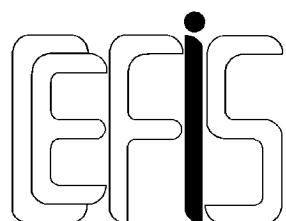


FORMATION

**E
C
H
E
N
E
D
U
C
A
T
I
O
N
R
E
C
H
E**

EN EDUCATION

5.53



**Pour une utilisation efficace des
technologies de l'information**
Notions de base

Etienne Vandeput
Septembre 1997

FACULTES UNIVERSITAIRES
NOTRE-DAME DE LA PAIX



NAMUR

DEPARTEMENT EDUCATION ET TECHNOLOGIE

OBJECTIFS

Cette formation s'adresse à tous les enseignants qui ont envie d'intégrer l'outil informatique dans leur pratique professionnelle (rédaction de notes de cours, gestion d'informations diverses: listes d'élèves, d'exercices, de questions..., utilisation de l'ordinateur comme super-tableau...) et qui ne le connaissent pas ou si peu.

Pour parler clairement, cette formation n'a certainement pas comme objectif de faire acquérir aux participants les connaissances et les compétences nécessaires à cette intégration. **Son but est plutôt de donner une idée précise de ce que représente le travail avec un ordinateur et des logiciels, comme démarche intellectuelle, à travers un mélange d'activités pratiques et de réflexions communes plus théoriques.** Elle met en garde contre des tas d'idées reçues telles: "en cinq minutes vous avez tout compris", "les enfants utilisent très facilement un ordinateur", "le travail avec un ordinateur ne demande pas de réflexion", etc. A son terme, les participants devraient pouvoir décider dans quelle mesure ils souhaitent suivre la voie qui s'ouvre à eux et selon quelle(s) stratégie(s).

La réflexion aura pour objet de faire comprendre le dessous des cartes, dans une raisonnable mesure, pour aider à fonctionner en utilisateur averti, organisé et efficace. Elle se fera par le biais et les manipulations de base de plusieurs "outils": un environnement graphique de travail (Windows 3.11 ou Windows 95), un traitement de texte (Word 2.0, 6 ou 7, WordPerfect 6 ou 7, PerfectWorks,...), un logiciel de dessin par points (PaintBrush, Paint), un logiciel de dessin vectoriel (Presentations, Draw ou le module de dessin de Word 6 ou 7) et quelques petits utilitaires disponibles à partir de Windows.

L'esprit de cette formation est de respecter le rythme des participants, mais aussi de répondre à toutes les questions qui surgiront ou qui sont déjà présentes dans leur tête. Il ne s'agit pas de leur faire ingurgiter des connaissances volatiles (tout change tellement vite à ce niveau) mais de leur faire entrevoir des modes de travail et de leur fournir des images mentales du fonctionnement de l'outil qui leur garantissent l'acquisition d'une autonomie d'apprentissage.

Quelques questions générales à propos desquelles il faudrait avoir des choses à dire à l'issue de la formation:

Que faut-il entendre par "traitement de l'information"?

Comment un ordinateur équipé de différents logiciels traite-t-il l'information?

Quelles nouvelles perspectives cela ouvre-t-il?

Quels nouveaux modes de travail cela induit-il?

(Etant donné la rapidité de l'évolution, tant du matériel que du logiciel...) Y a-t-il des invariants dans le domaine de l'informatique et des technologies de l'information qui garantissent des apprentissages utiles à long terme?

Quelques savoir-faire espérés:

Editer un texte d'une page ou deux (le créer, le sauvegarder, en modifier le contenu, la mise en page...)

Editer un dessin selon les deux modes principaux disponibles et l'intégrer dans le texte

Organiser correctement un support d'information (disque dur, disquettes)

Décrire, à tout moment du travail, sur quels supports se trouve un document (disque dur, disquette, écran, mémoire centrale...)

Décrire les applications présentes en mémoire à tout moment du travail et les rendre actives

PROGRAMME

La répartition approximative des contenus dans les différentes séances est calculée sur cinq journées. Il est évident que la progression est dictée par le groupe, par l'intérêt qu'il marque pour l'un ou l'autre des thèmes abordés. L'objectif étant de rendre les utilisateurs aussi autonomes que possible, les matières abordées ne sont importantes que dans la mesure où elles ne mettent pas en péril l'acquisition de cette autonomie.

Séance n°1

A propos d'exécutants et de "faire faire"...

- Une introduction très imagée du comportement attendu de l'utilisateur efficace et des problèmes qu'il devra résoudre: primitives, stratégies, traduction...
- Exercices de mise en condition

L'exécutant ordinateur

- Le principe de la numérisation (existence de codages), une source de réponses à de nombreuses questions et de solutions à de nombreux problèmes: exercice de codage d'un texte
- Quelques éléments de connaissance des ordinateurs indispensables en vue de conditionner de bons réflexes de la part de l'utilisateur et notamment:
 - un schéma simplifié et (assez) universel de l'ordinateur accompagné d'une évocation des différents périphériques
 - une description des différents supports d'information: papier, mémoire de masse, mémoire centrale...

L'exécutant "dessin par points"

- Comment est-il possible de numériser un dessin: exercice de codage?
- Les conséquences sur les primitives attendues de ce type d'exécutant
- Problèmes à résoudre avec les exécutants de cette catégorie disponibles sur place

Séance n°2

L'exécutant "dessin vectorisé"

- Comment éviter les inconvénients liés au codage du dessin par points?
- Les conséquences sur les primitives attendues?
- Problèmes à résoudre avec les exécutants de cette catégorie disponibles sur place

La valse (ou la métamorphose) des exécutants: évolution au cours d'une séance de travail

- description de l'évolution de l'état de la mémoire centrale au cours d'une séance de travail
- lien avec les primitives du système d'exploitation qui assurent la communication entre la mémoire centrale et la mémoire de masse

Séance n°3**L'exécutant "traitement de texte"**

- Qu'est-ce qu'un texte (à la lumière de ce qui précède)?
- Classification des traitements: le contenu (primitives d'édition), la forme (primitives de mise en page)
- Problèmes et réflexion sur les stratégies possibles d'utilisation

Séance n°4**L'exécutant "système d'exploitation et environnement graphique"**

- Rôle et évolution des systèmes d'exploitation
- Description des primitives de base en fonction des situations locales (celle de l'école, et dans la mesure du possible, celle des participants)
- Problèmes à résoudre avec l'exécutant disponible

Stratégies de gestion des mémoires de masse

- Comment sont organisées les données sur un disque dur, une disquette, un cédérom?
- Comment les organiser quand c'est possible?
- En quoi consiste l'installation d'un programme?

Séance n°5

Outre des développements complémentaires sur les thèmes déjà précisés, un exercice générique est possible. Selon les intérêts du groupe, le thème suivant peut être abordé.

La navigation

- Réflexion à propos des logiciels interactifs sur cédéroms
- Description succincte des outils d'internet et stratégies d'utilisation

1. A propos d'exécutants et de "faire faire"...

Depuis quelques décennies, on assiste à des modifications profondes et des extensions fulgurantes des pratiques en matière de traitement de l'information. Le mot utilisé jusqu'il y a peu pour désigner à la fois, la science, les techniques, les pratiques fondamentalement nouvelles a été le mot "**informatique**". Aujourd'hui, avec la banalisation de certaines de ces pratiques, le terme a plutôt tendance à se rattacher à nouveau à une activité professionnelle relativement précise, à un métier. Pour caractériser l'envahissement de l'électronique dans la vie de tous les jours, on parle plus volontiers de "**technologies de l'information**". Ces appellations et ces changements de vocabulaire sont loin d'être innocents. Le pluriel du mot "*technologies*" traduit une grande variété des domaines d'application et une impossibilité, même pour les spécialistes, d'assurer une expertise dans tous ces domaines. Il met en évidence l'aspect didactique que nécessite une bonne perception et une appréhension correcte du phénomène. Quant au mot "*information*", il apparaît maintenant comme l'élément central du problème. Il ne s'agit donc pas de prendre les choses à la légère. L'acquisition de savoir-faire et de compétences dans le traitement de l'information demande du temps, de la pratique, mais surtout de la clairvoyance.

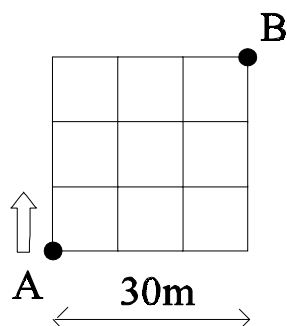
Dans cet ordre d'idée, nous avons l'ambition de faire de vous des utilisateurs efficaces, ou en tous cas, de vous montrer une voie qui vous conduira à le devenir. Nous insisterons donc souvent sur les aspects méthodologiques. Nous passerons fréquemment du temps à discuter ensemble et à réfléchir à des questions dont les réponses risquent fort de vous faire comprendre quels sont les enjeux de ce défi que vous vous lancez aujourd'hui et de vous montrer à quelles conditions vous pourrez le relever.

Mais bien sûr, compétence et expérience sont les produits d'un subtil mélange de théorie et de pratique. Si donc cette dernière fait un peu défaut dans le cadre strict de la formation que vous suivez, il nous paraît important que vous la développiez entre les séances de formation avec l'aide et la complicité des personnes-ressources qui vous encadrent.

En guise de mise en condition, nous allons vous proposer un petit problème qui va nous servir de point de départ à une première réflexion et qui donnera d'emblée un ton souvent adopté au cours de cette formation.

Histoire d'extra-terrestre. Cette histoire est tirée de [1].

Un martien se trouve dans un parc dont les allées (symbolisées par des lignes) sont séparées de 10 mètres. Il se trouve en A, dans la direction indiquée par la flèche. Et vous devez lui donner les ordres pour qu'il se rende en B.



Les seules actions dont il est capable sont de se déplacer de 10 mètres d'un coup, droit devant lui, et de pivoter d'un quart de tour à gauche. Les choses se compliquent un peu si vous savez que pour lui faire accomplir la première action décrite, vous devez prononcer le mot *blanc* et le mot *noir* pour la seconde. Que devez-vous lui dire?

Tâchez de trouver plusieurs solutions au problème. Nous pourrions ensuite les commenter et réfléchir à nos manières de procéder.

Un peu de vocabulaire...

Qu'il s'agisse d'un martien, d'un robot, ou d'un quelconque personnage ou dispositif auquel on peut donner des ordres ou des commandes, nous parlerons d'un **exécutant**.

Lorsque nous désignerons des actions dont est capable un exécutant limité, nous parlerons d'**actions primitives**.

Diriger un exécutant, c'est en quelque sorte définir une stratégie pour lui faire accomplir une tâche donnée, simple ou complexe, à partir de ses primitives.

Ainsi, la conduite du martien de A en B nécessite une stratégie, une route à suivre qui va de temps en temps se heurter à des difficultés liées à l'absence de certaines primitives (pour *tourner à droite*, par exemple). Dans ces cas-là, il s'agit d'examiner si, en décomposant la difficulté, les primitives existantes permettent de la lever (*tourner à gauche* trois fois de suite équivaut à *tourner à droite*).

Si pour solutionner des problèmes, une connaissance minimale de l'exécutant est nécessaire, la résolution de problèmes (en termes d'exécution de tâche) est aussi un moyen de découvrir un peu plus de quoi il est capable. S'installe donc une dynamique d'auto-apprentissage qui peut s'avérer extrêmement efficace au travers de la succession des différents problèmes rencontrés dans le travail de tous les jours.

2. L'exécutant ordinateur

La réflexion qui précède n'a d'autre objet que de vous faire prendre conscience de la manière dont vous serez obligés de vous comporter lorsque vous voudrez effectuer un travail quelconque au moyen d'un ordinateur. Il vaudrait d'ailleurs beaucoup mieux parler de "*faire exécuter*", même si vous avez tout de même le sentiment tout à fait justifié de travailler vous aussi.

En réalité, il convient de distinguer votre travail qui consiste à guider un exécutant, du travail de cet exécutant. En réalité, c'est lui qui traite les informations que vous lui fournirez, de la manière dont vous lui indiquerez.

Il importe donc que vous fassiez plus ample connaissance avec cet exécutant, et surtout, que vous preniez conscience des éléments fondamentaux de sa conception qui orienteront votre façon de travailler avec lui.

A. L'ordinateur n'est rien sans un programme pour le gouverner

Les véritables exécutants auxquels vous aurez à faire répondent tous à l'équation suivante:

$$EXECUTANT = ORDINATEUR + PROGRAMME$$

Qu'est-ce qu'un programme? C'est, d'une certaine manière, l'ensemble des capacités de l'exécutant. Ainsi, si le martien existe "en dur", il existe aussi parce qu'il comprend certains ordres. C'est le programme du martien qui le fait avancer de 10 mètres lorsque le mot *blanc* est prononcé.

Le mot *logiciel* est souvent utilisé à la place du mot *programme*. On distingue aussi le *hardware* du *software*, le *matériel* du *logiciel*.

Pour être utilisable, un ordinateur doit donc posséder dans sa "mémoire", un programme qui lui confère différentes capacités. Vous pouvez vous figurer qu'il existe, par exemple, des exécutants "traitement de texte". N'importe quel ordinateur dont la mémoire contient un programme le transformant en machine à travailler les textes, une sorte de machine à écrire évoluée (quoique, cette image n'est peut-être pas la meilleure) sera pour nous un exécutant "traitement de texte". Nous parlerons aussi d'exécutants "dessinateurs", pour désigner tout ordinateur dont la mémoire contient un programme de dessin, d'exécutants "comptables" etc. En réalité, chaque programme transforme l'ordinateur en un exécutant capable d'effectuer un certain nombre d'actions répondant à un type de travail donné.

Il existe donc des millions d'exécutants, compte tenu de la très grande diversité des programmes existants et des ordinateurs les supportant.

Vous comprenez qu'il vous faut, dès maintenant, vous faire une raison de ce grand nombre d'exécutants présents, et surtout à venir.

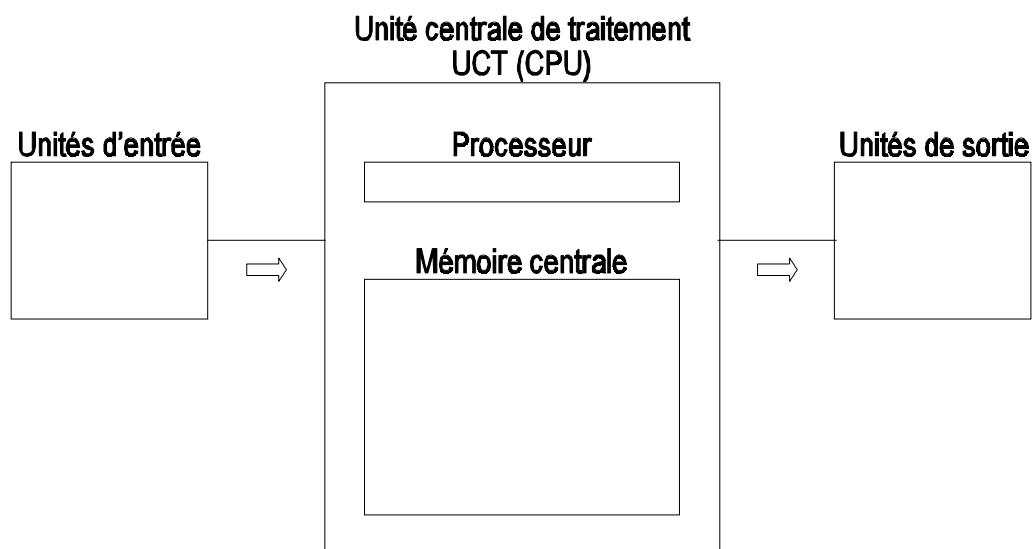
Pour vous permettre de vous familiariser avec eux, nous allons commencer par vous donner une idée de la manière dont est conçu un ordinateur, comment très globalement il fonctionne, et surtout, quelles en sont les limites bien réelles (certains préjugés pourraient nous laisser croire le contraire).

B. Schéma simplifié d'un ordinateur (du point de vue de son fonctionnement)

Nous avons bien écrit *schéma* et non *dessin*. Cette remarque est importante car elle est à la base de confusions gênantes chez de nombreux utilisateurs. Ces confusions sont de nature à rendre précaire, voire inefficace tout travail qu'il font avec un ordinateur. Effectivement, il y a ce que l'on voit et ce que l'on imagine. En matière de fonctionnement d'un ordinateur, c'est ce qu'on imagine qui est primordial.

On a souvent dit de l'ordinateur qu'il sert à traiter l'information. Cette petite phrase appelle de nombreux commentaires. A commencer par: quelle signification faut-il donner à ce mot *information*? quels sont les traitements possibles et pourquoi?...

Nous pourrions dire que depuis qu'ils existent, tous les ordinateurs répondent au schéma simplifié suivant:



Il convient de mettre ce schéma en accord avec ce que vous pouvez observer lorsque vous vous trouvez devant un ordinateur.

Comme vous pouvez le constater, en amont et aval du traitement de l'information par l'ordinateur se trouvent des unités d'entrée et de sortie. Ces unités, vous les trouvez généralement autour de la partie centrale de l'ordinateur. Pour cette raison, on leur donne souvent le nom de *périphériques* (d'entrée et de sortie). En guise de point de départ, pourriez-vous citer des éléments, des parties de l'ordinateur qui vous paraissent être des périphériques en précisant de quelle nature ils sont (entrée, sortie ou les deux)?

Périphériques d'entrée:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Périphériques de sortie:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Périphériques d'entrée/sortie:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Une question fondamentale se pose à leur sujet: à quoi servent-ils?

Pour répondre à cette question, il faut en savoir un peu plus sur la composition de la partie centrale de l'ordinateur, l'unité centrale de traitement (en anglais *CPU: Central Processing Unit*).

Ce n'est un secret pour personne, l'ordinateur fonctionne grâce à des circuits électroniques. Depuis le début, c'est l'électricité qui est à la base de tous les développements de l'informatique.

Les conséquences en sont énormes et conditionnent depuis toujours la nature des produits informatiques et leur influence sur le mode de travail des utilisateurs?

La partie centrale d'un ordinateur se compose d'une mémoire (centrale). Cette mémoire est destinée à contenir les programmes qui transforment l'ordinateur en différentes sortes d'exécutants et les informations que ceux-ci vont devoir traiter. Elle est composée de circuits électroniques. La caractéristique de chacun de ces circuits, c'est qu'à tout moment, il n'a que deux états possibles: ou bien il est parcouru par un courant, ou bien il ne l'est pas. Voilà sous quelle forme se trouve l'information dans l'unité centrale d'un ordinateur.

Cela implique qu'**une transformation de l'information est nécessaire en amont comme en aval**. Dans les deux cas, cette transformation se justifie par le fait que l'information telle que nous avons l'habitude de la traiter n'a pas la forme de milliards de circuits sous tension ou non. **Ces transformations se font de manière transparente pour l'utilisateur par l'intermédiaire des périphériques** dont nous avons déjà parlé. Toutefois, cela n'est pas sans implications. Nous allons d'ailleurs nous y intéresser sérieusement.

Pour l'instant, imaginez-vous que dans la mémoire centrale de l'ordinateur, toute information est irrémédiablement convertie en une suite composée des deux chiffres *0* et *1*, une manière commode de symboliser cette présence ou cette absence de courant dans un circuit.

Le rôle des périphériques d'entrée est de permettre une numérisation de l'information. Monsieur de La Palisse l'aurait bien dit: **pour pouvoir être traitée par un ordinateur, une information doit pouvoir être numérisée**. Celui des unités de sortie est de redonner à l'information un aspect familier à l'être humain. Celui des unités d'entrée/sortie, c'est de lui permettre d'être fixées sur un support qui en assure réutilisation.

Le parcours de l'information est donc linéaire. Elle est numérisée (codée) avant d'être traitée puis restituée grâce à un décodage qui lui redonne l'aspect original. On comprend dès lors qu'il faille un **coordonateur de toutes ces opérations**: saisies de l'information, traitement, restitution sous une forme ou sous une autre. Ce rôle est assuré par **le processeur**, un autre composant électronique capable de réaliser un nombre limité d'opérations élémentaires en des temps records.

Quelques processeurs (marque et type)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

C. Les mémoires

De ce qui précède, vous retiendrez que le passage en mémoire centrale de l'information est limité dans le temps. La mémoire centrale est une mémoire de travail. Une fois traitée, l'information résultante doit être copiée sur un support pour qu'une trace puisse en être conservée. **La mise hors tension de l'ordinateur a pour conséquence de vider la mémoire centrale de tout son contenu.** Quels sont les supports possibles?

Il y a d'abord le support traditionnel que constitue le papier. Si ce support est quelque peu durable, il est difficilement réutilisable car le texte n'est plus sous une forme aisément numérisable. Nous allons examiner de suite les possibilités de numérisation.

Il y a aussi et surtout les supports magnétiques tels les bandes enregistreuses, les disquettes et disques durs et des supports plus modernes comme les cédéroms ou le vidéo-disque (bien qu'actuellement, ces derniers soient essentiellement des unités d'entrée).

Observez qu'on parle d'*unité d'entrée* en ce sens que ce type d'unité permet d'entrer de l'information dans la mémoire centrale. On parle d'*unité de sortie* en ce sens que ce type d'unité permet de sortir de l'information de la mémoire centrale vers un autre support. A noter qu'*entrer* et *sortir* ont ici la signification très particulière de "*faire une copie de ... dans ou sur ...*".

On peut admettre que sur ces différents supports, l'information garde une forme numérisée, ce qui en facilite la réutilisation par un ordinateur. C'est la raison pour laquelle ils portent également le nom de *mémoires* mais on parlera de *mémoires externes* ou de *mémoires de masse* pour les désigner et pour les distinguer de la mémoire centrale. Contrairement à cette dernière, **les mémoires externes gardent leur contenu si l'ordinateur n'est plus sous tension.**

La taille des mémoires externes et centrale se mesure en fonction de la quantité d'information qu'elles peuvent stocker. L'unité de mesure est directement liée à la manière dont celle-ci est codée. Nous pourrions donc en parler plus à l'aise après avoir répondu à la question: comment numériser l'information?

D. La numérisation

L'histoire de l'informatique et le développement des technologies sont construits sur le même objectif qui est de rendre de plus en plus d'informations numérisables. A l'origine, c'est essentiellement pour calculer que les ordinateurs étaient construits. Les problèmes de numérisation étaient donc limités dans la mesure où il suffisait de convertir les nombres du système décimal dans le système binaire (seulement deux symboles: 0 et 1) et inversement. Mais assez rapidement, on s'est rendu compte que les caractères pouvaient aussi être numérisés, ce qui ouvrait la porte à la création de nouveaux programmes et donc de nouveaux outils.

Imaginez une manière de coder les lettres de l'alphabet qui vous permettent de communiquer avec un correspondant en ne lui transmettant que des chiffres (de 0 à 9)

.....
.....
.....

Le nombre de caractères différents utilisés pour écrire des textes n'est pas infini. Bien sûr, les lettres majuscules et minuscules représentent des caractères différents auxquels il faut ajouter les chiffres, les signes de ponctuation et, particularités de certaines langues (latines, scandinaves,...) les caractères accentués. Le tout ne dépasse guère la centaine.

Un seul circuit électronique permettant de coder deux caractères (puisque deux symboles sont disponibles), deux circuits permettent d'en coder quatre, trois circuits huit et ainsi de suite, huit circuits donnent accès à 256 codes différents. On pourrait s'étonner de ce choix de huit (alors que sept auraient pu suffire) mais les problèmes de conversion étant ce qu'ils sont, il est beaucoup plus simple de passer d'un système octal (à huit symboles) à un système binaire et réciproquement. Les circuits ont donc été rassemblés par paquets de huit, ce qui a eu comme conséquence de faciliter le codage des caractères habituels et de quelques autres puisque le nombre de possibilités dépassait les nécessités.

Ainsi, selon un code aujourd'hui standardisé, la lettre A (majuscule) est codée *01000001* dans la mémoire d'un ordinateur, la lettre B (majuscule) *01000010* et ainsi de suite...

Un circuit correspondant à un chiffre binaire porte le nom de **bit** (contraction de l'expression anglaise "**binary digit**": chiffre binaire). **La place que nécessite le codage d'un caractère est appelée octet** (un octet équivaut donc à huit *bits*). La taille des mémoires de masse comme de la mémoire centrale d'un ordinateur se mesure donc en *octets* (en anglais *bytes*). Une évolution rapide des technologies liées au stockage de l'information fait qu'aujourd'hui, on mesure davantage en *kilo-octets* (Ko ou Kb), en *méga-octets* (Mo ou Mb) voire en *giga-octets* (Go ou Gb). Les préfixes *kilo*, *méga* et *giga* correspondent habituellement à des puissances de dix (10^3 , 10^6 et 10^9). Dans ce contexte, ils représentent les puissances de deux (système binaire) les plus proches soit pour l'anecdote: 1.024, 1.048.576 et 1.099.511.627.776.

Notez ici quelques tailles actuellement courantes pour la mémoire centrale d'un ordinateur, une disquette, un disque dur, un cédérom, un vidéodisque:

.....

.....

.....

.....

E. Les conséquences (dramatiques) de la numérisation

Si l'on vous fournit un texte écrit en polonais (à supposer, et c'est fort probable, que vous ne connaissiez pas le polonais), serez-vous capable de compter le nombre de mots du texte? Pourrez-vous souligner tous les adjectifs, les mots de trois lettres, les verbes conjugués? Pourrez-vous y retrouver un mot donné? Si on vous donne un dictionnaire, pourrez-vous traduire ce texte en italien?

Quelles sont vos réponses à ces questions et pourquoi?

.....

.....

.....

.....

.....

Autant de questions et surtout de réponses qui montrent les limites d'un travail lorsque celui-ci est dépourvu de sens. Or c'est bien de cela qu'il s'agit pour nos exécutants, l'information sur laquelle ils travaillent étant numérisée. Quant aux traitements que vous allez leur demander de faire, ils ne pourront jamais concerner que des choses purement formelles.

Pour bien comprendre, rappelez-vous maintenant le problème du codage du texte. Imaginez qu'un intermédiaire existe entre votre correspondant et vous, qui ne reconnaisse que les chiffres. Cet intermédiaire pourrait-il compter le nombre de fois qu'apparaît la lettre A dans le texte? Comment lui expliqueriez-vous la façon de faire? Pourrait-il retrouver un mot, compter le nombre de fois qu'il apparaît dans le texte? Pourrait-il résumer le texte?

La problématique que nous venons d'évoquer apparaît au détour de chaque problème que rencontre l'utilisateur d'un produit informatique. Qu'il s'agisse en traitement de texte d'envoyer le point d'insertion au début du mot suivant, qu'il s'agisse avec un logiciel de dessin de reproduire une figure existante, qu'il s'agisse avec un programme de navigation de rechercher des informations à propos d'un sujet donné sur Internet...

Les progrès de l'informatique, et plus généralement des technologies de l'information, ne sont qu'un combat perpétuel dans lequel se multiplient les tentatives pour exprimer de manière formalisée des traitements qui à priori ne le sont guère. Mais arrivera-t-on un jour à écrire des programmes pour résumer des textes, des programmes pour les traduire de manière cohérente, des programmes pour en corriger complètement l'orthographe?

Si vous répondez oui à cette dernière, examinez donc un peu la phrase suivante: *Je t'en laisserai, tu t'enlacieras!* Ne faudrait-il pas lui préférer la suivante: *Je t'enlacierai, tu t'en laisseras!* Alors comment expliqueriez-vous cela à un exécutant très formaliste?

Si vous le souhaitez, notez ici d'autres réflexions sur le traitement formel

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

F. La valse des exécutants

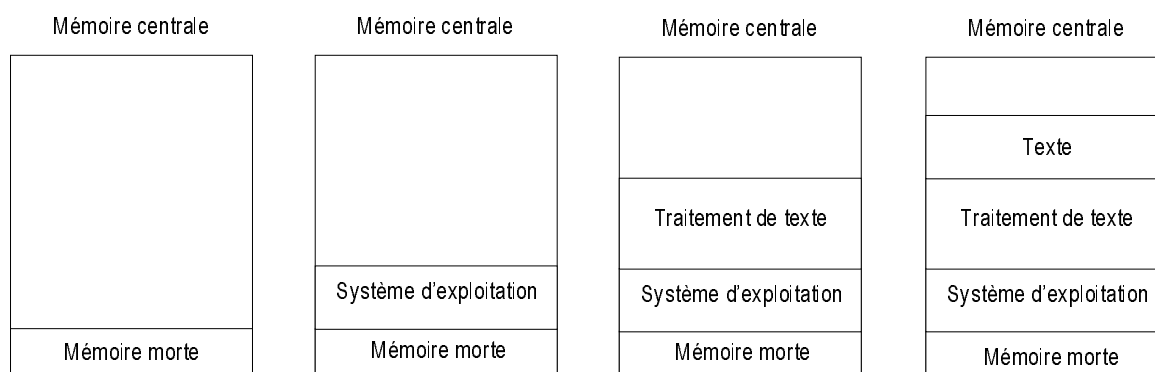
Vous êtes presque parés pour une confrontation avec un premier exécutant. Il nous reste à éclaircir un point important: que se passe-t-il à la mise sous tension d'un ordinateur? d'où viennent les exécutants? comment évoluent-ils au cours d'une session de travail?

Il y a une espèce de petite contradiction dans ce que nous avons affirmé jusqu'à présent. Si nous nous retrouvons, à un moment donné, devant un exécutant, c'est qu'un programme se trouve dans la mémoire centrale. Un exécutant est en effet la combinaison d'un ordinateur et d'un programme. On comprend bien qu'un tel exécutant puisse transformer ses capacités s'il peut faire entrer d'autres informations dans la mémoire. Ainsi, un exécutant peut évoluer au cours d'une session de travail ou laisser la place à un autre selon l'image que vous préférez. Mais nous avons affirmé qu'à la mise hors tension de l'ordinateur, la mémoire centrale de l'ordinateur se vidait complètement. Comment un programme peut-il donc se retrouver en mémoire si l'exécutant au départ a un "encéphalogramme plat"?

La réponse à cette question, c'est que la mémoire centrale d'un ordinateur n'est jamais complètement vide. Une petite partie de celle-ci (qui était d'ailleurs plus importante dans un passé récent) contient un "souffle d'intelligence" sous la forme de quelques instructions systématiquement exécutées à la mise sous tension de l'ordinateur. Cette partie ineffaçable de la mémoire centrale est appelée **mémoire morte** (ou *ROM, Read Only Memory*) par opposition à la **mémoire vive** (ou *RAM, Random Access Memory*). Les instructions qu'elle contient font que l'ordinateur n'est pas seulement un "tas de ferraille et de plastic" mais un exécutant très limité qui exécute, à chaque mise sous tension, le même travail. En gros, et pour faire simple, ce travail consiste à mettre en mémoire, à partir d'une mémoire externe (souvent le disque dur) des capacités minimales sous forme d'un programme ou groupe de programmes appelé **système d'exploitation**.

Une fois le *système d'exploitation* en mémoire, l'exécutant de départ a changé. On est maintenant en présence d'un exécutant capable d'**exploiter** ses ressources et notamment, ses périphériques et sa mémoire centrale.

A partir de ce moment, l'utilisateur que vous êtes peut prendre en charge cet exécutant pour lui confier des tâches telles la gestion des supports d'information (effacer, faire des copies, déplacer...) et surtout le chargement en mémoire d'autres programmes qui vont encore affiner les qualités de l'exécutant comme, par exemple, mettre en mémoire un programme de traitement de texte.



Décrivons un exemple de session de travail.

A la mise sous tension de l'ordinateur, le processeur exécute les quelques instructions contenues dans la mémoire morte de l'ordinateur. Parmi celles-ci, des instructions ordonnent la recherche dans les mémoires externes (disquette et à défaut disque dur) du système d'exploitation pour en faire une copie en mémoire. A la fin des opérations, l'utilisateur que vous êtes n'est, en principe, pas intervenu et il est en présence d'un exécutant "système d'exploitation" ou, si vous voulez, un exécutant de base.

Vous pouvez alors commander à cet exécutant de charger dans sa mémoire d'autres capacités sous forme d'un programme supplémentaire, un programme de traitement de texte, par exemple. Après cette opération, vous êtes devant un exécutant "traitement de texte". Attention, l'expression **charger en mémoire** a le sens précis de **faire une copie dans la mémoire centrale d'informations se trouvant dans une mémoire externe**.

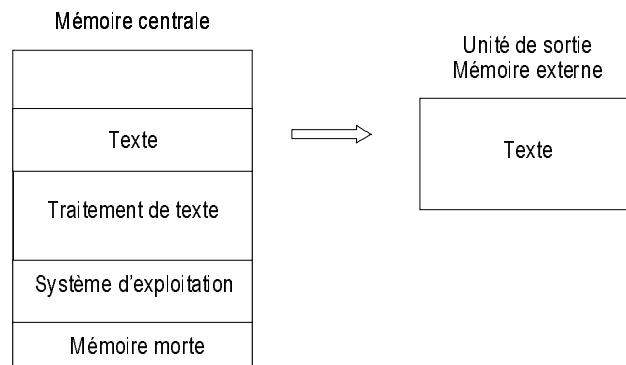
Cet exécutant, vous pouvez le faire travailler dans le sens où vous lui fournissez des informations, disons des données, sous forme d'un texte que vous encodez au clavier. Ces données se retrouvent également dans la mémoire centrale.

Conséquence: **dans la mémoire centrale**, se retrouvent aussi bien des **programmes** qui confèrent à l'exécutant une intelligence très artificielle (pensez au côté formel des traitements réalisables) que des **données**: un texte, un dessin, un air de musique... et toute information également numérisable. Il en sera donc de même pour les mémoires externes: un disque dur, une disquette, un cédérom... Les uns comme les autres contiendront deux types d'informations: des programmes et des données.

La suite des opérations peut globalement consister en deux choses: vous souhaitez travailler avec un autre exécutant ou vous souhaitez cesser le travail. Dans l'un comme dans l'autre cas, des précautions sont à prendre.

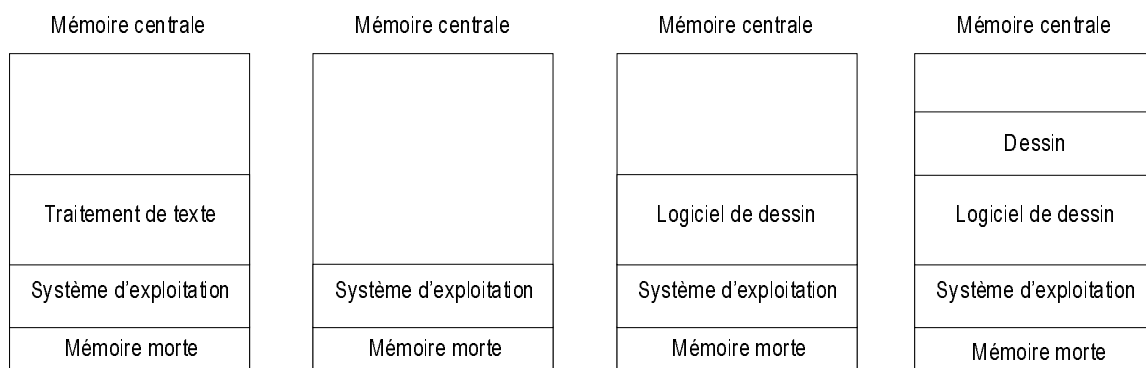
Pour travailler avec un autre exécutant, il faut que ses "capacités" soient présentes en mémoire sous forme d'un autre programme, un logiciel de dessin, par exemple. Si vous ne souhaitez plus travailler avec l'exécutant précédent, une bonne chose est de le faire disparaître de la mémoire centrale, lui et le travail qu'il a permis de faire. C'est l'exécutant "système d'exploitation" qui va prendre cette tâche en charge. Rappelez-vous que la mémoire centrale est une mémoire de travail et que les informations qu'elles contiennent sont de toute façon vouées à disparaître lors de la mise hors tension de l'ordinateur. La disparition du programme ne pose pas de problème, puisqu'il s'agit d'une copie. Celle des données est plus gênante, car aucune copie de ces données

n'existe sur un support non "volatile". Vous devez alors faire exécuter par le système d'exploitation une opération de sauvegarde des données en mémoire externe (le plus souvent disquette ou disque dur) avant de lui donner l'ordre de vider les données, voire le programme de la mémoire centrale.



La sauvegarde effectuée, il faudra demander au système d'exploitation de décharger le texte (les données) et le traitement de texte (le programme) avant de faire procéder au chargement d'un autre programme. Qui dit nouveau programme dit nouvel exécutant donc autre type de tâche (par exemple, du dessin).

Quelques commentaires importants sont à faire. D'abord, la sauvegarde et le déchargement du texte et du traitement de texte ne sont pas obligatoires pour pouvoir travailler avec un autre exécutant. Il est possible de mettre l'exécutant "traitement de texte" en sommeil et de le rappeler plus tard. Toutefois, si une tâche est terminée, c'est une bonne habitude que de la clôturer de la manière que nous venons de décrire.



Il faut également dire un mot de l'exécutant "système d'exploitation". Comme le montrent les schémas qui précèdent, dès qu'il est mémoire, il y reste jusqu'à la fin de la session de travail. C'est assez logique dans la mesure où les fonctionnalités de base doivent rester disponibles.

Si comme nous l'avons dit, la vocation d'un système d'exploitation est essentiellement de permettre d'exploiter les ressources, il faut préciser qu'aujourd'hui, on lui adjoint souvent des

fonctionnalités liées à la gestion d'un environnement graphique. Cet environnement assure un confort d'utilisation dans la mesure où de nombreuses manipulations peuvent être réalisées de manière dynamique (en faisant glisser la souris, par exemple). Quelquefois, le programme de gestion de l'environnement graphique est séparé du système d'exploitation (*Windows 3.0, 3.1, 3.11*) quelquefois, il y est intégré (*Windows 95*). Plus rarement, on travaille sans environnement graphique avec un système d'exploitation fonctionnant en mode texte (*MS-DOS, Disk Operating System*).

Notez ici quelques indications à propos du système d'exploitation (et de l'environnement graphique) que vous utilisez à l'école, à domicile

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Indiquez ici les commandes qui vous permettent de charger un programme en mémoire centrale. Comment faire pour sauvegarder des données, pour décharger un programme de la mémoire centrale, pour terminer correctement une session de travail? Notez aussi le vocabulaire et les concepts spécifiques au système d'exploitation et à l'environnement graphique utilisés.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

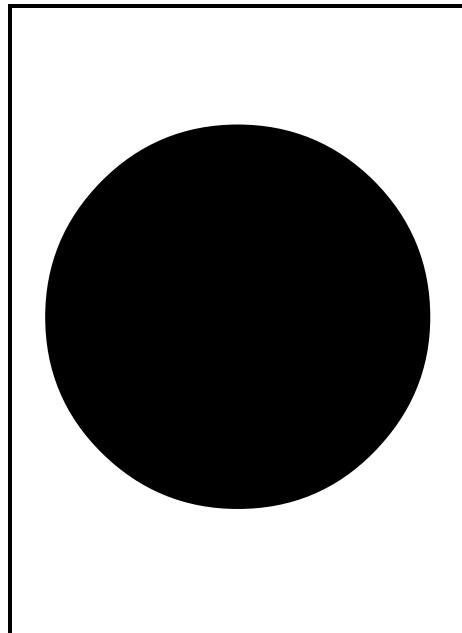
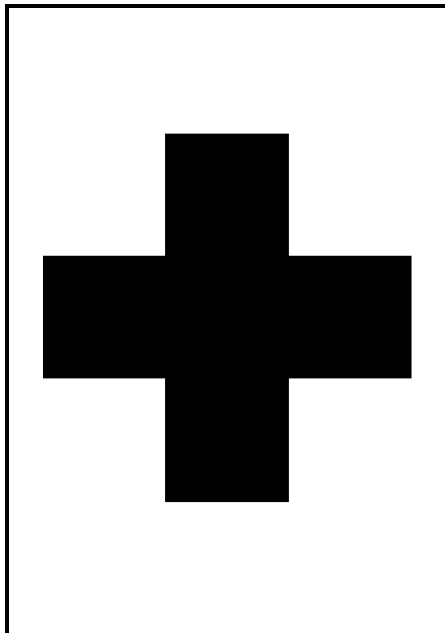
3. L'exécutant "dessin par points"

A. Comment numériser une image?

Vous avez imaginé comment numériser des caractères pour faire en sorte qu'un ordinateur soit à même de traiter des textes. Posez-vous maintenant la même question à propos d'une image.

Vous désirez communiquer des images à un correspondant, mais vous devez le faire par l'intermédiaire d'un dispositif qui n'accepte que des chiffres. Comment pourriez-vous procéder?

A titre d'exercice, voici deux images très simples, sur une feuille de papier. Pouvez-vous imaginer une technique, un dispositif qui permette de convertir ces images en une suite de chiffres que quelqu'un d'autre pourra reconvertir pour reconstituer les images?



.....

.....

.....

.....

.....

.....

.....

.....

Ne tournez pas la page si vous désirez rechercher vous-mêmes une solution.

Voici une manière de procéder:

préalablement la couleur. Quelle sera la stratégie pour (faire) dessiner un rectangle rouge? La logique veut que vous choisissiez la couleur rouge dans la palette de couleurs, que vous choisissiez un outil vous permettant de dessiner des cercles (l'ordre de ces deux opérations a relativement peu d'importance), puis que vous définissiez, au moyen de la souris, une zone d'une certaine forme dont les pixels auront cette couleur.

Si par la suite, vous décidez de dessiner un rectangle vert qui recouvre totalement ce cercle rouge, c'est un peu comme si vous utilisiez de la peinture, les pixels de couleur rouge ont pris la couleur verte. Le cercle, comme le rectangle n'existent pas en tant que formes géométriques, mais comme ensembles de pixels.

Formes et traits

Les outils de dessin disponibles sont généralement le cercle ou l'ellipse, le rectangle, la ligne droite, la ligne brisée, la ligne courbe... Pour chacun d'eux, il importe de faire **préalablement** des choix comme: la couleur, l'épaisseur et le style du trait, la couleur ou le motif de remplissage...

La stratégie de dessin d'une forme ou d'un trait est donc la suivante:

- choisir l'outil qui correspond à la forme ou au trait
- choisir les attributs de la forme ou du trait
- définir la zone (au moyen de la souris)

Existe-t-il d'autres outils? Si oui, ils doivent procéder de la même manière à savoir, modifier la couleur des pixels d'une zone définie par l'utilisateur. C'est le cas de la **gomme**, un outil dont la taille est un attribut à fixer préalablement. Le principe est que les pixels situés dans les zones parcourues par la gomme voient leur couleur modifiée. On peut encore citer le pinceau, le crayon, l'aérographe,... qui sont des outils possédant comme les autres plusieurs attributs à déterminer avant toute utilisation.

Remplissage

Nous avons insisté sur la nécessité de fixer les attributs d'un trait ou d'une forme avant de les dessiner. Il existe une possibilité de corriger un oubli, c'est d'utiliser avec précaution l'outil de remplissage. Celui-ci entraîne la modification de la couleur de tous les pixels adjacents de même couleur que celui qui est pointé avec la souris. Attention, il arrive qu'une image se colorie de manière uniforme après l'utilisation de cet outil!

Texte

Il existe un outil un peu plus particulier qui est l'outil texte. Il est vrai qu'une image, une affiche, un dessin sont souvent agrémentés d'un bout de texte. L'outil texte permet d'éditer du texte dans un cadre, c'est-à-dire de l'encoder et d'en modifier le contenu et la forme. Toutefois, lorsque cette édition est terminée, le texte est devenu un ensemble de pixels. Il n'est donc plus possible de l'éditer à nouveau. Il faut utiliser la gomme pour l'effacer.

Comment réparer les gaffes?

Il existe plusieurs manières de corriger des erreurs dues à la maladresse (manque d'habileté dans l'utilisation de la souris), à de mauvais choix d'attributs, à de mauvaises appréciations. Nous

avons déjà parlé de la gomme qui est loin de résoudre tous les problèmes. Il existe un moyen plus radical. Puisque l'ordinateur mémorise (même temporairement dans la mémoire centrale), l'exécutant est de plus en plus souvent capable de mémoriser les dernières actions qu'il a effectué pour vous. De la sorte, il lui est donné de pouvoir reproduire la (les) situation(s) antérieure(s). Le vocable utilisé est **Défaire** ou **Annuler**. Inutile de dire qu'un utilisateur patenté use et abuse de cette opportunité qui n'est d'ailleurs pas l'apanage des logiciels de dessin.

S'il s'agit de petites retouches à faire au dessin, il existe une autre façon de procéder, propre aux logiciels de dessin par points, c'est d'employer le **zoom** pour modifier l'image pixel par pixel. L'image étant un ensemble de pixels, il est normal que la plus petite partie du dessin sur laquelle il soit possible de travailler soit le pixel.

Sélectionner, couper, copier, coller

Vous vous rendez compte que ce type d'exécutant va exiger que vous soyez assez sûrs de vous quant à la position des formes et des traits et aux attributs que vous leur donnez. Malgré tout, il vous sera possible de "faire bouger" certaines choses dans une certaine limite, grâce à un autre principe qui n'est pas non plus propre aux logiciels de dessin mais plutôt au système d'exploitation qui est toujours en arrière-plan. La technique du **sélectionner-couper-coller** ou du **sélectionner-copier-coller** est inspirée de ce que vous pourriez faire pour créer une affiche: découper des parties de dessin ou les polycopier et les assembler à votre guise. L'élément virtuel qui permet à nos exécutants de travailler de la sorte s'appelle le **presse-papiers**. Il vous est loisible de sélectionner une partie du dessin, de l'envoyer dans le presse-papiers puis de le récupérer. Seul ennui, le presse-papiers ne peut contenir qu'une seule sélection à la fois.

L'opération qui consiste à envoyer la sélection dans le presse-papiers en la faisant disparaître s'appelle **couper**. La même opération sans faire disparaître la sélection s'appelle **copier**. La récupération du contenu du presse-papiers s'appelle **coller**.

La sélection, avec un logiciel de dessin par points, se fait au moyen de la souris, soit à main libre, soit en zone rectangulaire.

Indiquez ici les détails qui vous paraissent importants concernant les commandes spécifiques du logiciel de dessin par points que vous utilisez: son nom, les outils disponibles, certaines commandes ou primitives particulières ou intéressantes

.....

.....

.....

.....

.....

.....

.....

.....

4. L'exécutant "dessin vectoriel"

A. Une solution alternative de numérisation

Vous vous doutez bien que la transformation d'un dessin en pixels entraîne une numérisation très gourmande. Numérisé, un dessin qui occupe une page de format A4 est donc beaucoup plus volumineux qu'une page de texte numérisée du même format. Un autre inconvénient ne vous a pas échappé à l'utilisation: il en est du dessin par points comme de la peinture, tout nouveau trait, toute nouvelle "couche" vient irrémédiablement recouvrir l'ancienne (si l'on excepte la possibilité de plus en plus courante de demander des "retours en arrière" puisqu'on peut programmer l'exécutant pour qu'il mémorise un certain nombre des états antérieurs). Il n'est en tous cas pas possible de modifier les formes, mais plutôt de "retoucher" éventuellement le dessin, pixel par pixel.

Pourriez-vous imaginer une autre manière plus économique de coder (transformer en chiffres) les deux dessins qui précèdent? Décrivez-la du mieux que vous pouvez.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Quelles seraient, d'après vous, les conséquences de cette autre façon de coder sur les primitives attendues des exécutants fonctionnant de la sorte?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

L'exemple du disque noir est le plus simple. Il suffit de préciser sa couleur (au moyen d'un code, bien sûr), la coordonnée de son centre et son rayon et de fournir à l'exécutant la formule mathématique qui permettra de le dessiner.

Celui de la croix est un peu plus complexe. Il faudra y voir, par exemple, deux rectangles superposés pour lesquels il faudra aussi préciser, toujours par exemple, la coordonnée du point de départ, la longueur et la largeur ainsi que la formule de dessin.

Bien sûr, s'il nous fallait aller jusqu'au bout de cette numérisation, cela resterait un peu complexe. Mais vous percevez que c'est du domaine du possible et que cela convient bien à un exécutant dont la vocation est de calculer. Vous devinez aussi que le gain de place lié à ce type de codage est considérable. Un autre avantage non négligeable: le dessin n'étant plus considéré comme un ensemble de points, il sera désormais un ensemble de formes et de lignes dont les attributs (couleur, épaisseur et style du trait, couleur ou style de remplissage,...) pourront être modifiés "a posteriori". Plus encore, les dimensions de ces formes pourront changer puisqu'elles sont représentées à partir de formules.

Une remarque importante toutefois, si le dessin est mémorisé comme un ensemble de formules, l'exécution de ces formules produira à la sortie (à l'écran comme à l'imprimante) un dessin par points.

B. Les primitives des logiciels de dessin vectoriel

Vous admettez que dans ce contexte, un dessin est un ensemble de formes et de traits élémentaires pouvant donc être sélectionnés séparément ou en groupes, essentiellement en vue d'en modifier les attributs.

Formes et traits

On retrouve généralement, et à peu de choses près, les mêmes formes et les mêmes traits que ceux qui sont offerts par les logiciels de dessin par points. La différence se situe surtout au niveau des modifications ultérieures possibles car ces formes et ces traits ne sont plus codés comme des ensembles de pixels. On parlera donc très volontiers d'objets pour les désigner.

Il n'est plus question de gomme. En effet, chaque objet pouvant être sélectionné, il peut tout aussi bien être supprimé. Vous devinez que les stratégies d'édition seront assez différentes de celles utilisées avec les logiciels de dessin par points.

Il n'est plus question de zoom, dans le sens où celui-ci autorise une modification du dessin, pixel par pixel. Si une commande de zoom existe, c'est uniquement pour faciliter le travail de l'utilisateur et lui donner une vue plus ou moins précise de son dessin.

Sélectionner, couper, copier, coller

Il ne s'agit plus, lors d'une sélection, de définir un ensemble de pixels, mais de choisir un ou plusieurs objets. Le *couper, copier, coller* reste évidemment d'actualité.

Il est utile de découvrir les techniques permettant la sélection d'un ou plusieurs objets. Pour un seul objet, un simple clic fait souvent l'affaire. Pour en sélectionner d'autres, l'utilisation conjointe d'une touche du clavier est nécessaire car la sélection d'un nouvel objet entraîne la

suppression de la sélection du précédent. Si les objets sont proches les uns des autres, des techniques de sélection liées à la définition d'une zone (rectangulaire par exemple) sont quelquefois disponibles. Un objet est sélectionné lorsqu'apparaissent des poignées qui permettent de le saisir.

La sélection des objets ne sert d'ailleurs pas qu'au *couper, copier, coller*. Les objets peuvent être déplacés et leurs dimensions modifiées.

Texte

L'outil texte est aussi présent avec, cette fois, l'opportunité d'une édition (insertion, suppression, modification des attributs) puisqu'il s'agit d'un objet texte.

De nouvelles primitives

Nous avons déjà évoqué la possibilité de sélectionner des objets, puis de les déplacer, d'en changer les dimensions. Selon les circonstances, il est quelquefois possible de leur faire subir d'autres transformations telles: les faire tourner ou de les déformer. Lorsque plusieurs objets sont sélectionnés, on doit pouvoir les grouper et à l'inverse les dissocier. Enfin, et ce n'est pas limitatif, vous pouvez aussi jouer sur les différents plans puisque les objets se superposent les uns aux autres. Vous trouverez une description assez détaillée de ces primitives dans [2]

Comment réparer les gaffes?

La gomme n'est plus d'application pour les raisons que nous avons expliquées. En revanche, des options du style *Défaire* ou *Annuler* ont toujours leur raison d'être. Vient s'ajouter à cela la possibilité de supprimer purement et simplement les objets sélectionnés et rien qu'eux.

Le choix du type de logiciel

Selon les travaux à accomplir, le logiciel de dessin par points sera préféré au logiciel de dessin vectoriel ou inversement. Le choix n'est pas irréversible dans la mesure où il existe des logiciels de conversion des dessins d'une catégorie dans l'autre. Toutefois, s'il est assez simple de convertir des dessins vectoriels en points, vous comprendrez que l'inverse est plus complexe et ne fournit pas toujours des résultats très satisfaisants.

Indiquez ici les détails qui vous paraissent importants concernant les commandes spécifiques du logiciel de dessin par points que vous utilisez: son nom, les outils disponibles, certaines commandes ou primitives particulières ou intéressantes

.....

.....

.....

.....

.....

.....

.....

5. Manipuler la souris dans l'environnement graphique

Nous examinerons plus tard avec plus de détails les primitives de l'exécutant "système d'exploitation et environnement graphique". Mais à ce stade de nos développements, il est important que nous relevions les principaux éléments graphiques de cet environnement et les moyens de les utiliser, à savoir: les manipulations de base de la souris pour laquelle cet environnement est prévu.

A. Un peu de vocabulaire

La plupart des environnements graphiques ont comme concept de base la **fenêtre**. Différentes fenêtres peuvent être ouvertes à l'écran sur ce qu'il est convenu d'appeler un **bureau** virtuel. Les fenêtres sont munies de petits dessins ressemblant à des **boutons** qui, lorsqu'ils sont actionnés, ont un effet particulier sur la fenêtre. La manière d'actionner un bouton est d'y déplacer le **pointeur de la souris** avant de **cliquer**. Cliquer est l'opération qui consiste à enfoncer le bouton (bien réel celui-là) de la souris. Lorsqu'il y a plusieurs boutons, c'est habituellement du bouton gauche qu'il s'agit.

La fenêtre contient soit des **icônes** sur lesquelles il est de nouveau possible de cliquer en vue d'ouvrir une nouvelle fenêtre, soit un espace de travail et des éléments graphiques utilisables avec la souris en liaison avec le type d'application, de programme à laquelle la fenêtre correspond.

B. Les manipulations de la souris

L'emploi des objets graphiques est lié à des manipulations de la souris dont les plus courantes sont:

- le clic bouton gauche
- le clic bouton droit
- le double clic bouton gauche
- le triple clic bouton gauche
- le glissé (cliquer et traîner) bouton gauche

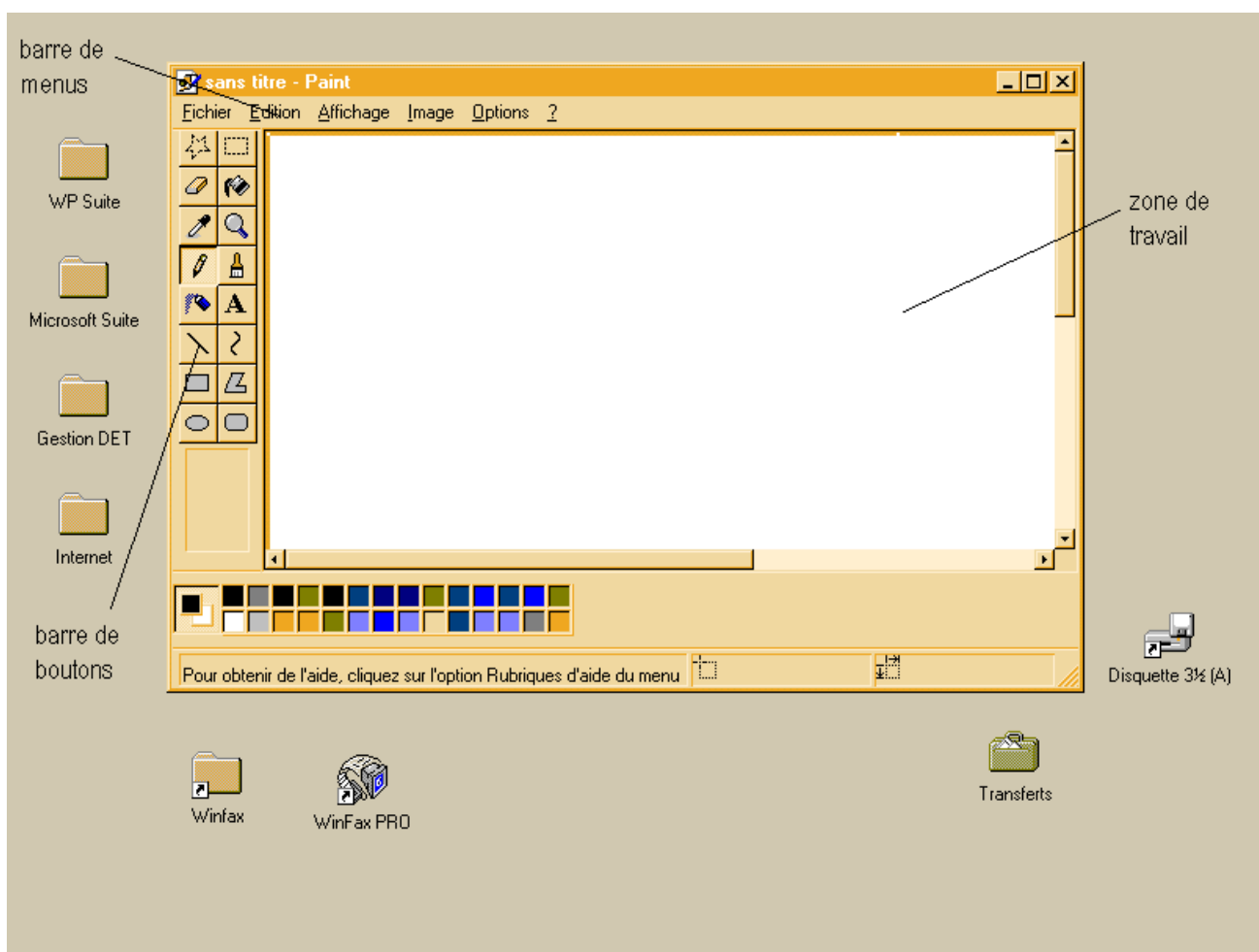
Toutes ces appellations sont loin d'être contrôlées. Elles font plutôt partie d'un jargon. Quant à leur effet, il dépend essentiellement de la façon dont l'exécutant a été programmé. Il arrive que toutes ces manipulations soient sans effet, mais les programmes qui ne reconnaissent pas la souris sont de plus en plus rares. Pour certains, seule une partie de ces manipulations est reconnue. Il est de la responsabilité du concepteur du programme de décider quelles commandes il associe aux diverses utilisations des boutons. Il en est d'ailleurs de même en ce qui concerne les touches du clavier.

Certaines souris sont munies de trois boutons mais les logiciels qui exploitent le bouton central sont quasiment inexistantes. Avec un certain matériel (*Macintosh* de *Apple* par exemple), les souris ne sont équipées que d'un seul bouton.

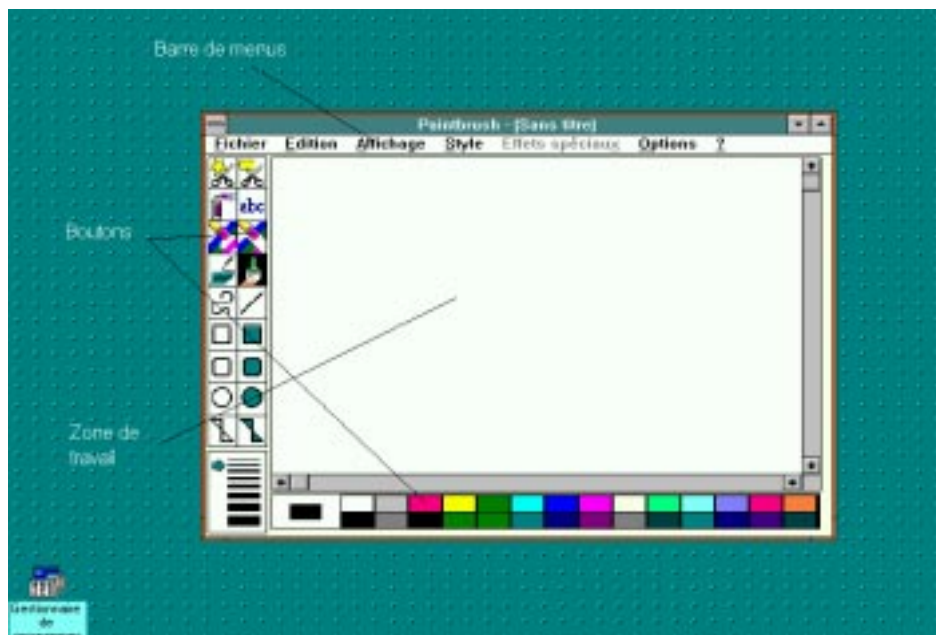
Quelques observations à propos de ces manipulations:

- le double comme le triple clic (lorsque celui-ci est reconnu) sont des manipulations qui demandent une certaine dextérité de la part de l'utilisateur. En réalité, si le délai entre les "clics" n'est pas assez court, ce sont plusieurs simples clics qui sont enregistrés avec les conséquences que vous imaginez.
- le simple clic sur le bouton droit est un moyen utilisé depuis peu pour faire apparaître des menus de commandes dits **menus contextuels**. Ces menus proposent des commandes qui dépendent de l'endroit à l'écran (contexte) où le clic s'est produit.

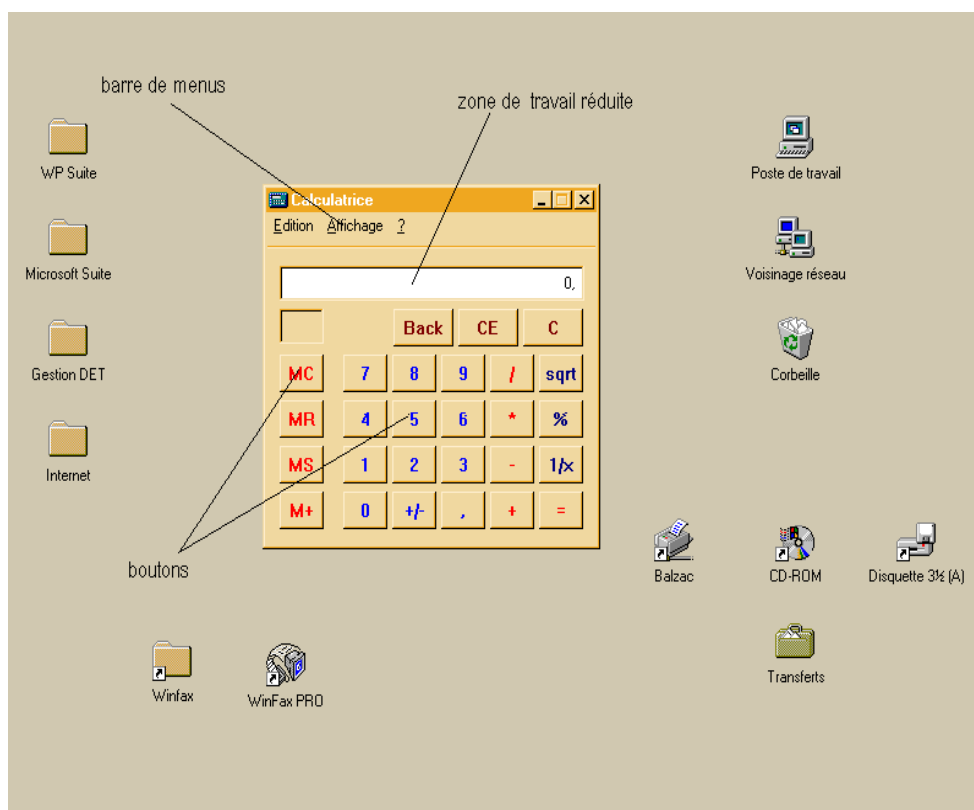
C. Les manipulations de base dans l'environnement graphique



Fenêtre d'application (Windows 95)



Fenêtre d'application (Windows 3.1)

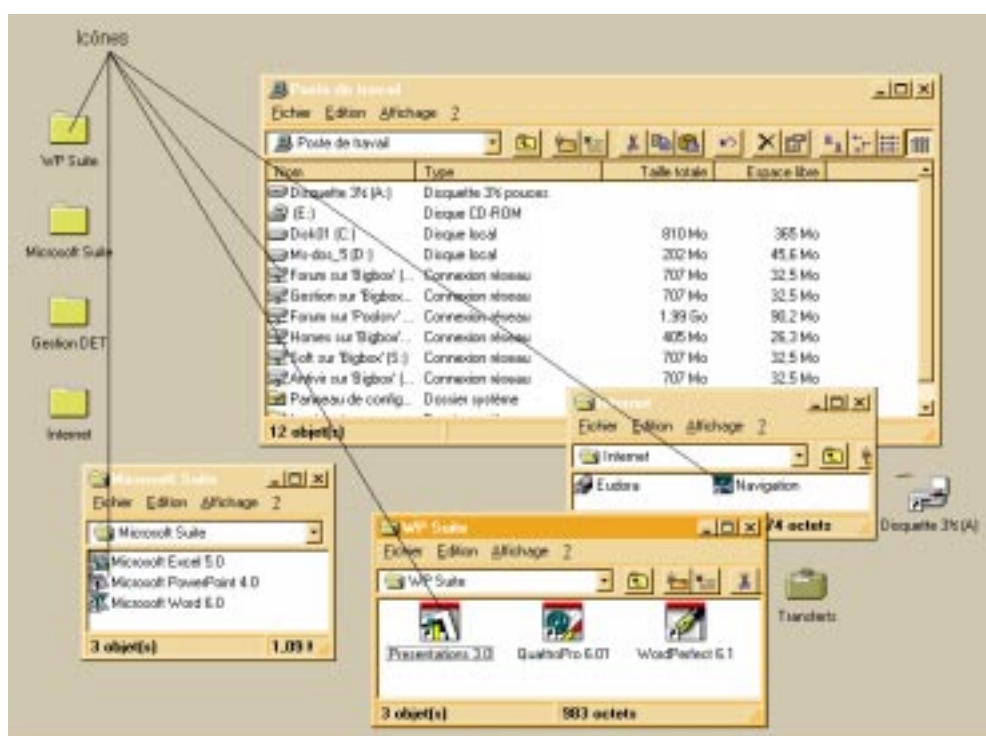


Fenêtre d'application à zone de travail réduite (Windows 95)

L'élément le plus important, celui qui est à la base du développement de ce genre d'environnements, c'est la fenêtre: une zone de l'écran, dont la taille peut être modifiée, qui peut être déplacée, ouverte ou fermée. Généralement, les fenêtres contiennent soit des icônes, soit une zone de travail plus ou moins importante entourée d'outils graphiques spécifiques à un programme. Dans certains cas, la zone de travail est absente dans la mesure où le programme ne nécessite que très peu de données et que celles-ci peuvent être symbolisées par des objets graphiques: programme de gestion d'impression, horloge... On parle de fenêtre d'application et de fenêtres de groupe (d'icônes).

Les fenêtres d'application contiennent presque toujours une zone de travail encadrée par des éléments graphiques permettant de contrôler le programme: barres de boutons, barres de menus...

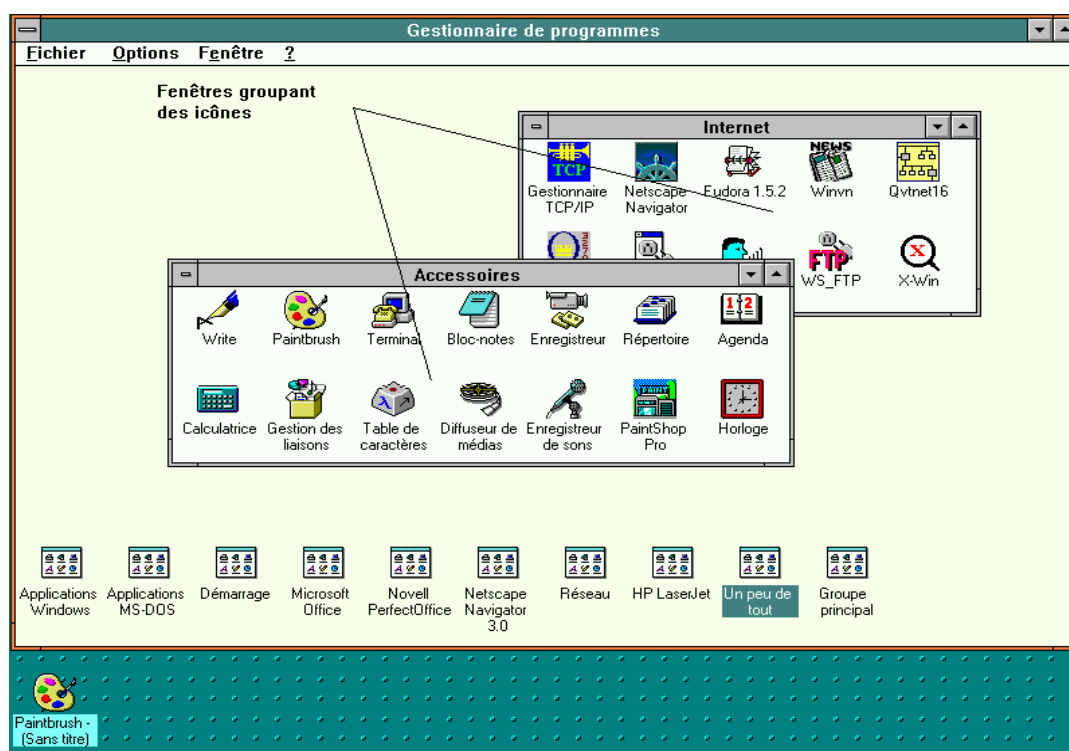
La proportion d'espace occupé dans la fenêtre par la zone de travail est très variable. Dans l'illustration qui précède, il s'agit d'une fenêtre correspondant à un programme de dessin. Dans celle qui suit, le programme dont il est question simule une calculatrice. Comme vous pourrez le constater, il s'agit aussi d'une fenêtre d'application, mais la proportion d'espace occupé par les boutons est nettement plus importante. En d'autres termes, la zone de travail est ici



Fenêtres groupant des icônes sous divers formats (Windows 95)

extrêmement réduite. Dans d'autres cas encore, il n'y aura pas de zone de travail.

Les autres fenêtres contiennent des icônes. Elles symbolisent soit des programmes, soit des **applications** si l'on donne à ce mot le sens de **programme agissant sur des données**. Une icône peut représenter un programme de dessin, de calcul, de traitement de texte mais aussi parfois un texte lié à un programme de traitement de texte, un dessin lié à un programme de dessin etc. Le double clic sur une icône a comme effet de charger en mémoire le programme et, le cas échéant, les données et d'ouvrir la fenêtre associée.



Fenêtres groupant des icônes (Windows 3.1)

En présence d'un environnement graphique dont l'élément fondamental est la fenêtre, vous devez vous familiariser rapidement avec les primitives qui vous permettront de les manipuler avec aisance. Parmi celles-ci:

- l'ouverture
- la fermeture
- le déplacement
- l'agrandissement à tout l'écran
- le rétrécissement à une partie de l'écran
- la réduction
- la modification des dimensions

Lors de quelques-unes de ces manipulations comme lors du travail dans une fenêtre d'application, vous pourrez observer que le pointeur souris change d'aspect. Les nombreux changements d'aspect sont autant d'indicateurs pour l'utilisateur de l'effet possible d'un prochain clic. Par exemple, lorsque le pointeur souris se déplace sur les bords d'une fenêtre, il prend un aspect particulier qui vous indique qu'un glissé de la souris va en modifier les dimensions et dans quel sens.

6. L'exécutant "traitement de texte"

S'il y a des logiciels dont la fréquence d'utilisation est élevée, ce sont bien les programmes de traitement de texte. Rares sont les utilisateurs qui n'y font pas appel, mais rares aussi sont ceux qui en ont un emploi tout à fait efficace. Cela vaut donc la peine que nous réfléchissions à ce que ce genre d'exécutant est capable de nous proposer et à la manière de travailler qu'il risque de nous imposer.

Comme point de départ, tâchez un peu de préciser ce qu'est pour vous un texte, puis, ce que cela risque de devenir pour un exécutant formel comme ceux que nous cotoyons depuis le début de notre réflexion.

Pour vous un texte c'est:

.....

.....

.....

.....

Pour un exécutant formel, cela risque d'être:

.....

.....

.....

.....

A. Le contenu d'un texte

Vous avez déjà envisagé, dans ce document, la possibilité de définir un code de numérisation des caractères les plus courants. Pour rappel, un octet autorise l'emploi de 256 codes différents. Le choix d'un code universel s'impose pour éviter les ennuis d'un résultat illisible au décodage. Notez que des codes unifiés sont apparus (*ASCII*, *ANSI*), mais malgré cela, de légères différences subsistent. C'est assez pour rendre quelquefois les conversions de documents imparfaites.

Cette première observation nous engage donc à considérer qu'**un texte est** avant tout **un ensemble de caractères**. Le terme *suite* serait d'ailleurs plus correct.

Première conséquence: pour encoder du texte, vous êtes amenés à frapper les caractères de ce texte les uns après les autres. Mais si l'utilisateur d'une machine à écrire mécanique doit se préoccuper des passages à la ligne, l'utilisateur d'un traitement de texte est libéré de cette contrainte vu qu'un exécutant formel est capable de les gérer. En réalité, il s'agit d'un calcul qui tient compte de certaines données pouvant être fixées au départ (la grandeur des marges et le format du papier à l'impression notamment). Et nos exécutants sont des calculateurs par excellence. On dit donc couramment qu'avec un traitement de texte, **la frappe est kilométrique**.

Deux observations malgré tout: elles concernent deux caractères particuliers. Le premier, c'est

L'insertion d'un caractère

Le caractère frappé au clavier est inséré à la position courante. Celle-ci est déterminée par un petit symbole graphique discret appelé comme il se doit: **point d'insertion**. Lorsque vous créez un texte, celui-ci est fatalement vide au départ. Sa taille augmente au fur et à mesure que vous frappez les caractères. Lorsque vous souhaitez modifier le texte, vous devez placer le point d'insertion à la position voulue avant de frapper les caractères à insérer. Cela implique que les techniques de déplacement du point d'insertion valent la peine d'être découvertes.

Ne confondez pas *point d'insertion* et *pointeur souris*! Ce n'est pas parce que ce dernier est placé à l'endroit souhaité pour l'insertion que le point d'insertion s'y trouve. La manière la plus simple (mais pas toujours la plus efficace) de déplacer le point d'insertion est de cliquer sur le bouton de la souris après avoir positionné le pointeur souris à l'endroit voulu. Mais il y a bien d'autres moyens que vous pouvez et devez utiliser.

Il est à noter qu'une touche du clavier permet de passer du **mode insertion** au **mode surfrappe** et inversement. Dans ce cas, le caractère frappé remplace le caractère qui suit le point d'insertion. En réalité, il s'agit d'une suppression suivie d'une insertion réalisées toutes deux en une seule frappe. Cette opportunité n'est de mise que dans des circonstances très particulières (frapper des chiffres sur d'autres chiffres, par exemple).

La suppression d'un ou plusieurs caractères

La démarche inverse consiste à supprimer un caractère du texte pour le modifier de manière élémentaire. La position du point d'insertion est tout aussi déterminante. Tous les programmes de traitement de texte exploitent le clavier de la même manière pour autoriser la suppression du caractère qui suit le point d'insertion ou de celui qui précède.

Notez ici comment sont référencées ces touches sur votre clavier (car les appellations parfois varient)

.....
.....

Observez que les mots "*précède*" et "*suit*" font référence au concept de texte séquentiel ininterrompu. Dans ce contexte, le caractère qui précède le premier caractère d'une ligne est le dernier caractère de la ligne précédente. Celui qui suit le dernier caractère d'une ligne est le premier caractère de la ligne suivante.

Dans le même ordre d'idée, il est possible de supprimer un groupe de caractères pour autant que ceux-ci se suivent. On délimite un groupe de caractères de la manière la plus intuitive en cliquant sur le premier caractère du groupe et en traînant la souris jusqu'au dernier (ou l'inverse). Ici aussi, bien d'autres moyens sont de mise dans des contextes particuliers. Nous les évoquons plus loin.

Encore une chose, compte tenu du codage des espaces et des retours forcés comme les autres caractères, certains traitements sont à formuler en conséquence. Ainsi, *regrouper deux mots en un seul, c'est supprimer l'espace qui les sépare*. *Diviser un paragraphe en deux* équivaut à *insérer un retour forcé à l'endroit voulu pour la séparation*. Tout ceci n'est pas de la mise en forme, même si cela y ressemble. En procédant de la sorte, vous agissez sur le contenu du texte,

même si vous avez l'impression que sa forme change.

Une conséquence importante de ce qui précède et une saine attitude d'utilisateur est de **s'interdire l'emploi de l'espace et du retour forcé à des fins de mise en forme du texte.**

Les déplacements automatiques du point d'insertion

Nous avons déjà indiqué une voie qui permet le déplacement du point d'insertion: pointer et cliquer au moyen de la souris. C'est une voie intéressante dans la mesure où le point de départ et le point d'arrivée du bloc à sélectionner sont visibles tous les deux à l'écran. Et même dans ce cas, ce n'est pas forcément la voie la plus simple vu que nous avons à faire à un exécutant qui peut formellement identifier certains blocs de texte particuliers. Cela signifie que si le bloc est formellement définissable, l'exécutant peut très bien être programmé pour en trouver lui-même les limites sur demande de l'utilisateur.

Quels sont d'après vous les blocs de caractères formellement définissables?

.....

.....

.....

.....

.....

Quels sont ceux qui sont effectivement reconnus par votre exécutant traitement de texte dans le cadre du déplacement du point d'insertion? Inspirez-vous du système d'aide en ligne.

.....

.....

.....

.....

.....

La notion fondamentale ici est la notion de délimiteur ou de **séparateur**. L'exécutant reconnaît formellement certains blocs de caractères parce que ceux-ci sont limités d'une manière ou d'une autre.

S'il existe des variantes d'un logiciel à l'autre, il y a des blocs particuliers qu'un utilisateur ne peut de toute manière ignorer. Le **mot** est un de ceux-là. Il faut lui reconnaître une définition qui n'est pas tout à fait celle que nous avons dans la tête. Il n'est pas possible de donner la définition formelle du mot. On peut cependant considérer que l'espace est un de ses séparateurs potentiels. Il est intéressant d'observer comment se comporte l'exécutant dans les situations les plus diverses (signes de ponctuation, chiffres, autres caractères particuliers...)

Exercez votre perspicacité à rechercher ce que votre exécutant considère comme un mot. Pour votre traitement de texte, un mot c'est...

.....

.....

.....

.....

Un autre bloc fondamental dans le traitement des textes est le **paragraphe**. Là aussi, la définition formelle est quelque peu différente de la définition habituelle. Ainsi, si l'on estime en littérature qu'un nouveau paragraphe marque le développement d'une nouvelle idée, en traitement de texte, il est formellement défini par le fait qu'il se termine toujours par un retour forcé (et cela, même si quelquefois il est possible de "prolonger" le paragraphe au-delà d'un retour forcé). La notion de paragraphe est surtout intéressante pour des questions de mise en forme dont nous parlerons plus loin.

D'autres blocs particuliers sont couramment reconnus tels: la ligne, la page, l'écran, etc. Il n'est donc pas étonnant de trouver des commandes qui permettent d'envoyer le point d'insertion au début ou à la fin d'une ligne, au début du mot ou du paragraphe suivant, au début de la page suivante, au début ou à la fin du document, etc..

Les sélections automatiques des blocs de caractères

Ce qui vient d'être dit est particulièrement utile dans le contexte de la sélection d'un bloc de caractères. Si la sélection est appréciée dans le cadre de la mise en forme d'un document, elle l'est d'abord dans le cadre de la modification de son contenu à savoir: la suppression, le déplacement, voire la copie de blocs de texte.

Les blocs particuliers que nous avons cités peuvent être repérés par l'exécutant lui-même, pour autant que l'utilisateur le lui demande. Ainsi, vous pourrez sélectionner un mot, un paragraphe, une ligne,... en un minimum de temps, et sans avoir à préciser vous-mêmes où se trouvent leur début et leur fin. Vous augmenterez considérablement l'efficacité de votre travail si vous avez conscience de l'existence de ces primitives et si vous les mettez rapidement en pratique.

Comment pouvez-vous sélectionner rapidement les blocs particuliers les plus courants avec votre traitement de texte?

.....

.....

.....

.....

.....

.....

Couper, copier, coller

La modification du contenu d'un texte passe donc par la sélection de blocs de texte en vue de les supprimer, les déplacer ou les reproduire. Le déplacement et la reproduction ne sont pas des primitives nouvelles. Nous les avons déjà rencontrées lors du travail avec les exécutants de type "dessin". Elles s'appellent familièrement *couper-coller* et *copier-coller* et font usage du presse-papiers.

Des commandes dans le menu d'édition du logiciel, des boutons dans les barres d'outils, des raccourcis clavier existent pour réaliser ces opérations. En ce qui concerne le déplacement (*couper-coller*), la plus dynamique des manières de faire consiste à traîner au moyen de la souris le bloc sélectionné jusqu'à l'endroit voulu. A nouveau, cette manoeuvre ne présente d'intérêt que lorsque le bloc sélectionné et l'endroit final du déplacement sont tous deux visibles à l'écran. Dans les autres cas, d'autres stratégies sont plus adaptées.

La commande *coller* insère le contenu du presse-papiers à la position du point d'insertion et cela, autant de fois que vous le souhaitez tant que le contenu du presse-papiers n'évolue pas.

C. La forme du texte

Il existe deux facteurs, parfois complémentaires, parfois contradictoires qui conditionnent les effets d'un texte sur son lecteur. C'est, d'une part, la qualité de son contenu et d'autre part, le soin apporté à sa mise en forme. D'aucuns considèrent que ces facteurs interviennent de manière égale. Nous serons plus nuancés en reconnaissant que bien souvent, il existe une relation de proportionnalité inverse entre la quantité de soin apportée à la rédaction du contenu et la quantité de soin apporté à sa forme. Il suffit pour s'en convaincre d'analyser certains quotidiens ou périodiques pour se rendre compte que les revues à sensation mettent davantage l'accent sur la forme alors que les revues d'un certain niveau intellectuel se préoccupent davantage des idées et choisissent des mises en page plus sobres.

Quels que soient vos choix en la matière, il importe que vous reconnaissiez les éléments qui font partie de cette mise en forme et comment il vous est possible d'en modifier les valeurs.

Observez diverses publications et citez des éléments qui vous paraissent déterminants dans la mise en forme. Pensez, en les citant, que vous travaillez avec des exécutants spécialisés dans le calcul.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tous les éléments que vous pouvez citer font partie de deux catégories de paramètres: ceux qui agissent sur la forme des caractères, et ceux qui conditionnent leur disposition sur le support d'impression (papier, transparent...), voire le support d'affichage (écran).

Comment ces paramètres sont-ils enregistrés? Vous savez que chaque caractère possède son code, mais il existe aussi des codes de contrôle, essentiellement destinés aux périphériques (écran, imprimante,...) et mémorisés à chaque fois qu'une commande de mise en forme est donnée. Ces codes sont rarement visibles à l'utilisateur qui, pour en annuler les effets, doit alors donner une nouvelle commande. Mais voyons d'abord quels sont les paramètres essentiels de la mise en forme.

D. La forme des caractères

Assez simplement, on peut dire que cette forme dépend de trois choix:

- la police,
- la taille,
- le style.

La police

Bien avant que l'imprimerie ne soit inventée, l'Homme s'est intéressé à diverses manières de dessiner les symboles d'écriture. De grandes familles de caractères sont nées, en ce sens que des artistes ont donné aux lettres de l'alphabet, des formes qui leur conféraient un air de ressemblance. Dès l'instant où des procédés d'impression ont été mis en oeuvre, ces formes ont été conservées sur des supports en plomb et étendues à des caractères spéciaux (typographiques).

On parle de **polices de caractères** pour désigner ces différentes familles. Celles-ci portent des noms quelquefois évocateurs tels: *Times, Swiss, Monotype, Humanist, Courier,...* C'est souvent l'histoire et les circonstances qui ont été à la base de leur nom.

Aujourd'hui, les avantages de la numérisation sont tels que les supports nécessaires à ces symboles sont devenus virtuels et n'exigent, comme place pour être conservés, qu'un peu de mémoire. La création d'une nouvelle famille est presque à la portée du premier venu, même si celui-ci ne développe pas particulièrement des qualités artistiques. Ce n'est donc pas étonnant que de nouvelles familles se soient développées de manière pléthorique et que les contre-façons soient nombreuses.

Si la chose vous intéresse, vous trouverez des éléments historiques intéressants dans [3] et également dans [4], d'autant que les notions de familles et de polices de caractères ne sont pas tout à fait équivalentes.

Il fut un temps où les seules polices disponibles étaient celles que possédaient les imprimantes dans leur petite mémoire. Depuis, on utilise rarement les polices d'imprimantes et on demande plutôt à ces dernières de dessiner les caractères selon des modèles contenus dans des **fontes** sous forme de fichiers enregistrés sur des supports externes. Aujourd'hui, sur les disques durs des ordinateurs, sont enregistrées plus de polices (fontes) de caractères que nécessaire à une mise en page décente. Il convient donc de les utiliser avec parcimonie. On veillera particulièrement à ne pas utiliser plus de deux polices différentes dans un texte (une pour les titres, une autre pour le corps de texte) au risque, sinon, de fatiguer l'oeil du lecteur.

Les ouvrages cités en bibliographie vous prouveront que plusieurs distinctions sont à faire entre les polices de caractères. Nous n'en retiendrons qu'une, mais elle est importante. Il existe des polices **avec empattement** et **sans empattement** (on dit quelquefois *sans serif*). En voici des exemples:

Police sans empattement: Arial : **ABCDEF abcdef**

Police avec empattement: Times New Roman: **ABCDEF abcdef**

Si l'empattement donne beaucoup de relief au caractère, il est quelquefois exubérant. Il convient donc de le choisir intelligemment et en fonction des circonstances. Signalons aussi que dans de faibles tailles, le caractère avec empattement devient difficilement lisible.

ABCDEF abcdef

Dans de grandes tailles, il manque parfois de discrétion.

La taille

Nous venons déjà de l'évoquer, la taille du caractère revêt une importance particulière. Elle est exploitée pour mettre en évidence certains contenus et particulièrement, les titres ou encore le chapeau d'un article.

La numérisation des polices a considérablement simplifié les choses en ce sens que le choix de la taille est à peu près équivalent au choix d'une échelle pour la représentation des dessins qui constituent les différents caractères. En d'autres termes, si jadis on évoquait des paramètres comme la **hauteur**, la **chasse** (largeur), le **corps** (hauteur de la "bande" occupée par l'ensemble des lettres), aujourd'hui, on ne parle plus guère que de police et de **taille**. Par exemple, une police de chasse plus petite portera un autre nom (*Arial* et *Arial Narrow*).

Depuis longtemps, la taille des caractères se mesure en **points**. Le point typographique ou point Didot vaut 0,3759 mm. La numérisation a entraîné le choix d'une unité de mesure plus anglo-saxonne du point: 0,351 mm.

Arial 10 points

Courier New 20 points

Times New Roman 30 points

Le seuil de lisibilité se situe aux alentours de 10 points. Généralement, pour le corps de texte, le choix de la taille se situe entre 10 et 13 points. Au-delà, les textes sont plutôt destinés aux personnes dont la vue est mauvaise ou aux enfants. En deçà, la lecture apparaît difficile, même si elle est possible pour certains.

Le style

Certaines parties de texte méritent d'être mise en évidence pour des raisons diverses: titres, mots importants, mots ou expressions à dégager du texte, mots ou expressions ayant un statut particulier. La taille n'est pas le seul moyen disponible pour réaliser cette mise en évidence. D'ailleurs, dans le corps du texte, jouer sur la taille n'est pas très souhaitable. On choisit alors de donner à ces blocs de caractères des attributs particuliers. Les plus courants et les mieux reconnus par le lecteur dans la signification qu'ils peuvent avoir sont: le **gras**, l'**italique**, le **souligné**. Il en existe bien d'autres, mais à ne pas utiliser en doses massives. Ces attributs peuvent être cumulés.

Texte en gras

Texte en italique

Texte souligné

Texte en ombré

Texte en gras et italique

Texte en italique double soulignement

Avec votre traitement de texte, de quelle(s) manière(s) pouvez-vous choisir les paramètres que nous venons d'évoquer?

.....

.....

.....

.....

.....

.....

Assez paradoxalement, si l'exécutant traitement de texte ne se préoccupe pas du sens, c'est formellement que celui-ci apparaît.

Vous ne serez pas étonnés de constater que la plupart des programmes de traitement de texte vous offrent de fixer ces trois paramètres ou catégories de paramètres dans un même menu et sous une même rubrique. Ce menu est souvent désigné par le mot *format* car il s'agit de fixer la *forme* du caractère, alors qu'un menu *édition* contiendra l'essentiel des rubriques correspondant aux primitives de modification du contenu.

Quels sont les attributs de style que vous pouvez modifier?

.....

.....

.....

.....

.....

Il va de soi que **la modification de certains paramètres, pour un bloc de texte donné, passe par la sélection préalable de ce bloc de texte**. Un ensemble choisi de paramètres peut être enregistré sous un nom et réutilisé pour d'autres blocs de texte. Cela permet d'augmenter l'efficacité du travail de mise en forme et de modifier ceux-ci par la suite et d'un seul coup pour tous les blocs de caractères ayant le même statut. On parle alors de définir des **styles de texte**, mais cette appellation n'est pas à confondre avec celle dont nous venons de parler qui est plus restrictive.

La meilleure façon de travailler est sans doute de définir progressivement (de jour en jour) une ou plusieurs séries de styles selon les types de document à rédiger de manière à constituer des **modèles**. A la longue, la mise en page du texte peut être exécutée rapidement en appliquant les styles voulus aux blocs concernés après que le contenu ait été finalisé. Une mise en forme minimale peut cependant avoir lieu en cours d'encodage.

E. La disposition sur le support (ou la gestion de l'espace disponible)

La forme des caractères n'est pas tout ce que la mise en page permet de faire. Une autre grande décision à prendre par le "metteur en page", est de disposer le texte sur le support utilisé, le plus souvent, une feuille de papier de format fixé.

Dans ce domaine, il serait sans doute possible de décrire beaucoup de choses, surtout en ce qui concerne la possibilité d'insérer des éléments graphiques particuliers pour habiller le texte: graphiques, tableaux, images, schémas,... Dans le cadre strict des notions de base que nous voulons évoquer, nous nous contenterons de parler des paramètres principaux permettant de cantonner le texte dans des zones particulières du support.

Le format du papier

Nous en avons déjà parlé, c'est l'élément le plus important au départ puisqu'il conditionne l'organisation du document en pages. S'il est évident que l'exécutant gère automatiquement les retours à la ligne, il gère aussi bien les fins de page (à moins que vous n'en décidiez autrement). Mais il ne peut les gérer qu'en fonction du format du papier.

Il arrive rarement que vous deviez préciser ce format dans la mesure où, par défaut, c'est le format A4 qui est de mise. Cependant, deux dispositions sont possibles: classiquement, en **portrait** (ou à la française) et de manière plus originale, en **paysage** (ou à l'italienne).



Portrait



Paysage

Les marges

Les marges supérieure et inférieure, de gauche et de droite, donnent un cadre strict au texte. Ces paramètres associés aux choix liés aux caractères permettent à l'exécutant de gérer la disposition globale.

Les espacements

Divers espacements peuvent être fixés. Si les marges se mesurent généralement en centimètres, les espacements se calculent habituellement en points.

Par ordre d'intérêt décroissant citons:

- l'espacement des paragraphes,
- l'espacement des lignes ou **interligne**,
- l'espacement des caractères.

Fixer un espacement avant ou/et après un paragraphe est utile si on veut gérer cet espacement de manière uniforme dans tout le document. Il est en tous cas préférable de le fixer, plutôt que d'utiliser des marques de fin de paragraphe dont ce n'est pas le rôle.

Il devient rare de modifier l'interligne d'un texte (pratique héritée de l'emploi des machines à écrire). Dans certains cas particuliers, et pour certains documents, cette opportunité reste intéressante.

Il est encore beaucoup plus rare qu'un utilisateur débutant modifie lui-même l'espacement entre les caractères voire entre certains caractères (crénage). C'est une pratique que l'on réservera aux spécialistes de la mise en page.

Il est fondamental de connaître l'étendue du texte auquel s'appliquent des paramètres de ce type. Par exemple: dans quelles limites le choix d'un nouvel interligne est-il valable? Cela dépend bien entendu de votre programme de traitement de texte.

Les retraits

Une mise en page, fut-elle sobre, se passe difficilement des retraits de certains paragraphes et, dans certains cas, d'un retrait de la première ligne. Ces retraits permettent une indentation du texte qui est aussi porteuse de sens pour le lecteur. Il est à noter que ces retraits peuvent se faire en avant comme en arrière (négativement et positivement), à gauche comme à droite. Ils sont mesurés par rapport aux marges de gauche et de droite. En voici quelques illustrations:

Retrait positif de paragraphe (à gauche)

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
 Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
 Du texte Du texte Du texte Du texte Du texte Du texte

Retrait positif de paragraphe (à gauche et à droite)

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
 Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
 Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
 Du texte Du texte Du texte

Retrait positif de première ligne

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte

Retrait positif de paragraphe (à gauche) et retrait négatif de première ligne

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte

Comme vous pouvez le constater, il existe beaucoup de manières de réaliser des indentations.

L'alignement

La gestion des caractères dans la ligne est du ressort de l'exécutant traitement de texte, nous l'avons déjà dit. Celui-ci peut donc veiller à une répartition des caractères à partir de la marge de gauche, de celle de droite ou du centre des marges. Il peut faire mieux: en rajoutant des portions d'espaces (à la manière des typographes d'hier), il peut faire en sorte que l'alignement soit parfait, tant à droite qu'à gauche. On parlera donc respectivement d'alignement à **gauche**, à **droite**, **centré** et **justifié**.

Alignement à gauche

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte

Alignement à droite

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte

Alignement centré

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte

Alignement justifié

Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte Du texte
Du texte

Comme pour l'espacement, il incombe à l'utilisateur de savoir à quelles étendues de texte s'appliquent les paramètres de retraits et d'alignement.

D'autres paramètres et dispositions originales

Les paramètres que nous venons de citer ne sont pas les seuls que vous pouvez fixer pour modifier la disposition du texte. Nous devons encore mentionner les possibilités, pour un tel exécutant, de gérer les fins de pages en évitant les **lignes veuves et orphelines** (pas de ligne isolée d'un paragraphe en fin ou en début de page) et en rendant, si nécessaire, des lignes ou des paragraphes solidaires.

Il existe des dispositions de texte plus originales, telles celles que vous pouvez trouver dans un journal, une revue, un magazine... Dans ce domaine aussi, il existe de nombreuses opportunités. Ne citons par exemple que la répartition de texte en colonnes sur tabulations, la disposition en colonnes de type journal et, bien entendu, l'usage des tableaux. Mais leur description dépasse largement le cadre de cette approche de base. Vous trouverez de nombreuses idées à ce propos dans [5] et des exercices à réaliser dans [6].

Indiquez ici la manière de déterminer, avec votre traitement de texte, les paramètres dont nous avons parlé. Indiquez aussi à quelle étendue chaque paramètre s'applique.

Format du papier:

.....
.....
.....
.....
.....
.....
.....

Marges:

.....
.....
.....
.....
.....
.....
.....

Espacement:

.....
.....
.....
.....
.....
.....
.....

Retraits:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Alignement:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

7. L'exécutant "système d'exploitation"

Tous les exécutants déjà évoqués ne pourraient exister sans celui-ci. Nous avons coutume de l'appeler *exécutant de base* car il est à la base de toute autre transformation. En d'autres termes, c'est notamment grâce à lui que d'autres programmes peuvent se retrouver dans la mémoire centrale de l'ordinateur.

A. Système d'exploitation et environnement graphique

Quelle est l'étendue de l'appellation "système d'exploitation"? L'expression désigne un ensemble de programmes (système) qui rendent le processeur capable de gérer, d'exploiter les ressources matérielles mises à sa disposition. L'utilisateur doit pouvoir communiquer avec l'exécutant "système d'exploitation" (en anglais: *Operating System* et en abrégé *OS*). La manière de donner les commandes peut varier très fort en fonction de l'exécutant, c'est-à-dire en fonction du type de machine et en fonction du système d'exploitation lui-même.

Dans certains cas, ce mode de communication est très rudimentaire. La syntaxe des commandes est assez rigide et celles-ci ne peuvent être données qu'en mode texte. L'exemple le plus célèbre pour illustrer est celui de *MS-DOS* (*Microsoft Disk Operating System*) qui est resté très longtemps le système d'exploitation le plus utilisé sur les ordinateurs de type PC et compatibles.

Dans d'autres cas, le système d'exploitation est doublé d'une série de programmes permettant à l'utilisateur d'évoluer dans un environnement graphique dans lequel la souris est un périphérique important. C'est le cas, depuis qu'ils existent, du système d'exploitation tournant sur les ordinateurs de type *Macintosh* de *Apple* par exemple.

Pour éviter les inconvénients des OS peu conviviaux, des programmes de gestion d'environnement graphique ont été écrits. Chargés en mémoire par l'OS lui-même, ces programmes donnent l'illusion de travailler avec un système qui intègre les commandes graphiques. C'est le cas de *Microsoft Windows* (3.0, 3.1 ou 3.11) qui est chargé par *MS-DOS*. Quant à *Windows 95*, s'il donne l'impression d'un système intégré, il offre aussi des possibilités de retour à une version *DOS*.

Tout laisse cependant à croire que désormais, les systèmes d'exploitation des ordinateurs intégreront la gestion de l'environnement graphique. Nous commettrons donc un abus de langage en assimilant les deux types de programmes.

A quel(s) système(s) d'exploitation avez-vous à faire ?

.....

A la mise sous tension de l'ordinateur, disposez-vous directement d'un environnement graphique ? Si non, comment y arrivez-vous ?

.....

B. Le rôle du système d'exploitation

Il faut bien reconnaître qu'une bonne partie du travail du système d'exploitation échappe complètement à l'utilisateur. Et fort heureusement, car il s'agit d'un travail extrêmement fastidieux, résultant d'algorithmes (marches à suivre) très complexes, mais qui trouve une large justification dans la vitesse de travail élevée du processeur.

En revanche, ce qui intéresse l'utilisateur au premier chef, c'est de pouvoir dialoguer simplement avec le système d'exploitation dans des circonstances précises que nous allons décrire.

Le chargement de programmes dans la mémoire centrale

C'est l'interaction de base de l'utilisateur avec le système d'exploitation. Dans les systèmes qui n'intègrent pas d'environnement graphique, le chargement d'un programme en mémoire se fait généralement en précisant à l'OS un nom de programme qu'il reconnaît (par exemple, en *MS-DOS*, un nom de huit caractères au maximum). Plus fréquemment aujourd'hui, la même opération se réalise en cliquant sur une icône ou dans un menu.

Citez différentes manières de charger un programme en mémoire centrale avec votre système d'exploitation

.....
.....
.....
.....

Le chargement de données dans la mémoire centrale

Dans les systèmes sans environnement graphique, c'est par l'intermédiaire du programme que le chargement peut avoir lieu. Par exemple, lorsqu'un programme de traitement de texte est chargé dans la mémoire centrale, vous pouvez donner l'ordre à ce programme de traitement de texte de charger un texte se trouvant sur un support externe. Le programme de traitement de texte s'adresse alors au système d'exploitation qui effectue le chargement.

Dans les systèmes avec environnement graphique, la même technique peut être employée. Mais des méthodes plus rapides permettent de charger en une fois, les données et le programme qui les traite.

Qu'il s'agisse de programmes ou de fichiers de données, **charger en mémoire signifie effectuer une copie en mémoire centrale d'informations se trouvant en mémoire externe.**

Citez différentes manières de charger un fichier de données en mémoire centrale avec votre système d'exploitation

.....
.....
.....
.....

L'enregistrement des données en mémoire externe

Il n'y a aucun sens à enregistrer les programmes, puisque ceux-ci proviennent par copie de la mémoire externe. En revanche, un fichier de données qui vient d'être créé ou qui vient d'être modifié doit être enregistré en mémoire externe au risque sinon de rendre inutile le travail effectué.

L'enregistrement se fait par l'intermédiaire du programme qui a créé ou modifié le fichier de données. C'est lui qui s'adresse au système d'exploitation en lui fournissant deux renseignements importants : le nom du fichier de données et l'endroit logique de son enregistrement.

A ce propos, il faut préciser que l'OS impose des contraintes d'organisation. Le nom doit respecter certaines limitations, et le support externe d'enregistrement est assez classiquement organisé en arborescence de **répertoires** (pour ce qui est du vocabulaire utilisé plus anciennement dans le monde PC et compatibles) ou de **dossiers** (pour ce qui est du vocabulaire utilisé plus récemment dans le monde PC et compatibles et depuis toujours dans le monde Macintosh). Quant aux fichiers de données, on les appellera respectivement **fichiers** ou **documents**. A noter cependant que quelques ambiguïtés subsistent parfois du fait de l'importance historique du mot *fichier* dans le vocabulaire des technologies de l'information. Ce mot est d'ailleurs souvent le premier qui apparaît dans une barre de menus et il donne accès à tous les traitements dont nous sommes en train de parler.

Votre système d'exploitation utilise plutôt le vocabulaire suivant :

.....
.....

Il vous impose les contraintes suivantes pour les noms utilisés :

.....
.....

Il ne peut exister deux fichiers portant le même nom dans un même dossier ou répertoire. En conséquence, tout enregistrement d'un fichier de données portant déjà un nom a pour effet de remplacer le contenu du fichier de données sur le support externe par une copie du fichier de données présent en mémoire centrale. Si cette destruction n'est pas souhaitable (parce que vous souhaitez garder aussi l'ancienne version), il vous suffit d'enregistrer les données sous un autre nom.

Décrivez comment votre système d'exploitation vous permet d'enregistrer un fichier de données. Expliquez comment procéder pour gérer les situations qui viennent d'être décrites (conservation ou non d'une version antérieure du fichier de données)

.....
.....
.....
.....
.....

Décharger un fichier de données de la mémoire centrale

C'est l'opération qui consiste à enlever de la mémoire centrale un fichier de données qui n'est plus nécessaire. Elle se fait par l'intermédiaire du programme qui s'adresse une fois de plus au système d'exploitation. Elle peut s'effectuer indépendamment du déchargement du programme mais peut aussi s'effectuer conjointement. Elle n'a de sens que si les modifications des données ont été enregistrées. Toutefois, si cet enregistrement n'a pas eu lieu, le système d'exploitation se chargera généralement de vous le rappeler.

Expliquez comment faire pour décharger un fichier de données de la mémoire centrale. Décrivez les variantes.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Décharger un programme de la mémoire centrale

Cette opération est souhaitable lorsqu'un programme n'est plus nécessaire dans la suite d'une session de travail. Le maintien en mémoire centrale d'un programme inutile ne fait qu'encombrer celle-ci. C'est donc une bonne habitude d'effectuer ce déchargement dès que possible.

Expliquez différentes manières de décharger un programme de la mémoire centrale

.....

.....

.....

.....

L'organisation des informations sur les différents supports

Par informations, il faut entendre : les données et les programmes. Vous savez que le système d'exploitation considère les uns comme les autres comme des **fichiers**, appellation issue d'une époque révolue où l'unité d'entrée par excellence était le lecteur de cartes (fiches) perforées.

C'est donc naturellement à vous, utilisateurs, d'organiser ces informations de manière logique. Quant à l'organisation physique, elle vous importe peu.

L'organisation logique est, rappelons-le, une arborescence de répertoires, ou de dossiers (appellation que nous utiliserons désormais systématiquement). Chaque dossier peut contenir d'autres dossiers et bien entendu des fichiers (documents et programmes). Il convient de créer

des dossiers spécifiques pour les documents qu'il vaut mieux séparer des programmes. Il n'est pas non plus souhaitable de placer tous les documents créés dans le même dossier.

Quelques opérations fondamentales à découvrir sont donc :

- comment créer un nouveau dossier ?
- comment supprimer un ou plusieurs documents?
- comment déplacer un ou plusieurs documents ?
- comment copier un ou plusieurs documents ?
- comment changer le nom d'un document ?
- comment changer le nom d'un dossier ?
- comment supprimer un ou plusieurs dossiers?

Décrivez comment ces opérations sont possibles avec votre système d'exploitation

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

8. Les échanges d'informations entre différents documents

Une autre mission du système d'exploitation est d'assurer des possibilités de transfert d'informations d'un document dans un autre. Différentes techniques existent. Nous nous contenterons d'en décrire les principes dans le cadre des produits qui ont été examinés dans ces notes, à savoir, le traitement de texte et le dessin et nous nous limiterons aux systèmes d'exploitation qui gèrent le déroulement simultané de plusieurs tâches.

Nous avons déjà découvert qu'il était possible d'envoyer une sélection d'informations dans ce que nous avons appelé le *presse-papiers*. Nous en avons l'utilité lorsque nous souhaitions déplacer (*couper-coller*) ou reproduire (*copier-coller*) cette sélection à un autre endroit du même document.

A. Documents gérés par un même programme

Cette technique est toujours valable si vous souhaitez effectuer ces opérations d'un document créé par un programme dans un autre créé par le même programme. Si l'on veut bien donner le nom de **tâche** à l'association d'un programme avec un document, il suffit de lancer les deux tâches (programme X avec document Y et programme X avec document Z), d'effectuer le *couper* ou le *copier* dans le premier document et le *coller* dans le second. C'est particulièrement pratique si vous voulez fusionner des informations provenant de plusieurs textes par exemple.

B. Documents gérés par des programmes différents

Imaginez que vous souhaitiez intégrer un dessin par points dans un texte. Il s'agit cette fois de deux programmes différents. La technique du *couper-copier-coller* mérite toujours d'être tentée. Le résultat n'est pas garanti. Il dépend de beaucoup de choses et principalement de l'évolution des techniques d'échange en fonction des versions des systèmes d'exploitation.

Dans le meilleur des cas, le document qui "reçoit" les informations a été créé par un programme qui est une **application cliente** de l'autre programme, celui qui a créé le document qui "offre" les informations aussi appelé **application serveur**. Dans ce cas, tout se passera bien et même, si par la suite vous souhaitez modifier les informations insérées, c'est l'application serveur qui reprendra la main pour vous permettre de faire toutes les modifications que vous souhaitez.

Un programme de traitement de texte est souvent une application cliente pour de nombreux autres programmes alors qu'un programme de dessin est plutôt une application serveur par vocation. Il existe aussi des programmes qui sont serveurs pour certains clients et clients pour certains serveurs.

Une autre possibilité existe, c'est que le programme qui gère le document "récepteur" ne soit pas une application cliente du programme qui gère le document "donneur" mais qu'il accepte tout de même d'adopter les informations offertes en leur donnant un format personnel. Ce format ne permettra souvent que peu de modifications par la suite, mais parfois, on s'en contentera.

Citez quelques programmes dont votre traitement de texte est une application cliente et les moyens d'y faire appel:

.....

.....

.....

.....

.....

.....

.....

9. Quelques références bibliographiques

1. F. SASS, E. VANDEPUT *Informatique utile et raisonnée Tome 1 Notions et méthodes fondamentales* Van In Lier 1993
2. F. SASS, E. VANDEPUT *FP 8.1: Dessin assisté par ordinateur: les actions de base* CeFIS - ICAFOC - FPE 1994
3. J.-P. ANCIAUX *Les règles d'or de la communication écrite, la mise en page* Les Editions d'organisation Paris
4. F. RICHAUDEAU *Manuel de Typographie et de Mise en Page* Retz Paris 1989
5. F. SASS *Mettre en page* Licap SeGEC Bruxelles 1996
6. E. VANDEPUT *FP 4.4: Traitements de texte: idées d'exercices et de problèmes* CeFIS - ICAFOC - FPE 1995