

ŒIL DE LYNX



ÉDITORIAL

Amis Lynxistes bonjour !

Par cette canicule estivale, la météo incite plus au farniente prolongé qu'au "remue méninges" de la programmation.

Pourtant nous ne doutons pas que malgré la chaleur accablante et l'appel de la mer, vous ne vous ruez sur ce numéro 2 d'Œil de Lynx. Tenant compte de vos suggestions – un grand merci à tous ceux qui nous ont écrit – nous avons inclus plus de programmes. Vous trouverez donc GESFICH, gestionnaire de fichier puissant pour 48 et 96 Ko, GRID TRAP jeu d'arcade en Baset, une boîte à outils sur quatre pages (avec plan de demos graphiques et deux vrais programmes de jeux).

Mais surtout, vous trouverez une nouvelle rubrique d'initiation pour vous permettre de mieux connaître votre machine. Dans ce numéro et les prochains, nous aborderons l'étude du langage machine qui devrait intéresser la majorité d'entre vous.

N'hésitez pas à nous écrire (notre adresse, Segme'n, 140 Bd Haussmann 75008 Paris) pour nous faire part de vos remarques, de vos critiques et – pourquoi pas – de vos félicitations... Et n'oubliez pas que la boîte à outils attend vos trucs, astuces et petites annonces.

Nous espérons que ce numéro vous aidera à passer de bonnes vacances et nous vous donnons rendez-vous en Septembre pour un "Spécial Sien".

N°2 AOÛT 84

30 F

news

L'actualité est plutôt calme en cette période estivale. Quelques produits ont cependant retenu notre attention. Deux classiques des jeux d'arcades Twinkle et Diggeman, tout d'abord, qui viennent compléter une bibliothèque déjà bien garnie. Wordprocessor et Card Index, deux logiciels d'application dédiés, une fois n'est pas coutume aux modèles 48 et 96 Ko : le traitement de texte et la gestion de fichier que les lynxistes attendaient. Mais le produit le plus marquant a été présenté à Micro Expo, en mai dernier, et il ne s'agit pas d'un ordinateur, ni de périphérique, ni de logiciel. Pourtant, "Joker" (c'est son nom) réjouira bien des utilisateurs en réduisant les problèmes d'encombrement. Enfin n'oubliez pas le numéro de téléphone de Lauréat-Assistance, Service de la Société Objectif Micro permettant aux habitants de la région parisienne d'obtenir tous les renseignements sur le dernier modèle de la gamme Lynx. Espérons que les provinciaux se rallieront à cet exemple !

Rendez-vous en septembre au Sicob !

Stand n° 76 — Niveau Boutique

WORDPROCESSOR (96 Ko)

Enfin un traitement de texte simple sur Lynx. Si les - heureux - utilisateurs du Lauréat n'ont que l'embarras du choix, il n'en est certes pas de même pour les autres modèles!

Le programme Wordprocessor est capable de créer des documents de plusieurs pages selon vos propres spécifications. A l'impression, les lignes peuvent comporter jusqu'à 126 caractères, et la longueur de chaque page peut être déterminée. Les documents peuvent être édités à l'écran, imprimés et stockés sur une cassette magnétique. Les textes sur cassettes peuvent être simplement incorporés dans le document en cours d'édition, ce qui permet de disposer d'un outil efficace et puissant.

L'utilisation de ce logiciel ne nécessite pas de connaissances particulières, à part l'usage du clavier, de l'imprimante et du magnétophone à cassettes. L'objectif de ses concepteurs était de réaliser un éditeur de texte performant, n'occupant que peu de place en mémoire interne (environ 3 Ko) et doté de toutes les fonctions classiques de ce type de programme. Objectif atteint !



TWINKLE ET DIGGERMAN (48 et 96 Ko)

Deux nouveaux jeux d'arcades sur Lynx qui réjouiront sans doute les nombreux amateurs.

Twinkle est une nouvelle version de la bataille de l'espace, basée sur le scénario des "Space Invaders". Mais la rapidité de l'action, la qualité des graphismes et le soin des détails en font sans conteste l'un des futurs best-sellers sur Lynx.



Diggeman est aussi une adaptation d'un classique du genre : poursuivi dans des échafaudages par des animaux aussi bizarres que mal intentionnés, vous ne pouvez leur échapper qu'en creusant des trous sous leurs pas... Parviendrez-vous à survivre longtemps ?

Écrits tous deux en langage machine par les programmeurs de "Geni Software", Twinkle et Diggeman ne doivent pas manquer à votre ludothèque !

CARD INDEX (96 Ko)

Le programme "Card Index" est un programme de gestion de fichiers destiné aux modèles Lynx 48 et 96 Ko. Chaque fiche (ou "card") peut contenir 10 rubriques différentes. Ainsi, un carnet de rendez-vous comprendra les enregistrements suivants : DATE, HEURE, LIEU et NOM DU CONTACT.

Les intitulés des rubriques sont donnés à la création du fichier, et chaque fiche doit être remplie dans le même ordre (un espace est nécessaire pour passer d'une rubrique vide à la rubrique suivante).

Ce programme, sauvegardant les enregistrements sur cassettes, à une capacité approximative de 23000 caractères, ce qui correspond en moyenne à 150 fiches.

news

"JOKER"

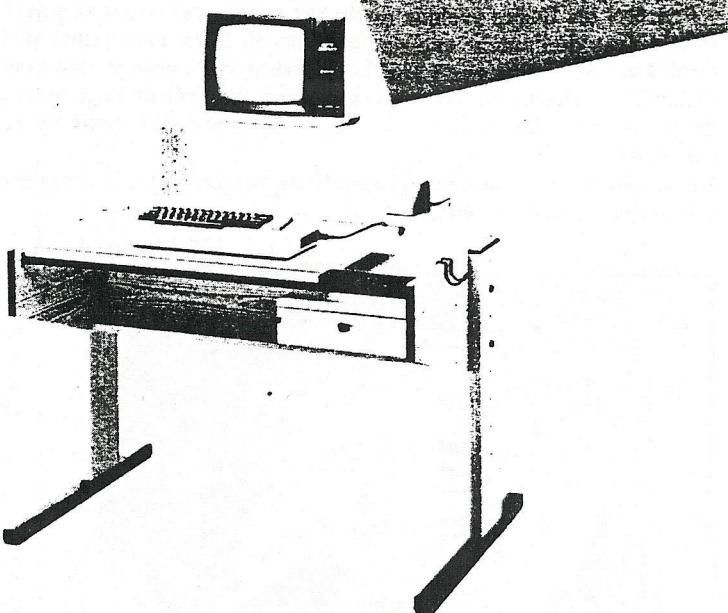
1.590 F ttc

NOUVEAU POSTE DE TRAVAIL POUR MICRO FAMILIAL

Tous les possesseurs de Lynx le savent bien, il est très difficile de trouver un emplacement agréable et fonctionnel pour un micro-ordinateur dans un appartement.

Pour répondre à ce besoin, la société Segimex vient d'inscrire à son catalogue un bureau conçu à cet usage : "Joker". Évolutif (hauteur réglable de 44 à 76 cm), il offre un large plan de travail avec inclinaison réglable, et deux tablettes arrières pour placer moniteur et périphériques.

Livré en kit complet (avec la colle et le tournevis nécessaire à son montage) au prix de 1590 F ttc, le bureau Joker existe en deux versions : laminé blanc ou couleur pin naturel. Précisons pour finir que ces deux revêtements sont lavables, et insensibles aux colles, encre et feutre...



UN LYNX CHEZ LES ÉDITEURS !

Lynxistes, réjouissez-vous ! le premier ouvrage consacré à votre micro-ordinateur favori vient de paraître chez l'un des plus grands éditeurs français dédiés à la micro-informatique Eyrolles.

Ce manuel est dû à la plume de deux des auteurs les plus connus de la Maison Du Bd St Germain, Bruno Van Rib et Roger Politis. Après le Multitech MPF II et le New Brain, c'est aujourd'hui le Lynx qui fait l'objet de leur dernier titre.

Consacré à la programmation en langage Basic à l'usage des débutants, ce livre reprend les différentes instructions spécifiques du Lynx (graphismes, redéfinition de caractères, son et synthèse vocale, procédures...).

Complément parfait du manuel de programmation du Lynx, cet ouvrage est à conseiller à tous les lynxistes débutants ou confirmés. Un coup de chapeau aux auteurs !

SEGIMEX PRÉSENT À MICRO EXPO

Un stand consacré au(x) Lynx(s) par la société Ségimex présentait aux visiteurs de Micro Expo, au Palais des Congrès, en mai dernier, les différents modèles (48, 96 Ko et bien entendu le Lauréat) qui ont suscité l'intérêt des visiteurs et des professionnels du Salon.

Une surprise, dont nous aurons sans doute l'occasion de reparler, fut même visible en avant-première !

Une exposition sans surprise, mais un succès pour la micro informatique familiale.

SERVICE ASSISTANCE LAURÉAT

Pour faire mieux connaître le dernier né de la famille Lynx, le Lauréat 128 Ko que nous vous avons présenté dans le numéro précédent, la société Objectif Micro a créé un service téléphonique, afin de répondre à toutes vos questions.

Il suffit en effet de composer le 265.62.89 et de demander le service assistance Lauréat pour obtenir tous les renseignements désirés.

Rappelons que cette société présente aussi la gamme complète des matériels Lynx dans ses deux centres de démonstration :

- 101 Bd Haussmann 75008 Paris (Métro St Augustin ou Miromesnil)
- 151 Bd Jean Jaurès 92110 Clichy



PROGRAMME DU MOIS

Programmer un jeu d'action rapide sur micro-ordinateur requiert souvent la connaissance du langage machine. Pourtant les Basics évolués comme celui du Lynx permettent des réalisations intéressantes. Ainsi le programme Grid Trap de l'auteur anglais M.S. Fowkes vous promet des heures de divertissement. Inspiré des classiques "Chenilles" chères aux Lynxistes anglo-saxons, Grid Trap vous place dans la peau d'un voleur de diamants. Le but du jeu est de collecter le maximum de diamants en évitant les obstacles et sa propre trace. De plus le temps imparti est limité.

Le succès dépend donc de la rapidité de vos réflexes. A chaque nouveau succès la partie reprend, plus rude et plus acharnée, ... jusqu'à votre défaite.



La structure du programme est relativement simple, puisqu'elle ne fait appel qu'à quelques procédures : initialisation du jeu (PROC INIT), disposition des diamants et des obstacles (PROC SETUP), redéfinition des caractères graphiques (PROC GRAPHISMS) et jeu proprement dit (PROC GAME).

Le mode d'emploi du jeu, contenu dans la procédure INSTRUCTIONS (lignes 1340 et suivantes), s'affiche automatiquement en début de partie.

Il est possible de modifier le niveau de difficulté du jeu en changeant la valeur maximale de la boucle FOR...NEXT de la ligne 380 qui fixe le nombre d'obstacles et de diamants de chaque tableau.

```
10 REM ** GRIDTRAP M.S FOWKES 1984 **
20 PROTECT 0
30 LET S=0, G=1, J=300
40 PROC GRAPHICS
50 PROC INSTRUCTIONS
60 PROC GRID
70 PROC INIT
80 PROC SETUP
90 PROC GAME
100 CLS
101 END
110 DEFPROC GRID
120 CLS
130 INK BLUE
140 FOR A=10 TO 190 STEP 10
150 MOVE 0,A
160 PLOT 3,250,0
170 NEXT A
180 FOR A=10 TO 240 STEP 10
190 MOVE A,0
200 PLOT 3,0,200
210 NEXT A
220 PROTECT BLUE
230 ENDPROC
240 DEFPROC INIT
250 INK RED
260 MOVE 0,0
270 PLOT 3,250,0
280 PLOT 3,0,200
290 PLOT 3,-250,0
300 PLOT 3,0,-200
310 PRINT @ 50,103;CHR$(24);"GRIDTRAP";
CHR$(25);
```

```
320 INK GREEN
330 PRINT @ 9,213;"GRILLE";@ 97,213;"SCORE";
340 INK YELLOW
350 PRINT @ 12,225;G;"      ";@ 98,225;
S;"      ";
360 ENDPROC
370 DEFPROC SETUP
380 FOR A=1 TO 20+G*4
390 LET I=10*(RAND(15)+2),C=10*(RAND(2
0)+2)
400 PRINT @ C/2-1,1-5;CHR$(1);CHR$(2)
;CHR$(129);
410 BEEP 70,80,60
420 NEXT A
430 LET C=10*(RAND(23)+1),L=10*(RAND(18)
+1)
440 PRINT @ C/2-1,L-5;CHR$(1);CHR$(6);C
HR$(130);
450 MOVE C,L
460 INK 4
470 FOR A=1 TO 20+G*4
480 LET C=10*(RAND(24)+1),I=10*(RAND(1
9)+1)
490 LET a=f*32+c DIV 8
500 LET b=2**7-c MOD 8
510 CALL &0070,&C000+a
520 LET d=SGN(HL BNAND b)
530 CALL &0069,&C000+a
540 LET d=d+SGN(HL BNAND b)
550 IF d>0 THEN GOTO 480
560 PRINT @ c/2-1,1-5;CHR$(128);
```

"GRIDTRAP"

par M. S. FOWKES

```
570 BEEP 100,90,60
580 NEXT A
590 ENDPROC
600 DEFPROC GAME
610 FOR A=100 TO 200 STEP 10
620 BEEP A,50,60
630 NEXT A
640 LET N=0
650 INK RED
660 PRINT @ 97,225;CHR$(1);CHR$(6);S;""
";CHR$(1);CHR$(2);
670 BEEP 110,120,45
680 LET T=1
690 REPEAT
700 LET T=T+1
705 LET M=INP(&0080),0=INP(&0980)
710 UNTIL T>J OR (((M=239 OR M=223) OR (0=251 OR 0=223)) AND (NOT(M>255 AND 0<>255)))
720 IF T>J THEN GOTO 760
730 BEEP 120,90,50
740 LET S=S-20*G
750 GOTO 660
760 IF M=239 THEN LET L=L-10
761 ELSE IF M=223 THEN LET L=L+10
762 IF 0=251 THEN LET C=C-10
763 ELSE IF 0=223 THEN LET C=C+10
770 LET Q=2**((7-C MOD 8))
780 CALL &0070,&C000+L*32+C DIV 8
790 LET Z=SGN(HL BNAND Q)
800 IF Z=0 THEN GOTO 810
805 LET S=S+10,N=N+1
807 GOTO 840
810 CALL &0069,&C000+L*32+C DIV 8
820 LET Y=SGN(HL BNAND Q)
830 IF Y=1 THEN PROC ENDGAMEF
840 LET S=S+1
850 DRAW C,L
860 IF N=20+G*4 AND S>0 THEN PROC GRIDCLEAR
870 GOTO 660
880 ENDPROC
890 DEFPROC GRAPHICS
900 CODE 00 00 0C 1E 3F 3F 1E 0C 00 00
00 00 3F 0C 3F 3F 0C 3F 00 00 00 FF FF F
F FF FF FF FF 00
910 DPOKE GRAPHIC,LCTN(900)
920 ENDPROC
930 DEFPROC GRIDCLEAR
940 DRAW C,L
950 FOR W=400 TO 180 STEP -15
960 BEEP W,40,60
970 NEXT W
980 FOR F=0 TO 31
990 OUT &0086,13
1000 OUT &0087,F
1010 BEEP 100-F,100,60
1020 NEXT F
1030 OUT &0086,13
1040 OUT &0087,0
1050 LET G=G+1
1060 IF J>100 THEN LET J=J-50
1070 CLS
1080 PROC INIT
1090 PROC SETUP
1100 PROC GAME
1110 ENDPROC
1120 DEFPROC ENDGAMEF
```

```
1130 FOR K=0 TO 20
1140 LET Y=RAND(30)+1
1150 OUT &0086,13
1160 OUT &0087,Y
1170 BEEP 120-Y,120,60
1180 NEXT K
1190 OUT &0086,13
1200 OUT &0087,0 → PROTECT φ
1205 EXT VRESET → CLS
1210 PRINT @ 50,35;CHR$(24);CHR$(1);CHR$(6);"GRIDTRAP";
1220 PRINT @ 45,45;CHR$(1);CHR$(2);CHR$(2);CHR$(0);"JEU TERMINE";CHR$(25);
1230 PRINT @ 27,207;CHR$(1);CHR$(6);"VOUS AVEZ TRAITE ";G-1;" GRILLE";CHR$(83)*
NOT(G-1=1));
1240 PRINT @ 27,217;"ET OBTENU ";
S;" POINTS";
1250 PRINT @ 27,238;"'O' POUR REJOUER-'N POUR FIN";
1260 LET I$=GET$
1270 IF I$="O" OR I$="o" THEN GOTO 1300
1280 IF NOT(I$="N" OR I$="n") THEN GOTO
1270
1289 CLS
1290 END
1300 LET S=0,G=1,J=300
1310 CLS
1315 PROC GRAPHICS
1320 GOTO 60
1330 ENDPROC
1340 DEFPROC INSTRUCTIONS
1350 PAPER 0
1360 CLS
1370 PRINT @ 50,0;CHR$(1);CHR$(2);CHR$(24);"GRIDTRAP";CHR$(25);
1380 PRINT @ 44,26;CHR$(1);CHR$(6);"INSTRUCTIONS";CHR$(1);CHR$(4);CHR$(10)
1390 PRINT "Votre BASE est representee par un carre ";CHR$(1);CHR$(6);"JAUNE";CHR$(1);CHR$(4);" au debut de la grille. Pour vous en echapper, vous devez :"
1400 PRINT "Collecter les DIAMANTS VERTS en evitant les OBSTACLES ROUGES."
1410 PRINT ;CHR$(1);CHR$(2);"SI VOUS REPASSEZ SUR VOTRE TRACE - OU HEURTEZ UN OBSTACLE -, VOUS NE POURREZ VOUS ECHAPPER DE GRIDTRAP"
1415 PRINT
1420 PRINT ;CHR$(1);CHR$(4);"POUR VOUS D'EPLACER, UTILISEZ LES TOUCHES DE COMMANDE DU CURSEUR"
1430 PRINT "LORSQUE TOUS LES DIAMANTS D'UNE GRILLE SONT COLLECTES, VOUS ALLEZ A LA SUIVANTE, PLUS FOURNIE EN DIAMANTS, MAIS AUSSI EN OBSTACLES"
1440 PRINT "VOTRE TEMPS EST LIMITE, SI VOUS ATTENDEZ VOUS PERDEZ DES POINTS. VOUS NE POURREZ CHANGER DE GRILLE AVEC UN SCORE NEGATIF."
1450 PRINT @ 16,241;CHR$(7);"PRESSEZ UNE TOUCHE POUR COMMENCER";
1470 LET I$=GET$
1480 ENDPROC
121250 PRINT @ 18,238;"'O' POUR REJOUER-'N POUR FIN";
131220 PRINT @ 45,45;CHR$(1);CHR$(2);CHR$(2);CHR$(0);"JEU TERMINE";CHR$(25);
```



SORTIE SON SUR TÉLÉVISEUR

Le haut-parleur du Lynx ne dispose pas d'un volume très puissant et ne permet aucun réglage. Pour remédier à ce petit inconvénient, il est possible de connecter la sortie magnétophone du Lynx sur l'entrée correspondante de la prise Péritelévision.

Pour réaliser ce petit branchement, il faut vous munir de :

- une longueur suffisante de fil blindé
- deux contacts pour prise Péritelévision
- une prise DIN 3 ou 5 broches (3 suffisent)
- un fer à souder avec un peu de tymol
- ... et un quart d'heure de patience !

Pour réaliser le montage désiré, il suffit donc de souder la tresse du fil blindé à la pine 6 de la Péritel (entrée audio monophonique) et l'âme à la pine 4 (masse commune audio).

Pour permettre la connexion à la sortie magnétophone du Lynx, il suffit de souder la tresse à la borne 2 (masse) et l'âme à la borne 1 (sortie gauche). Si votre prise DIN comporte cinq broches, les bornes 3, 4, 5 seront donc inutilisées.

Alain DEHU

CONTACTS

Habitant la région parisienne (Hauts de Seine), cherche contact avec utilisateur de Lynx 48 ou 96 Ko. Possède de nombreux logiciels, jeux et utilitaires divers, pour échange.

Jean-Marc Drouard
Tél. : (1) 782.94.00

PROGRAMMES SUR CASSETTES

Sur une cassette peuvent être sauvegardés deux types de programmes, écrits soit en Basic (sauvés par la commande SAVE), soit écrits en langage machine (sauvés par la commande D du moniteur). Ces programmes sont enregistrés sur la bande magnétique selon des formats différents (voir fig. 1).

Un certain nombre de sous-programmes de la mémoire morte permettent d'obtenir bien des renseignements :

0CF2: active le moteur du magnétophone

0CFB: désactive le moteur

0B65: lit la cassette jusqu'à ce que le code A5 soit trouvé (synchronisation)

0B85: lecture d'un octet dans le registre A

62A3: lecture d'un bit

DC27: active le moteur et lit la cassette jusqu'à ce que le nom du programme soit le même que celui précisé dans les registres DE

3FC4: désactive le moteur et affiche ????

3F6A: lit un programme machine dont le nom est en DE, puis exécute le programme si une adresse de lancement a été précisée lors de la sauvegarde par la commande D du moniteur

0B93: écrit n fois 256 octets nuls et du code A5

0BB4: effectue la même opération que 0B93 puis écrit le nom du programme précisé dans DE

0BB7: écrit le nom du programme précisé dans DE

62A6: écriture de l'octet A sur la cassette.

De plus, quatre adresses permettent d'obtenir tous les renseignements con-

cernant le mode d'enregistrement (commande TAPE):

	629C	629D	629E	629F
TAPE0	600 B &50	1	4	3
TAPE1	900 B &38	1	1	4
TAPE2	1200 B &2C	2	4	5
TAPE3	1500 B &23	2	2	6
TAPE4	1800 B &1B	2	1	7
TAPE5	2100 B &14	3	2	8

En 629F, on trouve la valeur N du nombre de fois 256 octets nuls réservés en tête d'enregistrement.

DES TOUCHES DE FONCTION SUR MESURE

Le Lynx possède 26 fonctions Basic préprogrammées (les lettres de A à Z pressées simultanément avec "ESC"), mais la disposition de ces quelques (!) touches ne convient pas forcément à tout le monde. Il est possible de les modifier, mais cela exige la connaissance de tous les codes de fonctions.

Le petit programme suivant permet de redéfinir simplement ces affectations :

```
10 code 3E 7B CF CD E2 10 11 00
      60 D7 FE OD C8 13 CD FA
      3D D6 41 38 1B FE 1A 30
      17 21 3A 62 85 6F E5 2A
      19 62 46 C5 CD 49 19 D1
      E1 38 05 7A 90 77 18 D2
      21 C8 3B CE 39 35 18 F6
```

20 CALL LCTN(10)

La liste des codes de la ligne 10 correspondant au programme assembleur suivant :

```
LD A,3C /RST 08/CALL 10E2
RST 10/CP OD/RET Z/INC DE/CALL
      3DFA
SUB 41/JR C,1B/CP 1A/JR NC,17
LD HL,623A/ADD L/LD L,A/PUSH
      HL/LD HL,)6219)
LD B,(HL)/PUSH BC/CALL 1949/POP
      DE
POP HL/JR C,05/LD A,D/SUB B/LD
      (HL),A
JR -2E/LD HL,3BC8/CALL 3539/JR -A
```

L'utilisation de ce programme est extrêmement simple puisqu'il suffit après un RUN d'entrer le symbole "flèche à droite", la touche suivie de sa nouvelle affectation :

```
>RUN
      A APPEND (return)
      W WINDOW (return)
      M MAMAN (return)
      (return)
```

REMARQUE : Pour sauver la "préprogrammation" ainsi obtenue, il faut entrer un petit programme machine à partir du moniteur :

```
> MON
OD44 3B64 630E 24BE
4D2A 0091 F5BF EF20 630D FFFF
9FF8 24BD
D 623A 6253 0 "preprog"
J
>
```

Le chargement s'obtient par MLOAD "PREPROG".

Notre dossier "Spécial Graphisme" a suscité de vives réactions parmi les Lynxistes et nous a valu un volumineux courrier. Des lettres de félicitations, dont notre modestie a évidemment souffert, mais qui prouve votre intérêt pour ce type d'applications.

Des envois de programmes ensuite, parmi lesquels nous publions nos préférés : "Planète", "Mappemonde" et enfin "Forêt".

Quelques reproches enfin. Certains d'entre vous trouvent le graphisme "bien beau" mais n'apprécient pas le côté gratuit des applications purement esthétiques

Ces remarques sont certes justifiées mais la connaissance des instructions graphiques nous semblent nécessaires à la programmation des jeux. Pour leur plaisir, et pour le notre voici un jeu, "Etoiles" qui réconciliera tout le monde...

```
100 TEXT
110 PROTECT YELLOW
120 INK BLUE
130 REM *****ETOILE*****
140 FOR V=1 TO 200 STEP 5
150 DOT RAND(V),RAND(256)
160 NEXT V
170 WINDOW 0,127,0,255
180 PROC INITIALE
190 REPEAT
195 LET C=0
200 POKE L+6,z
210 CALL L
220 REPEAT
230 PROC SAUT
240 UNTIL Z=256
250 LET Z=0,z=(z+1) MOD 8,Y=Y MOD 256
260 UNTIL FALSE
270 DEFFPROC SAUT
280 OUT &0087,z
290 PROC ASTEROIDE
300 IF INF(&0980)<>255 THEN PROC ACTION
310 PROTECT CYAN
320 PRINT @ X,Y;A$;
330 LET Y=Y+4,M=(M+1) MOD 64,Z=Z+32
340 ENDPROC
350 DEFFPROC ACTION
360 IF INF(&0980)=251 THEN LET X=X-1
370 IF INF(&0980)=223 THEN LET X=X+1
380 IF X=2 THEN LET X=3
390 ELSE IF X=118 THEN LET X=117
400 REM *****PERTE D'ENERGIE*****
410 BEEP E,1000 DIV E,63
420 LET E=E-1
425 IF E<=0 THEN PROC FINDEJEU
430 IF E<=0 THEN PROC FINDEJEU
440 ENDPROC
450 DEFFPROC ASTEROIDE
460 PROTECT BLUE
470 PRINT @ P(M),M*4;" ";
480 LET P(M)=RAND(126),C(M)=138+SGN(RAND
(17)-8)*2
490 IF C(M)<>138 THEN INK GREEN
500 PROTECT WHITE-INK
510 PRINT @ P(M),M*4;CHR$(C(M));
520 REM ****EXAMEN DE LA COLLISION****
530 IF P((M-54) MOD 64)>=X AND P((M-54)
MOD 64)<=X+4 THEN PROC COLLISION
540 ENDPROC
550 DEFFPROC COLLISION
560 IF C((M-54) MOD 64)=138 THEN LET C=
20
570 FOR V=4 TO 1 STEP -0.5
580 FOR W=1+C TO 31
590 BEEP W*V,4,31*V+1-W
600 NEXT W
610 NEXT V
620 IF C=20 THEN LET E=E+20
625 ELSE LET E=E-50
630 IF E<0 THEN PROC FIDEJEU
640 ENDPROC
650 DEFFPROC INITIALE
660 DPOKE GRAPHIC,LCTN(830)
670 DIM A$(9),P(63),C(63)
680 LET Y=20,M=61,Z=0,z=0,L=LCTN(820),E=
150
690 INK RED
700 PROTECT CYAN
710 REM *****BOUCLE D'INITIALISATION*****
720 FOR X=1 TO 9
730 READ V
740 LET A$=A$+CHR$(V)
750 NEXT X
760 DATA 1,2,128,130,22,22,10,132,134
770 FOR X=0 TO 63
780 LET P(X)=0
790 NEXT X
800 ENDPROC
810 REM ****CODE ****
820 CODE 01 86 0C ED 41 3E 04 D3 87 04
ED 41 3E 00 D3 87 C9
830 CODE 00 00 00 00 00 00 00 07 07 03
00
840 CODE 00 00 00 00 00 20 20 3C 38 30
00
850 CODE 01 01 09 09 0F 08 01 01 01 01
00 00 00
860 CODE 30 30 30 28 28 28 30 20 00 00
00 00 00
870 CODE 2C 18 34 18 00 00 00 00 00 00 00
00 00 00
880 CODE 24 18 18 24 00 00 00 00 00 00 00
00 00 00
890 CODE 14 20 0C 20 00 00 00 00 00 00 00
00 00 00
895 DEFFPROC FINDEJEU
900 OUT &0086,12
910 OUT &0087,0
920 OUT &0086,13
930 OUT &0087,0
940 WINDOW 3,123,2,245
```

```

90 REM "FOREST"      96 K
100 PAPER BLUE
110 CLS
120 FOR A=1 TO 12
130 READ C,X,Y,R
140 INK C
150 EXT CIRCLE 1,X,(Y-40),R
160 NEXT A
170 DATA 2,127,350,300
180 DATA 6,127,350,290
190 DATA 3,127,350,280
200 DATA 1,127,350,270
210 DATA 4,127,350,260
220 DATA 5,127,350,250
230 DATA 1,127,350,240
240 DATA 7,170,110,12
250 DATA 7,185,102,8
260 DATA 7,205,100,15
270 DATA 7,181,120,10
280 DATA 7,195,115,13
290 INK YELLOW
300 EXT TRIANGLE 127,175,0,251,255,251
310 INK 4
320 EXT TRIANGLE 255,251,127,175,255,150
330 EXT TRIANGLE 0,150,127,175,0,251
340 FOR T=1 TO 21
350 READ X,Y,A,B,C,D
360 EXT TRIANGLE X,Y,A,B,C,D
370 NEXT T

```

```

380 INK RED
390 FOR B=0 TO 200 STEP 2
400 LET Q=RAND(76)
410 LET W=127+SGN(RND-0.5)*RAND(Q*32 D
IV 19)
420 DOT W,Q+175
430 NEXT B
440 DATA 0,170,13,115,26,170
450 DATA 24,170,34,117,44,170
460 DATA 37,170,46,123,55,170
470 DATA 50,170,60,125,70,170
480 DATA 69,170,77,129,86,170
490 DATA 84,170,90,132,96,170
500 DATA 90,170,95,137,100,170
510 DATA 99,170,105,142,109,170
520 DATA 107,172,112,148,115,172
530 DATA 115,174,118,152,122,174
540 DATA 131,174,134,151,138,174
550 DATA 136,174,140,146,144,172
560 DATA 140,172,145,147,150,172
570 DATA 150,172,156,149,162,172
580 DATA 160,172,167,144,174,172
590 DATA 170,172,178,142,186,172
600 DATA 186,170,194,140,202,170
610 DATA 199,170,208,135,217,170
620 DATA 210,170,219,132,228,170
630 DATA 219,170,229,129,239,170
640 DATA 238,170,255,125,255,170
650 GOTO 650

```

```

80 REM "PLANETE"      96 K
90 RANDOM
100 CLS
110 PROTECT 0
120 INK BLUE
130 FOR I=0 TO 254 STEP 85
140 MOVE 128,124
150 PLOT 3,150-I,123
160 NEXT I
170 FOR I=1 TO 6
180 LET A=2**I+128
190 MOVE 0,A
200 PLOT 3,255,0
210 NEXT I
220 FOR I=1 TO 6
230 LET A=2**I
240 MOVE 128,124
250 PLOT 3,-128,A
260 MOVE 128,124
270 PLOT 3,127,A
280 NEXT I
290 FOR I=1 TO 30
300 LET X=RAND(256)
310 LET Y=RAND(100)
320 INK RAND(7)+1
330 DOT X,Y
340 NEXT I
350 FOR I=20 TO 200 STEP 5
360 INK 7
370 EXT CIRCLE 1,I,(130-I/2)-9,I DIV 2
0
380 PAUSE 4000
390 INK 0
400 EXT CIRCLE 1,I,(130-I/2)-9,I DIV 2
0
410 NEXT I

```

```

420 INK 7
430 EXT CIRCLE 1,200,20,10
440 BEEP 200,1000.63
450 GOTO 450

```

```

10 REM "MAPPEMONDE"      46 K
100 CLS
140 DOT 100,0
150 DRAW 100,200
160 DOT 0,100
170 DRAW 200,100
180 FOR W=0 TO 200 STEP 10
190 DOT 99,W
200 DRAW 101,W
210 DOT W,99
220 DRAW W,101
230 NEXT W
240 DOT 0,0
250 DRAW 200,200
270 DRAW 200,0
280 DRAW 0,0
281 DRAW 0,200
282 DRAW 200,200
283 DOT 0,200
284 DRAW 200,0
800 FOR W=15 TO 100 STEP 15
810 FOR Z=0 TO 2*PI STEP PI/80
820 DOT 100+W*COS(Z),100+100*SIN(Z)
825 BEEP 50,1.50
830 DOT 100+100*COS(Z),100+W*SIN(Z)
835 BEEP 100,1.50
840 NEXT Z
850 NEXT W
860 GOTO 100

```

la boîte à outils
 SEGIMEX
 140, Boulevard HAUSSMANN
 PARIS
 75008

Et pour finir le cadeau-vacances de la Boîte à Outils : un jeu de Master Mind !

```

100 CLS
110 PROTECT 0
120 VDU 1,2,2,0,24
130 BEEP 200,50,63
140 PRINT @ 40,60;"STOP THE TAPE"
150 PAUSE 5000
160 CLS
170 BEEP 100,100,63
180 PRINT @ 40,10;"MASTER MIND"
190 VDU 1,1,2,0,25
200 PRINT @ 3,50;"Il s'agit de trouver
la combinaison de 5 pions de 8 couleurs
possibles"
210 VDU 1,7,2,0
220 REM "je vais construire mes PIONS da
ns la Procedure CHR"
230 PROC CHR
240 PRINT @ 3,80;CHR$(128);
250 VDU 1,0,2,7,21
260 PRINT @ 3,80;CHR$(129);
270 VDU 1,7,2,0
280 PRINT @ 15,80;"Qui a le code 0"
290 FOR I=1 TO 7
300 VDU 1,7,2,0
310 PRINT @ 3,80+10*I;CHR$(128);
320 VDU 1,1,2,0,21
330 PRINT @ 3,80+10*I;CHR$(129); Q
ui a le code ";I;
340 NEXT I
350 PRINT @ 3,180;"Pour commencer tape
sur une touche"
360 LET X=GETN
370 VDU 20
380 DIM J(5),A(5),B(5),C(5)
390 CLS
400 PRINT "JE CHERCHE MON JEU"
410 FOR I=1 TO 5
420 LET J(I)=RAND(8)
430 NEXT I
440 PRINT @ 3,5;"VOICI LES CODES DES CO
ULEURS"
450 REM "J'ecris sur l'écran chaque code
dans la couleur qu'il représente"
460 PROC COUL
470 VDU 19
480 PRINT "TAPE TON JEU";
490 BEEP 200,100,63
500 LET I=25,J=0,E=0
510 LET J=J+1,F=0
520 INPUT A(J)
530 IF A(J)<8 THEN GOTO 570
540 LET J=J-1
550 GOTO 510
560 REM "J'inscris sur l'écran le pion d
ans la couleur choisie"
570 PROC IMP
580 IF J>5 THEN GOTO 510
590 VDU 1,7,2,0,20
600 PRINT CHR$(32)+CHR$(32);
610 LET K=J+2,J=0,E=E+1
620 REM "Je compare le jeu proposé avec
le jeu du LYNX"
630 PROC COMP

```

```

640 IF F<>5 THEN GOTO 510
650 FOR I=0 TO 10
660 BEEP 100+10*I,100-5*I,63
670 NEXT I
680 VDU 1,2,2,0,24
690 PRINT "TU AS GAGNE"
700 PRINT CHR$(25); "TU AS JOUE ";"E;" FOI
S"
710 PRINT "SI TU VEUX REJOUER TAPE SUR S
"
720 IF GETN=83 OR GETN=115 THEN GOTO 39
0
730 PRINT "DEJA FATIGUE !!!"
740 END
750 DEFFPROC COUL
760 FOR I=0 TO 7
770 VDU 1,1,2,0
780 IF I=0 THEN PAPER 7
790 PRINT @ 6*I+12,15;I;
800 NEXT I
810 ENDPROC
820 DEFFPROC IMP
830 VDU 1,7,2,0,20
840 PRINT @ 42+6*I,I;CHR$(32);CHR$(128)
;
850 VDU 1,A(J),2,0,21
860 PRINT @ 42+6*I,I;CHR$(32);CHR$(129)
;
870 ENDPROC
880 DEFFPROC COMP
890 FOR N=1 TO 5
900 LET B(N)=J(N)
910 NEXT N
920 FOR M=1 TO 5
930 IF A(M)=B(M) THEN LET J=J+1,C(J)=
1,B(M)=9,F=F+1,A(M)=8
940 NEXT M
950 FOR M=1 TO 5
960 FOR L=1 TO 5
970 IF A(L)=B(M) AND M<>L THEN LET
J=J+1,C(J)=7,B(M)=9,A(L)=8,L=5
980 NEXT L
990 NEXT M
1000 IF J=0 THEN GOTO 1050
1010 FOR L=1 TO J
1020 VDU 1,C(L),2,0
1030 PRINT @ 42+6*(K+L),I;CHR$(130);
1040 NEXT L
1050 LET I=I+10,J=0
1060 VDU 1,7,2,0
1070 PRINT @ 3,I;"TAPE TON JEU ";
1080 BEEP 200,100,63
1090 ENDPROC
1100 DEFFPROC CHR
1110 RESERVE HIMEM-30
1120 DPOKE GRAPHIC,HIMEM
1130 FOR J=0 TO 29
1140 READ A
1150 POKE LETTER(128)+J,BIN(A)
1160 NEXT J
1170 DATA 000000,000000,000000,000000,00
1100,010010,010010,100001,100001,100001
1180 DATA 000000,000000,000000,000000,000000,00
0000,001100,001100,011110,011110,011110
1190 DATA 000000,000000,000000,000000,000000,00
0000,000000,000000,001100,001100,001100
1200 ENDPROC

```

INITIATION

INITIATION AU LANGAGE MACHINE

Dans ce numéro 2 de l'Œil de Lynx débute une série d'articles d'initiation au langage machine pour les débutants. De nombreux utilisateurs de Lynx, déjà familiers de la programmation en langage Basic, souhaitent en effet passer à l'étape suivante et maîtriser le langage machine et l'assembleur.

Comme pour tout apprentissage, il ne faut pas se contenter de lire les explications, mais bien sûr, essayer et modifier tous les exemples. Malgré son abord rébarbatif, le langage machine n'est pas très délicat à apprendre, mais demande beaucoup de soins et surtout de rigueur.

Dans cette première partie, nous examinerons les différents systèmes numériques (décimal, binaire et hexadécimal) et les bases du fonctionnement d'un micro-ordinateur.

1 – LES SYSTÈMES NUMÉRIQUES

2 – LOGIQUE ET MICRO-ORDINATEUR

3 – LE MICROPROCESSEUR

LES SYSTÈMES NUMÉRIQUES

Avant de décrire d'autres systèmes numériques, examinons celui qui nous est le plus familier : le système décimal. Celui-ci utilise 10 symboles, les "chiffres", de 0 à 9, avec lesquels on peut représenter n'importe quel nombre. Comme vous le savez certainement, il suffit de procéder de la manière suivante pour compter : 0,1,2,3,4,5,6,7,8,9,10,11,12...

Le procédé peut être décrit comme cela :

vous commencez par ... 0

symbole suivant ... 1

symbole suivant ... 2

dernier symbole ... 9

lorsque vous avez épuisé tous les symboles, il suffit de créer une nouvelle colonne, et de continuer ainsi :

symbole suivant .. 10

symbole suivant .. 11

symbole suivant .. 99

symbole suivant .. 100

Il existe une autre optique sous laquelle considérer le problème. Il suffit de chercher la signification de chacun des symboles en fonction de son numéro de colonne. Ainsi, n'importe quel nombre est fabriqué à partir des 10 chiffres selon le même principe. Par exemple, les trois premières colonnes sont celles des "unités", des "dizaines" et des "centaines".

$$1 = 0 \times 100 + 0 \times 10 + 1 = 001$$

$$23 = 0 \times 100 + 2 \times 10 + 3 = 023$$

$$134 = 1 \times 100 + 3 \times 10 + 4 = 134.$$

Vous pensez, bien sûr, connaître parfaitement tout ceci, mais la généralisation de ce principe est à la base de tous les systèmes numériques, utilisés par les ordinateurs. Lorsque vous en avez adopté un, il n'y a pas de difficulté pour en adopter un autre.

Voyons maintenant le système "binaire". Si dans le système décimal, vous avez besoin de 10 symboles, le système binaire n'en utilise que 2, le "0" et le "1". On procède exactement comme ci-dessus.

	binnaire	décimal
vous commencez par	0	0
symbole suivant	1	1
ainsi de suite	2	2
symbole suivant	9	9
symbole suivant	A	10
symbole suivant	B	11
symbole suivant	C	12
symbole suivant	D	13
symbole suivant	E	14
symbole suivant	F	15
nouvelle colonne	10	16

Comme pour le système décimal, il est possible de considérer la valeur de chaque chiffre en fonction de la colonne qu'il occupe : à la place des unités, dizaines et centaines, on trouve la colonne des unités, celle des 2, celle des

4, celle des 8, etc.

Binaire	8:4:2:1	Décimal
0	0:0:0:0	0
1	0:0:0:1	1
10	0:0:1:0	2
11	0:0:1:1	3
100	0:1:0:0	4
101	0:1:0:1	5
110	0:1:1:0	6
111	0:1:1:1	7
1000	1:0:0:0	8
1001	1:0:0:1	9

Vous savez tout désormais sur le système binaire. Voyons maintenant le système "hexadécimal", du grec "hexa" (6) et de "décimal" (10). Ce n'est donc pas une surprise que le système "hexadécimal" utilise 16 symboles différents. Les 10 premiers sont évidemment les chiffres de 0 à 9, mais il en reste 6, que l'on choisit logiquement parmi les premières lettres de l'alphabet : A,B,C,D,E,F. Le principe reste toujours similaire : vous commencez par 0 et continuez à utiliser tous les symboles. Après le dernier, il suffit de créer une nouvelle colonne. Les nombres du système hexadécimal ressemblent donc à cela :

Hex	Décimal
vous commencez par	0
symbole suivant	1
ainsi de suite	2
symbole suivant	9
symbole suivant	A
symbole suivant	B
symbole suivant	C
symbole suivant	D
symbole suivant	E
symbole suivant	F
nouvelle colonne	10

Cette fois, les colonnes correspondent aux différentes puissances de 16, c'est-à-dire 1, 16, 256, 4096, 65536, ...

Hex	4096:256:16:1	Décimal
0000	0: 0: 0: 0	0
000C	0: 0: 0:C	12
002A	0: 0: 0:A	42
01A2	0: 1: A:2	418
043C	0: 4: 3:C	1084
4E2A	4: E: 2:A	20010

Les nombres hexadécimaux sont souvent suivis du symbole H pour les distinguer du système décimal : ainsi, 10 correspond bien au nombre 10 du système décimal alors que 10H signifie 16. Pour éviter la confusion, les nombres binaires sont toujours rassemblés en groupe de 8, appelés "octets", comme ceci : 0011 1010.

ASSEMBLEUR (I)

LOGIQUE ET MICRO-ORDINATEURS

Le Lynx, comme tout micro-ordinateur, base son fonctionnement sur la logique, c'est-à-dire sur des propositions pouvant prendre les valeurs VRAI ou FAUX, ce qui devient en langage machine 0 ou 1 (de là l'importance du système binaire!). Les micro-ordinateurs stockent les valeurs logiques dans de la "mémoire". La capacité de stockage est mesurée en BITS. Le mot BIT est formé à partir de l'expression BiNary digITS. Le BIT est la plus petite unité de mémoire logique, c'est-à-dire qu'il peut prendre soit la valeur 1 (VRAI), soit la valeur 0 (FAUX). Cette unité élémentaire est plus rarement utilisée que son multiple, l'octet, ou groupement de 8 bits. L'octet est lui aussi une unité très petite, et l'on utilise couramment le concept de Kilo-octet. Contrairement aux apparences, le terme "Kilo" ne signifie pas exactement multiplié par 1000, comme dans kilomètre, mais par 1024. Cela provient du fait que 2 à la puissance 10 vaut 1024, c'est-à-dire la puissance binaire la plus proche de 1000.

Ainsi, lorsque l'on parle des 48 Ko (abréviation de Kilo-octet) de la mémoire du plus petit modèle de Lynx, cela signifie qu'il dispose de $48 \times 1024 = 49152$ octets.

On distingue deux types élémentaires de mémoire, appelées "vive" et "morte" en français et RAM et ROM en anglais. Le mot RAM est formé des initiales de Random Access Memory (mémoire à accès aléatoire), qui permet au programmeur la lecture et l'écriture à n'importe quel emplacement. Les mémoires ROM (à partir des initiales de Read Only Memory), ne peuvent être que lues, et il n'est pas possible de les modifier par la programmation.

LE MICROPROCESSEUR

Le Lynx utilise un microprocesseur très répandu, le Z80. Il peut distinguer directement 64 Ko de mémoire. Dans ce cas, chaque octet est distingué des autres par sa position mémoire : chacune de ces positions a une "adresse" spécifique. La première adresse est évidemment 0000, et la dernière, 65535 en décimal, soit FFFF en hexadécimal.

Le microprocesseur lui-même dispose de 208 bits de mémoire vive, groupés en 22 "registres" de 8 ou 16 bits.

Les 6 premiers registres principaux sont référencés de A à F, les deux autres étant dénommés H et L. Le registre A est souvent nommé Accumulateur, et

tire son nom parce qu'il est généralement utilisé pour "accumuler" le résultat des opérations. Le registre F contient des "drapeaux" (en anglais Flags) indiquant les effets des opérations en cours. Les registres B,C,D,E et H sont des registres courants, utilisés pour toutes les étapes intermédiaires. Tous ces registres peuvent fonctionner en registres doubles, ou paires, selon la disposition du diagramme (AF,BC,DE et HL) pour traiter deux octets simultanément.

Les huit registres suivants fonctionnent de façon tout à fait comparables, d'où leur nom de "registres alternatifs". Nous ne nous étendrons pas sur leur emploi, puisque tout ce qui sera dit sur les registres principaux peut être étendu aux registres alternatifs.

Les six "registres spéciaux" correspondent à des applications particulières que nous détaillerons au cours de leur utilisation. Signalons simplement que quatre d'entre eux sont des registres doubles, c'est-à-dire fonctionnant sur 16 bits. Pour l'instant, nous n'utiliserons que le registre PC (de l'anglais Program Counter, compteur du programme) dont le Z80 utilise les 16 bits pour "pointer" l'instruction suivante.

Un programme en langage machine est constitué d'instructions qui (généralement) réalisent immédiatement une opération utile. Pour lancer un programme, le registre PC est d'abord rempli par le premier octet du programme. Le Z80 effectue ensuite la première opération. Le PC est directement (par le microprocesseur) pointé par l'instruction suivante, qui est ensuite lue et effectuée par le Z80, les instructions lues par le Z80 sont codées sur 8 bits. Théoriquement, il n'y a donc que 256 codes distincts. Mais pratiquement, l'utilisation de préfixes permet au Z80 d'en reconnaître près de 600.

Puisque le Z80 utilise des instructions codées en système binaire, il peut bien évidemment être programmé de la même manière. Ainsi, le petit exemple suivant :

00100001
01011011
01100010
01111110
01100011
00100011
01110111
00101011
01110000
11001001

sera tout à fait intelligible par le Z80.

Mais par vous? Le risque d'erreurs dans ce type de programmation est évidemment très important, et la plupart des programmeurs se sentent incapables de programmer directement en binaire.

Considérons maintenant un octet quelconque, par exemple 01011101. Il est possible de le considérer comme formé par la réunion de deux "quadruplets", soit 0101 et 1101. Le premier quadruplet vaut 5 en décimal ($0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$) et le deuxième 13 ($1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$). En hexadécimal, 5 est toujours représenté par 5, mais 13 devient D. L'octet considéré peut donc être représenté par le nombre hexadécimal 5D. Le plus grand quadruplet, 1111, est représenté par le symbole F. Le petit programme binaire devient maintenant :

21
5B
62
7E
46
77
2B
70
C9

Il ne signifie sans doute rien de plus pour vous, mais vous conviendrez sans peine que cette notation est plus simple que le binaire précédemment utilisé. L'hexadécimal est bien plus clair pour l'esprit humain.

Mais si nous préférions l'emploi de l'hexadécimal, le Z80 quant à lui persiste à ne comprendre que le binaire! Une "traduction" est donc nécessaire. Elle est normalement réalisée par ce que l'on nomme un chargeur hexadécimal. Heureusement, le Lynx dispose d'un tel programme en ROM. Il s'agit de la commande M du moniteur. Cette instruction nous permet d'examiner n'importe quelle adresse de la mémoire, et, s'il s'agit bien entendu de RAM, d'y écrire ce que l'on veut. Nous pouvons donc désormais entrer notre programme. Notre seul problème est de savoir où! C'est d'ailleurs la principale difficulté posée par la programmation en langage machine. Dans la seconde partie de cette initiation, nous examinerons donc plus particulièrement la "carte mémoire" du Lynx, c'est-à-dire l'emplacement des différentes adresses utiles.

SPÉCIAL FICHIER

De nombreux utilisateurs de micro-ordinateurs familiaux souhaitent utiliser leur machine autrement que pour jouer. L'Œil de Lynx N° 1 vous a donné deux exemples d'applications plus "sérieuses" : le tracé de courbes et la conception graphique. Le dossier spécial de ce mois-ci ne comprend qu'un seul programme, vu sa longueur ! Il transforme en effet votre Lynx 48 ou 96 Ko en un gestionnaire de fichiers très performant.

Si les utilisateurs de micro-ordinateurs pourvus de disquettes n'ont pas de problèmes pour la gestion de fichier, il n'en est pas de même pour les machines ne disposant que d'un magnétophone à cassettes standard. Il n'est alors pas possible de se déplacer aisément sur le support magnétique et tous les tris et recherches doivent s'effectuer en mémoire centrale. Le listing suivant répond à ces problèmes. Vous pouvez bien sûr l'entrer tel quel dans votre ordinateur, mais aussi le modifier à votre gré. A cette fin, nous avons rassemblé dans l'encadré "Structure du programme" tous les commentaires nécessaires.

```
10 TEXT
30 DIM N$(50)
50 DATA Chargement catalogue,Creation rubrique,Creation fiche,Liste des rubriques,Recherche libre,Modification,Suppression,Sauvegarde,Effacement memoire,9430,100
     ,690,500,2600,2300,2000,9300,9900,
60 LET D=&8900,H=D,R=1,T=&6255
80 RESERVE D-3
82 WHILE PEEK(H)=R
83   LET H=H+PEEK(H+1),R=R+1
84 WEND
85 GOTO 4002
87 PRINT "Aucune rubrique presente"
88 LET F$=""
90 RETURN
95 VDU 28,28,28,28,6,6,30
99 REM creation fichier
100 PROC RNL(H,D)
105 LET I=N
110 PRINT "Intitule de la rubrique ";N;
     :";
120 INPUT N$
130 PROC INS(LEN(N$)+3,1,N)
140 PROC ECR
150 PROC RNL(D+PEEK(D+1),D+PEEK(D+2))
160 IF N=1 THEN GOTO 290
170 LET i=N
180 PRINT "Complement de la rubrique"
190 LET k=1
200 PROC INT(1,0)
210 PRINT " ";
220 PROC INT(1,k)
225 PRINT "....";
230 PROC INT(I,0)
235 LET a=R+1
240 PROC RNL(R+PEEK(R+1),R+PEEK(R+2))
```

```
250 INPUT " :";N$
260 PROC INS(LEN(N$)+3,0,N)
265 PROC ECR
270 LET k=k+1
280 IF k<i THEN GOTO 200
290 PRINT "Autre fiche(O/N)?"
300 IF GETN=78 THEN GOTO 4002
310 GOTO 95
499 REM Listing fichier
500 CLS
510 PROC LIF(0)
520 INPUT "Numero de la rubrique a liste
r (si aucune 0)";K
522 IF K=0 THEN GOTO 4002
525 CLS
530 PROC LIF(K)
534 IF PEEK(R+1)=PEEK(R+2) THEN GOTO 40
00
535 IF N=0 THEN GOTO 850
536 INPUT "N. de fiche a lister (si aucu
ne 0)";K
537 IF K<>0 THEN PROC SOR(K)
538 IF N=0 THEN GOTO 850
540 PRINT "Autre rubrique? (O/N)"
550 IF GETN=78 THEN GOTO 500
560 GOTO 4000
670 REM Creation rubrique
680 VDU 28,28,28,28,6,6,30
690 LET F=1
710 PROC REC(F,H,D)
720 LET a=R+1
730 IF N=0 THEN GOTO 830
740 PROC INT(F,0)
750 PROC RNL(R+PEEK(R+1),R+PEEK(R+2))
780 INPUT " :";N$
790 PROC INS(LEN(N$)+3,0,N)
800 PROC ECR
810 LET F=F+1
820 GOTO 710
830 PRINT "Autre rubrique (O/N)?"
835 IF GETN=78 THEN GOTO 4002
837 GOTO 680
850 PRINT "NUMERO INCONNU"
860 GOTO 4000
900 DEFPROC EFF(F,f)
920 PROC REC(F,H,D)
940 IF N=0 THEN ENDPROC
960 IF f=0 THEN PROC DST
975 ELSE GOTO 1020
980 WHILE R<H
985   POKE R,PEEK(R)-1
990   LET R=R+PEEK(R+1)
995 WEND
1000 ENDPROC
1020 LET a=R+1
1040 PROC REC(f,R+PEEK(R+1),R+PEEK(R+2))
1060 IF N=0 THEN ENDPROC
1080 PROC DST
1100 POKE a,PEEK(a)-X
1120 ENDPROC
2000 PRINT "Pour supprimer :;"Une rubri
que ..Tapez (R)";CHR$(31);CHR$(9);CHR$(9
);;"Une fiche ..Tapez (F) "
2010 IF KEYN=82 THEN GOTO 2040
2020 IF KEYN=70 THEN GOTO 2080
2030 GOTO 2010
2040 PROC LIF(0)
```

```

2050 INPUT "N. de la rubrique a supprime
r :";F
2060 PROC EFF(F,0)
2070 GOTO 4002
2080 PROC LIF(1)
2090 INPUT "N. de la rubrique a supprimme
r :";f
2100 LET F=1
2110 PROC EFF(F,f)
2120 IF N=0 THEN GOTO 4002
2130 LET F=F+1
2140 GOTO 2110
2300 REM
2301 PROC LIF(1)
2310 INPUT "N. de la fiche a modifier";k
2315 PROC REC(k,D+PEEK(D+1),D+PEEK(D+2))
2316 IF N=0 THEN GOTO 850
2320 PRINT "Fiche a modifier entierement
(O/N)?""
2330 IF GETN=79 THEN GOTO 2490
2340 PRINT "Dans quelle rubrique?"
2350 PROC LIF(0)
2360 INPUT F
2370 PROC INT(F,0)
2373 IF N=0 THEN GOTO 850
2375 PRINT " a modifier :";
2380 PROC INT(F,k)
2383 IF N=0 THEN GOTO 850
2385 VDU 31
2390 PROC INT(F,0)
2400 PROC EFF(F,k)
2430 INPUT ":";N$
2440 PROC REC(F,H,D)
2450 LET à=R+1
2460 PROC RNL(R+PEEK(R+1),R+PEEK(R+2))
2465 PROC INS(LEN(N$)+3,0,N)
2470 PROC ECR
2475 PROC SOR(k)
2480 GOTO 4000
2490 LET F=1
2500 PROC EFF(F,k)
2510 IF N=0 THEN GOTO 680
2520 LET F=F+1
2530 GOTO 2500
2600 PRINT "Voulez-vous faire une recher
che par :"
2610 PROC LIF(0)
2620 INPUT "N.de la rubrique choisie";K
2624 PROC REC(K,H,D)
2625 IF N=0 THEN GOTO 850
2630 PRINT "Voulez-vous lister cette rub
rique (O/N)?""
2640 IF GETN=79 THEN PROC LIF(K)
2650 INPUT "Mot clef de recherche";N$
2655 CLS
2660 PROC CLF(K)
2663 IF c<>0 THEN PRINT N$;" inconnu da
ns cette rubrique"
2665 IF N=0 THEN GOTO 850
2670 PRINT "Autre recherche (O/N)?""
2680 IF GETN=79 THEN GOTO 2650
2690 GOTO 4002
2800 DEFPROC CLF(F)
2810 PROC REC(F,H,D)
2815 IF N=0 THEN ENDPROC
2820 LET a=R+PEEK(R+1),c=1
2830 LET R=R+PEEK(R+2)
2840 LET x=PEEK(R),X$=""
```

```

2850 FOR n=3 TO LEN(N$)+2
2860 LET X$=X$+CHR$(PEEK(R+n))
2870 NEXT n
2875 LET b=R
2880 IF X$=N$ THEN PROC SOR(x)
2890 LET R=b+PEEK(b+1),N=1
2900 IF R>=a THEN ENDPROC
2910 GOTO 2840
4000 PRINT "Retour menu :Pressez une tou
che"
4001 IF GETN THEN CONT
4002 CLS
4005 PRINT "MENU"
4010 VDU 10,10
4020 RESTORE
4030 FOR c=1 TO 9
4040 READ N$
4050 PRINT c;"...."N$
4060 NEXT c
4080 IF H=D THEN GOSUB 87
4100 LET K=GETN-49
4110 IF H=D AND K>1 THEN GOTO 4080
4120 CLS
4140 FOR c=c TO c+K
4150 READ K
4160 NEXT c
4170 GOTO K
4999 STOP
5000 DEFPROC RNL(e,R)
5010 LET N=1
5020 IF N<>PEEK(R) THEN ENDPROC
5024 ELSE LET N=N+1,R=R+PEEK(R+1)
5050 IF R>=e THEN ENDPROC
5060 GOTO 5020
5200 DEFPROC SOR(k)
5205 IF c=0 THEN PRINT "-----";
5210 LET F=1,c=0
5220 IF PEEK(T)>=230 THEN LET K=GETN
5225 PROC INT(F,0)
5230 IF N=0 THEN GOTO 5280
5240 PRINT ":" ;
5250 PROC INT(F,k)
5255 IF N=0 THEN ENDPROC
5256 PRINT
5260 LET F=F+1
5270 GOTO 5220
5280 LET N=1
5290 ENDPROC
6000 DEFPROC INS(X,x,p)
6020 FOR n=H TO R STEP -1
6040 POKE (n+X),PEEK(n)
6045 NEXT n
6050 LET H=H+X
6060 POKE R,p
6080 POKE R+1,LEN(N$)+3
6100 POKE R+2,(LEN(N$)+3)*x
6120 IF x=0 THEN POKE a,PEEK(a)+x
6140 ENDPROC
6500 DEFPROC REC(N,h,R)
6520 IF PEEK(R)=N THEN ENDPROC
6540 ELSE LET R=R+PEEK(R+1)
6560 IF R<h THEN GOTO 6520
6580 LET N=0
6600 ENDPROC
7000 DEFPROC ECR
7010 FOR n=1 TO LEN(N$)
7020 POKE R+n+2,ASC(MID$(N$,n,1))
```

```

7030 NEXT n
7040 ENDPROC
8000 DEFPROC DST
8010 LET X=PEEK(R+1)
8020 FOR n=R+X TO H
8040 POKE (n-X),PEEK(n)
8060 NEXT n
8080 LET H=H-X
8090 FOR n=H TO H+X
8100 POKE n,0
8110 NEXT n
8120 ENDPROC
9000 DEFPROC LIF(h)
9005 IF h=0 THEN GOTO 9025
9009 PROC REC(h,H,D)
9010 IF N=0 THEN ENDPROC
9012 PROC IMP(2)
9014 VDU 31,31
9015 IF PEEK(R+1)=PEEK(R+2) THEN GOTO 9
190
9016 LET h=R+PEEK(R+1),R=R+PEEK(R+2),a=1
9020 GOTO 9040
9025 LET R=D,h=H,a=2
9040 PRINT PEEK(R);". . .",
9080 PROC IMP(a)
9120 PRINT
9124 IF PEEK(T)<225 THEN GOTO 9140
9125 PRINT "Page suivante (O/N)"
9126 IF GETN=78 THEN ENDPROC
9140 LET R=R+PEEK(R+1)
9160 IF R<=h-1 THEN GOTO 9040
9180 ENDPROC
9190 PRINT "Rubrique vide"
9200 ENDPROC
9300 IF H=D THEN GOTO 85
9310 PRINT "Nom du fichier :";F$,CHR$(31)
);"Voulez-vous changer (O/N)?"
9320 IF GETN=78 THEN GOTO 9335
9330 INPUT "Nouveau nom :";F$
9335 PRINT "Demarrez votre magnetophone
/ RETURN"
9336 IF GETN THEN CONT
9340 LET N$=F$
9350 DPOKE D-3,H
9380 PROC SAVE(D-3,D-1)
9400 PROC SAVE(D,H)
9410 PRINT "Sauvegarde effectuée"
9420 GOTO 4000
9430 INPUT "Nom du fichier à charger :";
F$
9435 PRINT "Demarrez votre magnetophone"
9437 LET N$=F$
9440 PROC LOAD(D-3,D-1)
9450 LET H=DPEEK(D-3)
9470 PROC LOAD(D,H)
9480 GOTO 4002
9500 DEFPROC IMP(r)
9510 FOR n=3 TO PEEK(R+r)-1
9520 PRINT CHR$(PEEK(R+n));
9530 NEXT n
9540 ENDPROC
9600 DEFPROC INT(F,f)
9610 PROC REC(F,H,D)
9620 IF N=0 THEN ENDPROC
9630 IF f=0 THEN PROC IMP(2)
9640 ELSE GOTO 9660
9650 ENDPROC

```

```

9660 PROC REC(f,R+PEEK(R+1),R+PEEK(R+2))
9665 IF N=0 THEN ENDPROC
9670 PROC IMP(1)
9680 ENDPROC
9700 DEFPROC SAVE(d,e)
9710 POKE e,&0080
9720 POKE LCTN(9830)-1,&001F
9730 GOTO 9750
9740 DEFPROC LOAD(d,e)
9745 POKE LCTN(9830)-1,&001E
9750 WHILE LEN(N$)<30
9760 LET N$=N$+" "
9770 WEND
9780 LET R=LCTN(9830)-2
9790 PROC ECR
9810 LET Y=DPEEK(&61FA),Z=DPEEK(&61FC)
9815 DPOKE &61FA,d
9820 DPOKE &61FC,e
9830 LOAD "CO.CON."
"
9840 DPOKE &61FA,Y
9850 DPOKE &61FC,Z
9860 ENDPROC
9900 FOR n=D TO H
9910 POKE n,0
9920 NEXT n
9930 GOTO 10

```

STRUCTURE DU PROGRAMME

- 60 : o est l'adresse du début de la zone mémoire réservée au catalogue, H celle du dernier octet de cette zone
- 180 : si on ajoute une rubrique pendant le remplissage du catalogue, on définit cette rubrique dans toutes les fiches déjà existantes
- 900 : définition de la procédure d'effacement : efface la fiche f dans la rubrique F. Si f=0, efface toutes les fiches
- 2800 : procédure de recherche d'une fiche commençant par N\$ dans la rubrique F
- 4000 : affichage du menu et renvoi à la fonction choisie
- 5000 : recherche d'un numéro de rubrique ou de fiche libre de l'adresse R à l'adresse e
- 5200 : édition de la fiche (K)
- 6000 : routine d'insertion d'une fiche ou d'une rubrique de largeur X et de numéro n si x=0, fiche, si x=1, rubrique
- 6500 : recherche de la fiche ou de la rubrique N
- 8000 : effacement de la rubrique ou de la fiche N
- 9000 : liste la rubrique h à l'écran
- 9600 : affiche à l'écran l'intitulé de la rubrique F si f=0, ou de la fiche f dans la rubrique F si f<>0
- 9700 : routine de sauvegarde et de chargement d'un fichier sur cassette.