur utilisé (nécessaire !). La principale action est avant tout la suppression des «*start/stop bits*», ce qui permet déjà un gain de 20% sur le volume à transférer. Les algorithmes étant un peu différents, le MNP-5 arrive à compresser dans un rapport de 2:1 alors que le V.42bis arrive à un maximum théorique de 4:1.

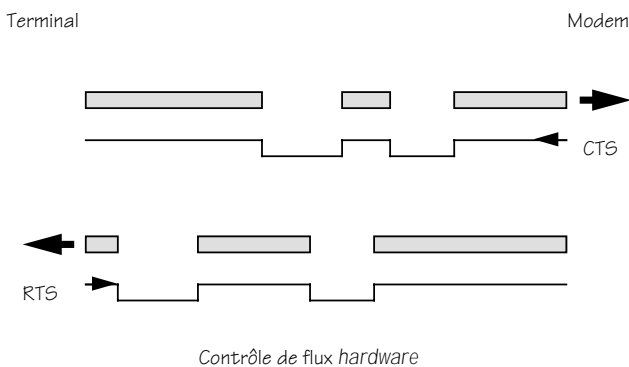
Cela explique parfois la confusion (volontaire?) de certains marchands qui prétendent que leur modem peut transférer des données à 38400 bps. Si certains modems peuvent en effet travailler à cette vitesse sur l'interface digital V.24, il vaudrait mieux exprimer la vitesse de transmission en caractères par seconde. Selon le type de fichier et le protocole, une vitesse *apparente* de 20000 bps est un maximum.

Pour les lignes *haute vitesse* de CARPE, des raisons techniques limitent actuellement le transfert de l'interface V.24 à 19200 bps.

LE CONTRÔLE DE FLUX

Avec les modems rapides, de nouvelles difficultés sont rencontrées au niveau de la régulation du flux des données. Un modem *haute vitesse* supporte généralement le contrôle de flux *hardware* (CTS/RTS) et *software* (XON/XOFF) et comporte un *buffer* interne pour les données.

Le contrôle de flux local est utilisé entre l'équipement terminal et le modem; il ne doit pas forcément être du même type que celui du modem à distance. Il doit cependant être valable dans les deux sens (terminal ↔ modem).



Lorsque le modem n'arrive plus à transmettre assez vite les données, son buffer étant presque plein, il le signale en supprimant le signal CTS (ou en envoyant le caractère XOFF). Le rétablissement du signal (ou l'envoi du caractère XON) rétablit le flux terminal → modem.

Lorsque le terminal n'arrive plus à absorber les données provenant du modem (cela n'était guère probable auparavant), il le signale en supprimant le signal RTS (ou par le caractère XOFF). Lorsque le terminal désire rétablir le flux, il remet le signal RTS (ou le caractère XON).

Certains programmes de communication (avec SLIP, par exemple) utilisent les caractères XON et XOFF

comme données et il est absolument nécessaire d'utiliser le contrôle de flux *hardware*, ce qui est le cas sur CARPE.

CONSEILS

Etant donné la diversité des constructeurs et des types de modem, aucune «recette de cuisine» ne peut être donnée pour l'utilisation correcte avec un équipement existant (CARPE, par exemple). La conformité aux normes du CCIT donne de meilleures informations que certains arguments publicitaires et un test avant l'achat permet de s'assurer que le modem peut être configuré correctement pour l'usage qu'on désire.

1. lire attentivement la notice du modem et prendre note de la configuration d'usine;
2. utiliser un câble adéquat entre l'équipement terminal et le modem (surtout lorsque le contrôle de flux *hardware* est utilisé);
3. lors de l'utilisation d'un logiciel de communication, lire également attentivement le mode d'emploi (permet parfois de savoir si le type de modem est supporté);
4. utiliser la vitesse correcte de communication entre l'équipement terminal et le modem (bien des modems détectent automatiquement la vitesse);
5. configurer l'équipement terminal pour travailler avec 8 bits, sans parité et 1 stop-bit (certaines rares applications demandent parfois 7 bits et parité paire!);
6. se méfier des impulsions de taxation pour le modem appelant, tous n'ont pas un filtre correspondant aux normes européennes (12 kHz au lieu de 16 kHz aux USA !).

Le SIC n'offre aucun support direct aux utilisateurs, seule une notice est disponible au secrétariat.

CONCLUSION

Un modem acheté bon marché à l'étranger, sans aucun support, peut être inutilisable avec l'application requise. De plus, selon la législation en vigueur, seuls les modems agréés par les PTT peuvent être utilisés sur le réseau téléphonique public (cela ne signifie pas forcément qu'ils répondent fidèlement aux normes «V» annoncées par le constructeur, mais qu'ils respectent seulement certains points édictés par les PTT).

Notes:

Un article en langue anglaise, beaucoup plus détaillé, est à disposition des lecteurs qui désirent en savoir plus sur le serveur anonyme ftp de SWITCH, nic.switch.ch dans le fichier docs/About_Modems

Les personnes qui utilisent le call-back de CARPE avec un modem «haute vitesse» peuvent envoyer par e-mail à virchaux@sic.epfl.ch la configuration (séquences AT) qu'ils utilisent en précisant le type d'équipement (PC, Mac, etc...) et le logiciel utilisé, le modèle et le fabricant du modem ainsi que toute autre information utile. Il serait ainsi possible de mettre ces données sur un serveur public par la suite. ■

C++, version 3.0

PAR FRANCIS LAPIQUE, SIC-INFORMATIQUE INDIVIDUELLE

Le langage C++ a subi et va subir encore des modifications sur la base du document *The Annotated C++ Reference Manual* ou ARM de Ellis et Stroustrup et ce, probablement jusqu'à ce que le comité ANSI C++ ait terminé son travail.

Dans sa dernière version, la version 3.0, on compte des modifications de deux sortes:

- des extensions de langages, c'est à dire de nouvelles fonctionnalités ajoutées au langage comme les formes template et la gestion des exceptions
- des changements apportés au langage, c'est à dire des modifications d'ordre sémantique comme par exemple l'introduction des types imbriqués à l'intérieur des classes.

Nous allons détailler dans cet article ces principaux changements en pensant à tout ceux qui ont l'intention de remettre à jour leurs vieux code 2.x avec une mise en garde: méfiez-vous de la qualité d'implémentation de ces nouveaux compilateurs. L'attitude raisonnable est avant tout de s'assurer que la compilation et l'exécution s'opèrent dans l'état, j'entends par là comme décrit dans l'ARM. C'est une des raisons pour lesquelles je renoncerais à décrire la gestion des exceptions car je n'ai trouvé aucune implémentation disponible.

TYPES IMBRIQUÉS

Les types imbriqués, c'est-à-dire la possibilité de déclarer une classe à l'intérieur d'une autre classe ont été profondément remaniés dans cette version 3.0. Dans la version 2.0, la portée des noms de types (classes, unions, enums, et typedef) ne dépendait pas du fait d'être définis à l'intérieur ou non d'une classe. Avec 3.0 on a refait les choses: le type imbriqué est non seulement englobé dans la portée de la classe englobante, mais le niveau d'accès est aussi conforme à la section de la classe dans laquelle il est placé. L'idée est simple: on veut désormais cacher un type qui ne présente un intérêt général que pour une classe cliente et ses classes dérivées. On résume ça en disant que les types imbriqués sont des mécanismes pour éviter des pollutions de l'espace global de noms.

La définition d'une classe imbriquée est la même que celle d'une classe non-imbriquée. Sa visibilité est donc limitée à la portée de sa classe englobante. Ce qui signifie par exemple qu'une classe imbriquée ne peut accéder directement aux membres non-statiques de sa classe englobante. Je dis directement car elle pourra le faire via un pointeur ou une référence.

Ces propos sont illustrés par l'exemple suivant (tiré de l'ARM), où la classe *inner* est la classe imbriquée, et *enclose* la classe englobante.

```
int x;
int y;

class enclose {
public:
    int x;
    static int s;
    class inner {
        void f(int i)
        {
            x = i;//erreur affectation de enclose::x
            s = i;//ok : affectation de enclose::s
            ::x = i;//ok:affectation de x globale
            y = i;//ok: affectation de y globale
        }

        void g(enclose* p,int i)
        {
            p->x = i;//ok:affectation de enclose::x
        }
    };
};

inner* p =0;//erreur 'inner' pas dans la portee
```

La variable statique *s* étant publique, l'affectation *s = i* ne pose pas de problème. L'affectation directe *x = i* n'est pas permise car *inner* n'a pas d'accès direct au membre *x* de la classe englobante *enclose*. Dans cet exemple, on a choisi d'affecter le membre *x* de la classe englobante à travers une fonction membre *g* qui compte comme paramètre un pointeur sur la classe englobante. Pour accéder l'instance globale *x*, il a fallu utiliser l'opérateur de portée globale *::x*. De manière générale la règle du jeu est la suivante: si le nom n'est pas trouvé à l'intérieur de sa portée, la recherche s'effectue dans la portée de la classe englobante; une fois la résolution de nom faite, on passe à l'étape suivante celle de la vérification des droits d'accès et de compatibilité de types.

La fonction membre *g* est soumise aux règles d'accès usuelles (public, protected et private):

```
class E {
    int x;
    class I {
        int y;
        void f ( E* p,int i )
        {
            p->x = i;//erreur E::x est private
        }
    };
    int g (I* p )
    {
        return p->y;//erreur: I::y est private
    }
};
```

Pour faire appel au type *inner* en dehors de la portée de la classe englobante, il faut utiliser une syntaxe particulière que l'on va décrire tout de suite. Toute fonction

membre, comme tout membre statique d'une classe imbriquée, peut être définie au niveau de la portée globale via un opérateur de portée de classe étendu :

```
class enclose {
    class inner {
        static int x;
        void f(int i);
    };
};

typedef enclose::inner ei;
int ei::x = 1;
void enclose::inner::f(int i) { /* ... */ }
```

Pour compléter, nous avons vu que cette restriction d'accès direct aux membres de la classe englobante ne s'appliquait pas aux membres statiques, il faut y ajouter également les noms de type à savoir les typedef et les énumérations.

LES CLASSES TEMPLATE

Avec la notion de types paramétrés (polymorphisme paramétrique), C++ 3.0 introduit un puissant mécanisme de réutilisation. Ces types paramétrés vont vous aider à définir des familles de types et fonctions. Par exemple, supposons que vous vouliez réutiliser les algorithmes d'une classe *Stack* pour des piles de tous types (entier, complexe, caractère, *ecole_ploytechnique*, *window*, *volcans*...). Si le type n'affecte pas le comportement de la classe, C++ vous offre la possibilité d'écrire une classe patron d'un type général T, où T est supposé avoir toutes les propriétés que *Stack* requiert. Le compilateur va prendre en charge la construction d'un *Stack* d'objets d'un type spécifique en substituant dans le corps du patron les caractéristiques de ce type à chaque apparition du paramètre de type T. Une instance de *Stack* de type int est réalisée par la syntaxe :

Stack<int> anIntegerStack;

Avant d'aller plus loin, un petit retour en arrière: qu'est-ce que l'on faisait pour s'en sortir avec la version 2.0. On avait plus ou moins les solutions suivantes:

- les facilités des macros du préprocesseur
- la commande sed
- l'héritage

```
#define name2(a,b) a ## b
#define declare(a,t) name2(a,declare)(t)
#define Stack(type) name2(type,Stack)
#define Stackdeclare(type)\
class Stack(type) {\
    private:\
        type *bottom;\
        type *top;\
        int totalsize;\
    public:\
        Stack(type)(int size) :totalsize(size) {\
            bottom = new type[size];\
            top = bottom + size;\
        }\
        type pop(void) {\
            return *top++;\
        }\
        void push(type item) {\
            *--top = item;\
        }\
        ~Stack(type)(){ delete bottom; }\
};
```

Avec les macros, en prenant l'exemple d'une classe *Stack*, on définit l'interface de la classe dans un fichier *stack.h* comme sur le code ci-dessus.

Un appel à la macro *name2(a,b)* génère le nom concaténé *ab*. Ensuite un appel à une instruction *declare(Stack,char)* réalise l'expansion *name2(Stack,declare)(char)* dont on réalise à son tour l'expansion en *Stackdeclare(char)* qui est étendu en :

```
class charStack { private: char *bottom; char *top; int
totalsize; public: charStack (int size) :totalsize(size) {
bottom = new char[size]; top = bottom + size; } char
pop(void) { return *top++; } void push(char item) { *--top
= item; } ~charStack () { delete bottom; }};
main()
{
    charStack ia(24);
}
```

Je vous laisse découvrir les inconvénients de cette méthode. Avec la commande sed vous allez passer quelque chose du genre :

sed -e «s/<T>/<ST1/g> -e «s/<T&&/<ST1ST1ACC/g>»
sur un code :

class <T>»SLL»ist etc.....

Je ne ferai aucun commentaire car je ne voudrais pas me mettre à dos nos gourous des commandes sed, awk et grep.

La question du choix entre une approche héritage/fonctions virtuelles et types paramétrés est plus sérieuse car elle souligne deux approches complètement différentes de la réutilisation: le paramétrage en C++ est en effet basé sur la **réutilisation du code source** et non sur celle du code objet. Si vous constatez que votre type T affecte l'interface de la classe il faut choisir l'approche héritage. En effet la seule chose que vous allez pouvoir faire dans ce cas, c'est une spécification d'interface et propager cette spécification par des fonctions virtuelles. Prenez l'exemple suivant où on va demander à ceux qui ont la charge d'implémenter les classe dérivées de la classe de base *Avion* de spécifier les méthodes d'*atterrissage* et *décollage*.

```
class Avion {
public:
    virtual ~Avion();
    virtual void atterrissage() =0;
    virtual void decolage() =0;
};
class Fa_18:public Avion {
public:
    void atterrissage();
    void decolage();
    .....
};
class Boing_747:public Avion {
public:
    void atterrissage();
    void decolage();
    .....
};
```

Revenons à nos classes template. Les classes template se comportent exactement comme les classes non-template. La déclaration d'une classe *Stack* paramétrée est la suivante :

```
template<class T>
class Stack {
// .....
};
```

Le mot réservé `template` est placé au début de chaque définition de classe template. Ce mot clé est suivi d'une liste de paramètres formels, délimitée par des signes d'inégalité. Le mot `class` dans ce contexte indique que l'identificateur qui suit représente un paramètre type.

Le code suivant illustre le célèbre exemple de la pile que nous allons détailler; il commence par une déclaration préalable de la classe `Stack`:

`template<class T> class Stack;`

```
#ifndef STACK_H
#define STACK_H
template <class T> class Stack;
template <class T>
class StackNode {
    T data;
    StackNode *next;
    StackNode(const T& newdata, StackNode<T> *nextNode)
        :data(newdata),next(nextNode) {}
    friend class Stack<T>;
    friend ostream& operator<<(ostream&, StackNode<T>&);
};
template <class T>
class Stack {
private:
    StackNode<T> *top;
public:
    Stack() :top(0){}
    ~Stack();
    void push(const T& object);
    T pop();
    friend ostream& operator<<(ostream&, Stack<T>&);
};
#endif
```

Cette définition préalable est suivie par celle de deux classes template `StackNode` et `Stack`. On commence par la classe `StackNode`. On utilise le paramètre `T` comme un spécificateur de type de la même façon qu'un nom de classe non-template. On a déclaré l'argument du constructeur comme étant un type référence pour des raisons de performance. La même préoccupation d'efficacité détermine l'implémentation du constructeur où on a utilisé une initialisation `:data(newdata)` et non une affectation `data = newdata`. En effet, dans le cas d'une affectation `data` est initialisé deux fois: le constructeur par défaut invoqué en priorité, puis le constructeur de copy. On a déclaré la classe `Stack` comme amie de la classe `StackNode` afin qu'elle puisse manipuler des objets de la classe `StackNode`. En ce qui concerne cette déclaration d'amitié on avait le choix entre deux écritures:

**`template<class T> friend class Stack;`
`friend class Stack<T>;`**

C'est cette deuxième écriture qui reflète notre pensée. Ce que l'on veut c'est que la classe `Stack` de types entiers soit amie de la classe `StackNode` de types entiers; on ne veut pas qu'elle soit amie d'une instance de complex ou window de la classe `StackNode`. Enfin on trouve une déclaration de l'opérateur de sortie `<<` de la classe `StackNode`. C'est une fonction template avec le même type de déclaration d'amitié. Quant à la classe `Stack` elle compte deux méthodes `push()` et `pop()` qui vont respectivement ajouter et sortir le premier élément de la pile.

La définition de l'interface de la classe étant définie, on passe à l'étape suivante celle de l'implémentation. Le destructeur vide la class `Stack` de ses éléments:

```
template <class TYPE>
Stack<TYPE>::~Stack()
{
    while(top) {
        StackNode<TYPE> *tmp = top;
        top = top->next;
        delete tmp;
    }
}
```

Les fonctions `push()` et `pop()` avec leur type de retour `void` et `TYPE`:

```
template <class TYPE>
TYPE Stack<TYPE>::pop()
{
    assert (top != 0 );
    StackNode<TYPE> *topOfStack = top;
    top = top->next;
    TYPE data = topOfStack->data;
    delete topOfStack;
    return data;
}
template <class TYPE>
void
Stack<TYPE>::push(const TYPE& object)
{
    top = new StackNode<TYPE>(object,top);
    assert ( top != 0 );
}
```

On veut fournir aux clients de notre classe paramétrée `Stack` la possibilité de visualiser le contenu de la pile. On va donc surcharger l'opérateur de sortie `<<`. On pourrait le faire pour les entiers, `String` :

`ostream& operator<<(ostream&, Stack<int>&);`
`ostream& operator<<(ostream&, Stack<String>&);`

mais on ne va pas faire comme ça. Ce que l'on va faire c'est surcharger notre opérateur de sortie en une fonction template:

```
template <class TYPE>
ostream&
operator<<(ostream &os, Stack<TYPE>& s)
{
    os << "< ";
    StackNode<TYPE> *p;
    for ( p=s.top;p;p=p->next )
        os << *p << " ";
    os << ">" << endl;
    delete p;
    return os;
}
```

On a bien pris garde de déclarer cet opérateur comme ami de la classe `Stack`. Il reste un dernier point à régler, on a reporté le problème de la sortie sur la classe `StackNode`:

`os << *p`

Comme pour la classe `Stack`, on va donc déclarer un opérateur de sortie de la classe `StackNode` comme une fonction template:

```
template <class TYPE>
ostream&
operator<<(ostream &os, StackNode<TYPE>& s)
{
    os << s.data;
    return os;
}
```

La fin est toute proche, reste à régler l'instanciation de ces classes template. Une classe template est instanciée en concaténant le nom de la classe à la liste des paramètres comprise entre des signes d'inégalité:

```
#include "TStack.C"
main()
{
    Stack<double> s;
    s.push(10.);
    s.push(50.);
    cout << s;
}
```

Tout va bien, c'est fini. En fait non, le cas des types `char*` n'est pas bien traité. Le constructeur par défaut est inapproprié

```
StackNode(const T& newdata, StackNode<T> *nextNode)
: data(newdata), next(nextNode) {}
```

On a pas d'autre choix que celui d'écrire une instance de classe template spécialisée pour le type `char*`:

```
StackNode<char*>::StackNode(char* str,
                             StackNode<char*> *nextNode)
: next(nextNode) {
    data = new char[strlen(str) + 1];
    assert ( data != 0);
    strcpy(data, str);
}
```

Le dernier petit (r)affinement consiste à imbriquer notre classe `StackNode`. Mais attention, les définitions template ne peuvent être imbriquées à l'intérieur d'une classe. Ce n'est pas un problème car on ne cherche pas des dualités multiples, mais univoque. C'est exactement le même problème qui a été décrit plus haut pour les fonctions amies.

CONCLUSION

Cette conclusion n'a pas un rapport direct avec le contenu de cet article, c'est plutôt une réflexion sur le support autour de C++ qui est en train d'évoluer. Nous avons à répondre à deux types d'interrogation:

- Qu'est-ce que vous pouvez me proposer comme formation?
- Qu'est-ce que vous pouvez me proposer comme bibliothèque de classe?

Le problème posé par la première question est celui de l'hétérogénéité des formations et motivations des demandeurs. Nous y avons répondu en mettant en place des cours de niveau moyen sur trois jours. Le problème posé par la deuxième question est lié à la disponibilité de chacun d'entre nous. On peut essayer d'entreprendre les actions suivantes :

- essayer de récupérer ce qui s'est fait à l'Ecole, le classer et le diffuser;
- avec l'aide du département de Mathématiques, mise en place d'une bibliothèque numérique (matrix.h, linpack.h) sur la base des BLAS;
- pour le monde Silicon Graphics et celui des applications 3D, une formation autour de la bibliothèque Inventor.

Je fais un appel aux bonnes volontés, ce n'est pas un SOS mais ça pourrait le devenir ! ■

Le Centre National de Supercomputing de Manno (CSCS) publie le journal CROSSCUTS depuis le mois de juin 1992. Ce journal s'adresse aux utilisateurs du CSCS ainsi qu'à toute personne intéressée par les applications scientifiques et techniques des ordinateurs à haute performance.

On peut s'y abonner sur simple demande en envoyant un message à: crosscuts@cscs.ch, ou en écrivant à l'adresse suivante:

*Centro Svizzero di Calcolo Scientifico
Via Cantonale
CH 6928 Manno*

Nous vous encourageons à contribuer au journal par des articles et des annonces. Toute proposition à ce sujet devrait être envoyée de préférence à l'adresse électronique sus-citée.

Bonne lecture

Marco Tomassini, CSCS

Cours pour les utilisateurs du NEC SX-3 du CSCS

Deux séries de cours sont organisées pour les utilisateurs du NEC:

- à Zurich du 19 au 21 octobre (ETHZ)
 - à Lausanne du 18 au 20 novembre (SIC-EPFL)
- Horaire: 9h00-12h.30 partie théorique
dès 14.00 travaux pratiques*

Les prérequis pour ces cours sont une connaissance d'Unix et de Fortran. Une expérience sur un ordinateur vectoriel n'est pas nécessaire. Les sujets suivants seront abordés:

- Organisation du CSCS
- Introduction à l'architecture du NEC SX-3
- Soumission à distance de jobs au SX-3
- Principales caractéristiques du Fortran SX-3
- Comment convertir des programmes pour les faire exécuter sur le SX-3
- Analyse de performances d'un programme
- Introduction à la vectorisation et à l'optimisation d'un programme

Les inscriptions sont à envoyer par courrier électronique à courses@cscs.ch, en indiquant le lieu du cours, le nom du participant, sa nationalité et l'institut.

*Date limite pour les inscriptions: **9.10.92 pour Zurich**
6.11.92 pour Lausanne*

Ces cours sont gratuits; ils pourront être annulés si il y a moins de 10 inscrits. Les frais de transport et de séjour sont à la charge des participants.

Marco Tomassini, CSCS

couverture sont celles du SIC: de 9h00 à 12h00 et de 14h00 à 17h00.

Quels services la PAT peut-elle vous rendre?

- assistance en salles publiques (stations, MAC, PC, scanner et imprimantes)
- consultation et/ou orientation documentation
- accès aux serveurs d'information
- assistance sur les serveurs centraux. *Remarque: vous pouvez envoyer directement à sic-cray@sic.*
- demandes d'informations générales sur le SIC
- boîte aux lettres pour d'autres services du SIC, si vous ne pouvez les joindre directement, comme le Bip réseau ou le Centre de Logistique

Nous vous rappelons également que le détail des services et des produits que le SIC vous propose se trouve dans le Catalogue des Prestations du SIC. La version actuelle est celle de mai 92. Si vous souhaitez l'obtenir, vous pouvez vous adresser à la réception du SIC (tel: 2211) ou au service de la Pat.

Marie-Christine Sawley, SIC-Assistance

CRAY 2 ET PASCAL: PLANIFICATION DES INTERRUPTIONS

Suite à la mise en production du serveur de calcul Pascal (Cray YMP4E), l'horaire de l'entretien préventif du Cray 2 est modifié de manière à assurer, sauf cas exceptionnel (problème urgent à résoudre, mise à jour du matériel ou du logiciel,...) la disponibilité permanente d'au moins l'un des deux serveurs de calcul.

Concrètement, l'entretien préventif a lieu alternativement chaque lundi matin de 07h00 à 10h30 environ, pour Pascal lors des semaines paires, et pour le Cray 2 lors des lundis des semaines impaires.

En cas de nécessité, la période de maintenance pourrait commencer déjà à 06h00.

Michel Jaunin, SIC-Exploitation

LES AVATARS DE NESTOR

Dès le début de l'année, lors de sa mise en route, puis plus tard, de sa mise en production, le serveur de fichiers Nestor a posé divers problèmes :

- *instabilité* : le système s'arrête de manière inattendue, avec dans certains cas destruction d'une partie des fichiers «système»;
- *performance* : les performances NFS sont inférieures à ce que l'on pouvait attendre de ce type de machine.

Après avoir tenté diverses modifications apportées au matériel et logiciel pour résoudre les problèmes d'instabilité (les plus critiques lorsqu'il s'agit d'un serveur général dont la disponibilité élevée faisait partie des contraintes du cahier des charges et des critères de choix), Cray Research a décidé de remplacer la configuration complète par du nouveau matériel sur lequel on trouvera la dernière version du système d'exploitation. L'installation actuelle, Y-MP EL/2-256, est la toute première livrée dans le monde. Plusieurs corrections et améliorations y ont été

apportées, simultanément à celles apportées aux systèmes sur chaîne de fabrication. Plusieurs dizaines de systèmes fonctionnent actuellement à satisfaction; le remplacement de notre machine, première de la série, par un nouveau système devrait permettre d'écarter toute une série de problèmes de jeunesse inhérents (?) aux premiers modèles de la série et par là d'atteindre la disponibilité attendue.

En ce qui concerne les mauvaises performances NFS, Cray a reconnu le problème. Les spécialistes des produits concernés sont en cours d'analyse de la situation, et ont pour mission d'améliorer sensiblement cette performance. Vu la complexité du problème, et tenant compte des contraintes de qualité à maintenir, si ce n'est à améliorer, les solutions prendront un certain temps à être implémentées; il est d'ailleurs probable que l'amélioration de performance soit réalisée par étapes.

Dans la pratique, le passage de Nestor à son successeur est prévu, au moment où nous écrivons ces lignes, pour fin septembre-début octobre (la date exacte sera communiquée par les moyens d'information usuels, *message of the day*, conférences électroniques *News*, etc.) après vérification de la stabilité du nouveau matériel et logiciel ce passage sera «presque» transparent pour l'utilisateur puisque la nouvelle machine reprendra le nom et les adresses réseau du Nestor actuel : il suffira de démonter puis de remonter à travers NFS les file-systems du serveur de fichiers auxquels on désire accéder. Egalement, les fichiers actuellement migrés (au sens DMF du terme) ne le seront peut-être plus sur la nouvelle machine, et réciproquement, des fichiers résidant maintenant sur disque sur Nestor pourront se retrouver archivés sur la nouvelle machine.

**Michel Jaunin & Martin Ouwehand,
SIC-Exploitation**

ARRÊT DE SICSUN

Comme annoncé dans le FI 5/92, le Serveur Unix du Sic *sicsun* arrête ses activités publiques le 31 octobre 1992.

Le nouveau serveur de calcul *pascal* (FI 8/92) prend la relève. Les utilisateurs de *sicsun* doivent donc migrer leurs applications et fichiers vers ce nouveau serveur.

Sur *pascal* sont déjà installés: matlab, maple, nag et uniras, ainsi que les compilateurs C, Fortran et Pascal.

Les utilisateurs de *sicsun* sont donc cordialement invités à passer au plus vite au nouveau serveur de calcul du SIC. Pour faciliter ce passage, les file-systems de *pascal* sont montés par NFS sur *sicsun* et la migration de vos applications devient facile en étant validé sur *pascal*. La validation sur le serveur de fichiers *nestor* est aussi conseillée (et nécessaire pour les utilisateurs manipulant de gros fichiers).

Toute demande d'ouverture de compte sur *pascal* et *nestor* doit être adressée à :

Adamo Vionnet (e-mail: vionnet@sic.epfl.ch, ☎ 2214).

Toute demande d'assistance à la migration doit être adressée à :

Steve Williams (e-mail: williams@sic.epfl.ch, ☎ 2253).

Bienvenue sur *pascal* et bonne migration.

Felipe Valenzuela, SIC-Exploitation

formation

Les cours ci-dessous sont ouverts à tous, membres ou non de l'EPFL. En priorité et gratuitement pour le personnel de l'EPFL. Nous pouvons également organiser des cours à la demande, un logiciel vous intéresse plus spécialement, faites-le nous savoir !

Inscriptions et renseignements:

Josiane Scalfo, SIC-EPFL, CP 121, 1015 Lausanne

☎ 693.2244 (le matin uniquement)

Fax : 021 693 22 20

QuickMail ou E-mail : scalfo@sic.epfl.ch

CRAY

Introduction aux serveurs centraux (1 demi-jour)

Ce cours est destiné aux nouveaux utilisateurs de Cray-2 ou de Pascal. Il sera centré sur les particularités d'Unicos et suppose comme pré-requis une connaissance des commandes de base d'Unix :

- Description des services centraux (Cray-2, Nestor et Pascal)
- Environnement de programmation
- Utilisation des cartouches
- Gestion de l'espace-disque
- Travail en batch

Renseignements :

Jacqueline Dousson, SIC-SA, E-mail : dousson@sic.epfl.ch

N° C55 01.10.92 08h30 - 12h30

MACINTOSH

Pour donner suite à plusieurs demandes, le SIC organise un cours **MACPROJECT** (voir liste ci-après) dont l'objectif est de donner les bases d'une gestion de projet, en général et grâce à MacProject, avec sa préparation, son suivi et ses problèmes. Le plan de cours est le suivant :

- le jargon de la conduite de projet
- la création d'un projet
- les 9 manières de voir un projet
- comment commencer
- le lissage des tâches
- le calage au plus tôt ou au plus tard
- le suivi
- la recherche
- la réduction de la durée totale
- la liste de choses à faire
- la réduction des coûts
- quelques exercices pratiques

Cycle de base complet (6 demi-jours)

Introduction au Macintosh, MacDraw Pro, Word 5.0,

Excel 4.0 et FileMaker Pro

N° Mac293 05,07, 12, 14, 19 et 21.10.92 08h15 - 12h00

N° Mac316 28,30.10 & 5, 6, 11 et 13.11.92 08h15 - 12h00

ClariscAD (2 demi-jours)

Connaissances approfondies d'un logiciel de dessin (MacDraw,...) indispensables!

N° Mac327 30.11 & 03.12.92 08h15 - 12h00

Excel 3.0 à 4.0 transition (1 demi-jour)

N° Mac315 02.10.92 08h15 - 12h00

Excel 4.0 avancé (3 demi-jours)

N° Mac321 26 & 29.10 & 02.11.92 08h15 - 12h00

FileMaker Pro avancé (3 demi-jours)

N° Mac322 01, 06. & 08.10.92 08h15 - 12h00

Illustrator (2 demi-jours)

Connaissances approfondies d'un logiciel de dessin indispensables!

N° Mac305 03 & 05.11.92 13h30 - 17h30

MacProject (1 demi-jour)

N° Mac329 12.11.92 08h15 - 12h00

PageMaker (3 demi-jours)

Connaissances approfondies de Word indispensables !

N° Mac326 28.10 & 04 & 11.11.92 13h30 - 17h30

PhotoShop (3 demi-jours)

Connaissances approfondies du Mac indispensables !

N° Mac304 20, 22 & 27.10.92 13h30 - 17h30

QuickMail (1 demi-jour)

N° Mac328 12.11.92 13h30 - 17h30

Scanner et OmniPage (2 demi-jours)

N° Mac306 13 & 15.10.92 13h30 - 17h30

Système 7.0, facilités du Mac (2 demi-jours)

N° Mac324 16 & 19.11.92 08h15 - 12h00

Système 7.0, partage de fichiers (1 demi-jour)

N° Mac325 23.11.92 08h15 - 12h00

Word 4.0 à 5.0 transition (1 demi-jour)

N° Mac318 23.10.92 08h15 - 12h00

Word 5.0 avancé (3 demi-jours)

N° Mac319 03, 05 & 10.11.92 08h15 - 12h00

4e Dimension, programmation (3 demi-jours)

Bonnes connaissances de 4D

N° Mac330 05 & 08.10.92 13h30 - 17h30

et 22.10.92 08h15 - 12h00

PC

Designer 3.01 introduction (2 demi-jours)

N° PC144 26 & 28.10.92 13h30 - 17h30

Designer 3.01 avancé (2 demi-jours)

N° PC151 05 & 08.10.92 08h15 - 12h00

N° PC166 17 & 19.11.92 08h15 - 12h00

DOS introduction (2 demi-jours)

N° PC158 02 & 04.11.92 08h15 - 12h00

Excel 4.0	introduction (2 demi-jours)
N° PC143	19 & 21.10.92 13h30 - 17h30
Excel 4.0	avancé (2 demi-jours)
N° PC153	12 & 15.10.92 08h15 - 12h00
Excel 4.0	base de données (1 demi-jour)
N° PC168	01.12.92 08h15 - 12h00
Excel 4.0	macros (2 demi-jours)
N° PC154	26 & 28.10.92 08h15 - 12h00
Excel 3.0 à 4.0	transition (1 demi-jour)
N° PC155	02.10.92 08h15 - 12h00
N° PC170	03.11.92 08h15 - 12h00
FileMaker	introduction (2 demi-jours)
N° PC164	20 & 22.10.92 08h15 - 12h00
Réseaux	(1 demi-jour)
N° PC157	15.10.92 13h30 - 17h30
Windows 3.1	introduction (3 demi-jours)
N° PC159	09, 11 & 13.11.92 08h15 - 12h00
Word Windows 2.0	introduction (2 demi-jours)
N° PC142	12 & 14.10.92 13h30 - 17h30
Word Windows 2.0	avancé (2 demi-jours)
N° PC152	13 & 16.10.92 08h15 - 12h00

UNIX

ANNONCE DE LA NOUVELLE SALLE DE COURS <STATIONS DE TRAVAIL UNIX>

Il y en avait une pour le monde PC, une autre pour le monde MAC; dès octobre, il y en aura aussi une pour le

monde des Stations de travail UNIX.

Cette salle de cours Stations de travail, avec ses 9 stations couleurs se situe dans les locaux du SIC.

Quelques cours sont déjà programmés:

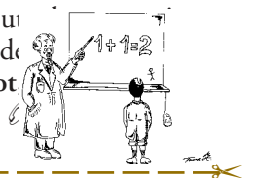
Unix	avancé (5 jours)
N° 3003	19.10 au 23.10.92 09h00 - 17h30
Administration Unix	Solaris 2.0 (5 jours)
N° 3008	23.11 au 27.11.92 09h00 - 17h30
Shell (ksh,sh)	(2 jours)
N° 3005	2.11 & 3.11.92 09h00 - 17h30
Utilisation et concept de X-Window	(2 jours)
N° 3006	16.11 & 17.11.92 09h00 - 17h30
Introduction à Unix	(1 demi-jour)
N° 3010	05.11.92 08h30 - 12h30
Réseau Unix	(1 jour)
<i>Insérer votre station Unix dans le réseau EPNET. Pour toute information complémentaire à propos de ce cours, contactez directement Richard Timsit, E-mail: timsit@sic.epfl.ch.</i>	
N° 3012	10.11.92 09h00 - 17h00

D'autres cours tels que: Sécurité sous UNIX, Communications et réseaux, Programmation C, Programmation système, Programmation réseau, Programmation X11, suivront. Les cours retenus seront ceux pour lesquels le nombre d'inscriptions sera le plus élevé, dans la limite du budget imparti. Vous pouvez d'ores et déjà vous inscrire en remplissant le formulaire d'inscription ci-dessous. Toutes questions relatives aux inscriptions se posent (gentiment) à Josiane Scalfo.

Je reste à votre disposition pour toutes questions et remarques concernant le contenu de

Franck Perrot

E-mail: fperrot@sasun4.epfl.ch,



FORMULE D'INSCRIPTION POUR LES COURS

merci de ne pas grouper sur le même bulletin, les cours C++, Cray, Mac, PC, Unix etc.

A retourner à Josiane Scalfo, SIC-EPFL, 1015 Lausanne

N° du cours	N° cours de remplacement	Nom du cours	Date du cours
-------------	--------------------------	--------------	---------------

.....

.....

Nom : Prénom : E-mail :

Institut : Dépt : Adresse :

s'engage à suivre les cours dans leur intégralité et à respecter l'horaire.

Tél. : Date : Signature :


Autorisation du chef hiérarchique (nom lisible et signature) :

calendrier

Date	Heure	Lieu	Réunion
<i>OCTOBRE 1992</i>			
Jeudi 1er	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles <i>Ch. Zufferey, ☎ 021-693.45.98+3, E-Mail: zufferey@sic.epfl.ch</i> <i>Serveur: sicsoft, polypc, polypc</i>
Jeudi 9	12h00		DÉLAI DE RÉDACTION DU FLASH INFORMATIQUE N° 8
Jeudi 15	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Mercredi 21	14h30	Salle Conférences SIC	MacLine — Groupe des utilisateurs de Macintosh <i>F. Roulet, ☎ 021-693.45.90, E-Mail: roulet@sic.epfl.ch</i>
Mercredi 28	14h15	Salle Conférences SIC	Covax — Réunion des Vax-Managers <i>P. Lachaize, ☎ 021-693.22.54, E-Mail: lachaize@sic.epfl.ch</i>
Jeudi 29	10h00	Salle Conférences SIC	SUNLine — Groupe des utilisateurs de Sun <i>A. Brossard, ☎ 021-693.22.41, E-Mail: brossard@sic.epfl.ch</i>
<i>NOVEMBRE 1992</i>			
Jeudi 5	12h00		DÉLAI DE RÉDACTION DU FLASH INFORMATIQUE N° 9
	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Jeudi 19	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Mercredi 25	14h15	DEC - St.-Sulpice	Codec — Coordination DEC - EPFL <i>P. Lachaize, ☎ 021-693.22.54, E-Mail: lachaize@sic.epfl.ch</i>
<i>DÉCEMBRE 1992</i>			
Mardi 1er	14h30	Salle Conférences SIC	CTI — Commission technique informatique <i>M. Reymond, ☎ 021-693.22.10, E-Mail: reymond@sic.epfl.ch</i>
Jeudi 3	12h00		DÉLAI DE RÉDACTION DU FLASH INFORMATIQUE N° 10
	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Jeudi 17	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles

Flash Informatique

Les articles de ce journal ne reflètent que l'opinion de leurs auteurs.
Toute reproduction, même partielle, n'est autorisée qu'avec l'accord
de la rédaction et des auteurs.

Rédacteur en chef : Marie-Christine Sawley
Comité de rédaction : Jean-Daniel Bonjour, Jean-Michel
Chenais, Nathalie Colombi,
Milan Crvcanin, Jacqueline Dousson,
Pierre-André Haldy, Véronique Jost,
Pierre-Jean Paris, François Roulet &
Jacques Virchaux
Composition : Appoline Raposo de Barbosa
Impression : 
Tirage : 4000 exemplaires
Adresse: SIC-SA EPFL 1015 – Lausanne
☎ 021/693 22 42 & 22 47
Prochaine parution : 27 octobre 1992
Délai de rédaction : 9 octobre – 12h00