



MOBILE COMPUTER

- L'OPL et la calculatrice
- Les mémoires de masse des Mobile Computer



ORGANISEUR II

- Le code à barres
- Trier un fichier sur Rampak, Flash Datapak ou Datapak
- Course de voitures
- Une aide sur la base de données
- TTRN & RTRN
- SHIFT EXE & SHIFT SPACE



PC4

- Une procédure d'entrée

ERRARE HUMANUM EST

Édité par : Aware
21, rue Olivier Métra
75020 Paris
Tél : (1) 46 36 46 47
Télex : 216 244 F
Fax : (1) 46 36 82 54
3616 code Aware



Aware a déménagé

A la fin de l'année 1990 (oui oui, je sais, ce numéro est daté de juillet !), un événement d'importance est venu pimenter mais aussi -un tant soit peu- perturber la vie des "Awariens" : en effet, nous venons de changer d'adresse.

Pour mieux vous répondre ou mieux vous recevoir, nous sommes passés de 400 m2 dans un quartier plus qu'encombré, à 800 m2 sur 5 étages dans un quartier mieux famé et surtout plus accessible (je pense là à nos visiteurs automobilistes puisque nous sommes à 5 minutes de la Porte des Lilas !). Grâce à ces locaux clairs, spacieux et agréables, nous sommes désormais frais et dispos pour accroître notre efficacité et mieux vous servir.

Alors, n'oubliez pas si vous voulez nous écrire ou nous téléphoner, de le faire à l'adresse suivante

Aware
21, rue Olivier Métra - 75020 Paris
Tél. (1) 46 36 46 47 - Fax (1) 46 36 82 54

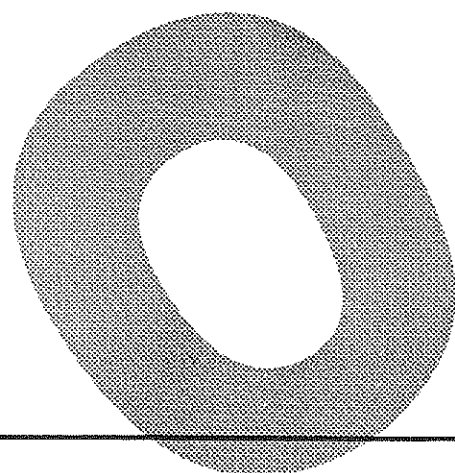
et non plus à la Rue des Petites Ecuries.

Dans ce numéro, vous trouverez des sujets d'articles aussi variés que les codes à barres, les différents supports de stockage (dont les nouveaux Flash Pack 256K) ou un petit jeu à réaliser soi-même (une course de voitures). Bien d'autres encore vous donneront une multitude d'informations précieuses pour approfondir vos connaissances de l'Organiseur ou du MC.

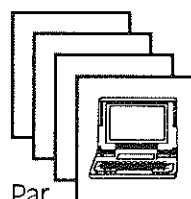
Enfin, et je devrais dire 'Avant tout', puisque nous sommes au début de l'année 1991, j'en profite pour vous présenter les meilleurs voeux de toute l'équipe et vous souhaiter toujours plus de satisfaction lorsque vous utilisez nos produits.

Excellent début d'année et à très bientôt !

Martine Garrigues



L'OPL et la calculatrice



Par
Gilles Jean

2

La calculatrice a été réalisée pour exécuter des calculs simples mais aussi des calculs très compliqués. Comme le titre en témoigne, nous allons traiter aujourd'hui le cas complexe du langage de programmation des MC200/MC400. Vous observerez, l'expérience venant, la formidable puissance de l'outil qui est entre vos mains.

Ces machines mettent à notre disposition une panoplie de fonctions numériques dont nous pouvons nous servir aussi bien avec la calculatrice qu'avec le langage de programmation OPL. Cela signifie que nous pouvons rédiger en OPL une procédure contenant une ou plusieurs formules de calcul que nous utilisons fréquemment afin de nous simplifier la vie ou de gagner encore un peu plus de temps. Dans ce même cas de figure, nous pouvons envisager la création de nos propres fonctions numériques utilisables par exemple au bureau.

La procédure EMP, listée ci-contre, permet de retourner à un programme appelant le résultat des calculs effectués qui correspond à des routines de calculs financiers concernant les emprunts bancaires. L'utilisation en est simple. Nous entrons trois des quatre paramètres nécessaires au calcul et le quatrième est alors calculé. Il est toutefois important de noter que l'ordre des variables est immuable et que le paramètre à calculer doit impérativement avoir une valeur 0. Utilisons l'éditeur de texte pour la saisir et la transcoder (Voir manuel de programmation).

```
proc emp: (m,t,a,p)
  local tx
  tx=t/100
  if m=0
    rem *****
    rem calcul du montant du prêt *
    rem *****
    return (p*(12*(1-(1+tx/12)**(-12*a)))/tx)
  elseif t=0
    rem *****
    rem calcul du taux du prêt *
    rem *****
    rem Il est possible d'améliorer la *
    rem précision du calcul en diminuant *
    rem l'incréméntation de la variable tx *
    rem *****
    do
      tx=tx+.0005
    until (m*tx/12/(1-((1+tx/12)**(-12*a))))>p
    return tx*100
  elseif a=0
    rem *****
    rem calcul de la durée du prêt *
    rem *****
    return (ln((12*p)/(12*p-m*tx))/(12*ln(1+tx/12)))
  elseif p=0
    rem *****
    rem calcul de la mensualité du prêt *
    rem *****
    return (m*tx/12/(1-((1+tx/12)**(-12*a))))
  endif
endp
```

3

Nous savons qu'une routine OPL peut retourner une valeur numérique dans la calculatrice. C'est ce point que nous allons développer maintenant. Cliquons sur l'icône de la calculatrice pour la sélectionner et validons par la touche Enter. En appuyant simultanément sur les touches Psion et O, nous pouvons alors choisir la procédure à charger en mémoire. Attention, il est impératif de charger le fichier comportant la procédure que nous allons utiliser. Validons le fichier en question. Nous pouvons maintenant utiliser la procédure EMP comme les exemples suivants :

Calcul d'une mensualité d'emprunt :

emp: (30000,12,2,0)

Calcul de la durée d'un emprunt :

emp: (30000,12,0,1412.20)

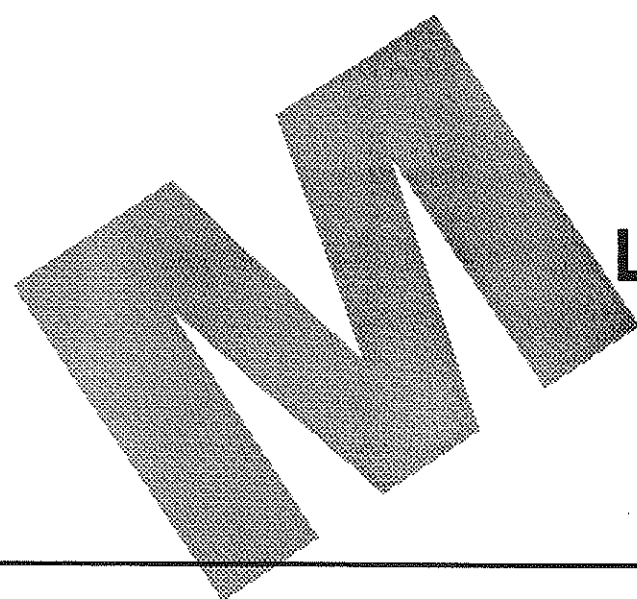
Calcul du taux d'un emprunt :

emp: (30000,0,2,1412.20)

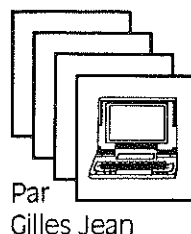
Calcul du montant de l'emprunt :

emp: (0,12,2,1412.20)

A vous maintenant de donner la pleine mesure de cet outil puissant qu'est le Mobile Computer.



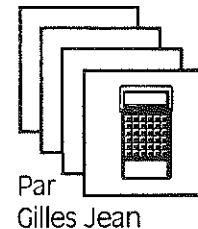
Les mémoires de masse des Mobile Computers



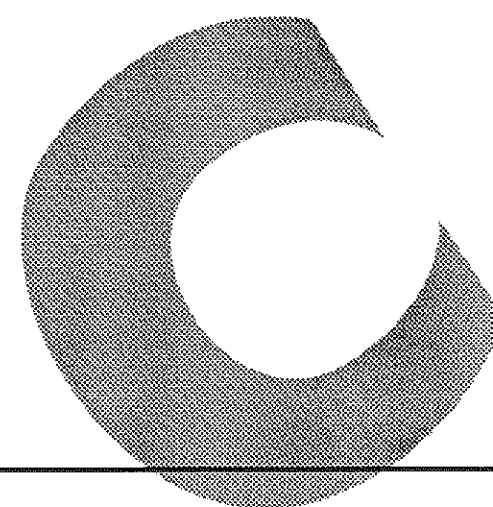
Par
Gilles Jean

Il règne, s'il on en croit vos questions sur notre serveur 3616 Aware, une certaine confusion concernant le nombre total de fichiers et de répertoires pouvant être créés à la racine d'un SSD (Solid State Disk). Nous allons tenter d'éclaircir ce point avec le tableau ci-dessous :

Type de Support		Nb max de fichiers/répertoires
SSD 64k	Ram	16
SSD 128k	Ram	32
SSD 512k	Ram	64
SSD 1Mb	Ram	128
SSD Chaque	Flash	>8000
Disque 720k	Floppy	112
Disque 1.44Mb	Floppy	224



Par
Gilles Jean



Le code à barres

Sans pour autant que cet article puisse être considéré comme un cours général sur les codes à barres, nous allons tenter de vous expliquer clairement les avantages et les inconvénients de ce système.

Comme vous l'avez constaté, un code à barres est constitué de barres et d'espaces, de différentes largeurs, représentant des informations numériques ou alpha-numériques. Il est aujourd'hui le procédé d'identification dont le ratio coût/simplicité/fiabilité/rapidité/flexibilité est le plus intéressant pour les entreprises.

La saisie des codes à barres s'effectue avec des lecteurs appropriés. Ces lecteurs émettent une source lumineuse constituée de un ou plusieurs rayons. Ces rayons sont alors absorbés par les barres alors que les espaces les réfléchissent. Ces variations de lumières sont converties en signaux électriques puis sont traités par un décodeur qui après validation, retourne une forme vers l'Organiseur II.

Il existe plusieurs technologies de lecteurs que nous allons passer en revue sous forme de tableau présentant les avantages et les inconvénients. Les informations de ce document se veulent volontairement générales et certains constructeurs peuvent proposer des alternatives intéressantes à un ou plusieurs inconvénients de cette liste. Vérifiez donc les caractéristiques du lecteur de votre choix.

Le stylo lecteur

Avantages

Coût réduit
Poids très faible
Consommation faible
Maintenance aisée
Lecture multi-taille

Inconvénients

Déplacement sur le support
Usure du support
Vitesse de lecture
Lecture au contact direct

La douchette

Avantages

Pas de déplacement
Pas ou peu de contact
Poids raisonnable
Pas d'usure du support
Utilisation facile
Vitesse de lecture

Inconvénients

Taille du support limité
Consommation moyenne
Prix

Le lecteur laser

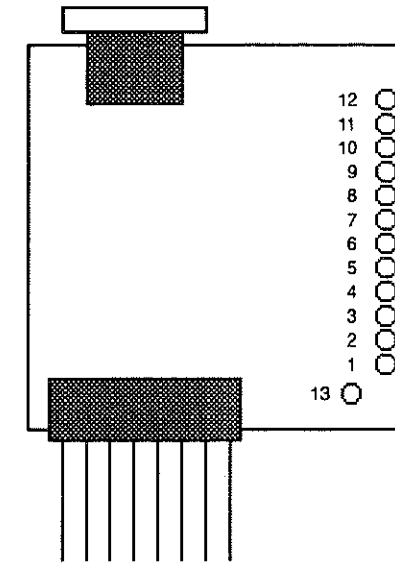
Avantages

Pas de déplacement
Lecture à distance
Lecture multi-taille
Utilisation facile
Vitesse de lecture
Précision lecture

Inconvénients

Consommation très importante
Prix

Vous avez certainement remarqué que nous ne disposons pas à notre catalogue de l'ensemble des éléments cités ci-dessus. Les lecteurs que nous distribuons disposent du meilleur rapport consommation/performance. Il est pourtant possible de connecter un autre lecteur, soit d'un autre type (stylo infra-rouge, télécamera, lecteur de carte codes à barres, lecteur à défilement, etc...), soit d'un autre fabricant sur l'interface de décodage PSION. Il faut toutefois vous assurer que vous connecterez un lecteur TTL sur cette interface dont le schéma ci-dessous vous explique la connectique. La Prom contenant l'intelligence de décodage mise au point par Psion étant toujours présente, elle permettra au développeur d'avoir en permanence à sa disposition la commande BAR\$. Une complète compatibilité sera ainsi assurée avec d'autres applications ou d'autres lecteurs.



Contact 12 Alimentation + 5 volts
Contact 10 Signal TTL
Contact 1 Masse

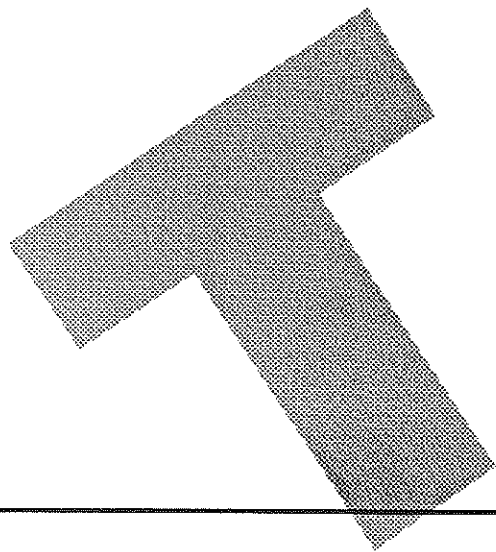
Le reste des contacts n'est pas utilisé mais il faut noter que les contacts 1, 2 et 13 sont reliés sur un seul et même circuit et sont donc interchangeables. Un autre élément doit être pris en considération, si vous optez pour un autre lecteur : c'est la consommation dudit lecteur. En effet, l'Organiseur II peut fournir jusqu'à concurrence de 150 mA (milli-ampère) interface de décodage comprise. Au delà, le lecteur devra obligatoirement disposer de sa propre alimentation.

Nous avons réalisé plusieurs tests avec les produits de notre catalogue en respectant le cahier des charges suivant :

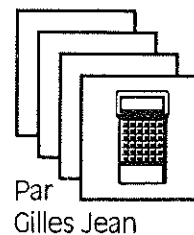
- Un Organiseur POS 350 + pile alcaline Duracell MN1604 6LR61.
- Un lecteur listé ci-dessous.
- Le même code est lu en continu puis stocké dans le fichier MAIN de la mémoire vive interne.

Lecteur HP noir	13000 lectures
Lecteur métal	10000 lectures
Douchette Spotron	11000 lectures
Douchette Lemco 7	6500 lectures

Pour conclure, il faut retenir que le choix du lecteur est dépendant du cahier des charges et des spécifications de l'application à réaliser. Prenez toujours bien soin d'analyser avec la plus grande rigueur le mode d'utilisation du terminal portable. C'est ainsi que vous pourrez au mieux conseiller un type de lecteur plutôt qu'un autre. N'hésitez pas à demander conseil, notre expérience et notre connaissance du marché en la matière peut vous apporter une solution à vos problèmes.



Trier un fichier sur Rampak Flash Datapak ou Datapak



Par
Gilles Jean

Si vous possédez un Organiseur modèle LZ ou LZ64, vous disposez déjà d'une option de tri alphanumérique dans le menu Xfich, permettant le classement du contenu d'un fichier situé uniquement sur la ram interne de la machine. Utilisateur d'une base de données, d'un fichier ou tout simplement du calepin électronique, voici un utilitaire qui vous permettra de trier sur la première rubrique de votre fichier et dans l'ordre croissant (de A vers Z) quel que soit l'emplacement physique du fichier (mémoire vive de la machine ou mémoire de masse). La portabilité de cette application sur une machine dotée d'un affichage deux lignes est aisée et expliquée dans la suite de l'article.

La routine "tri" listée ci-dessous permet de lancer le programme et intègre la gestion des éventuelles erreurs. Elle appelle la routine de saisie du nom du fichier à trier et la routine (compte tenu de la valeur retournée par la routine de saisie du nom de fichier) effectuant le tri et les mises à jour des enregistrements dans le fichier.

```
tri:
global p$(1),n$(8)
onerr e::
top::
cls
rem *** definition de l'icone ****
udg 0,0,16,24,28,30,31,0,31
```

Suite du programme Tri

```
rem *** affichage pour LZ *****
print chr$(0);rept$(chr$(2),14)
p$="A" :clock(1)
rem *** appel de la routine de ***
rem *** saisie du fichier *****
if snf:
trialpha:
endif
return
rem *** fin logique du programme ***
e::
rem *** affichage des erreurs *****
cls
print "ERREUR NUM",err
print err$(err)
print rept$(chr$(2),20);"Tapez une touche";chr$(16)
get
goto top::
```

Lecteur assidu de nos colonnes vous reconnaîtrez facilement la routine listée ci-dessous puisqu'elle nous sert traditionnellement à contrôler la saisie utilisateur d'un nom de fichier étant très proche de l'interface système de l'Organiseur II. La seule modification à apporter pour la porter sur une machine avec deux lignes d'affichage consiste à remplacer le chiffre 20 par le chiffre 16. Lorsque la saisie est correctement effectuée, la routine ouvre le fichier et retourne la valeur 1 au programme appelant. La valeur 0 est retournée dans le cas contraire.

```
snf:
local p%,k%
do
at 1,2
print chr$(15)+left$("Trier "+p$+" ":"+n$,20)
at 9+p%,2
cursor on
k%=get
if k%=1
if n$<>""
n$=""
p%=0
else
return 0
endif
elseif k%=2
do
if p$="A"
p$="B"
elseif p$="B"
p$="C"
else
p$="A"
endif
```

Suite du programme snf

```
until exist(p$+":main")
p%=0
elseif k%=5 and p%
p%=p%-1
elseif k%=6 and p%<len(n$)
p%=p%+1
elseif k%=7 and p%<len(n$)
n$=left$(n$,p%)+mid$(n$,p%+2,20)
elseif k%=8 and p%
n$=left$(n$,p%-1)+mid$(n$,p%+1,20)
p%=p%-1
elseif k%>32
if len(n$)=8
beep 99,99
else
n$=left$(n$,p%)+chr$(k%)+mid$(n$,p%+1,20)
p%=p%+1
endif
elseif k%=13 and len(n$)>0
open
p$+": "+n$,a,a$,b$,c$,d$,e$,f$,g$,h$,i$,j$,k$,l$,m$,n$,o$,p$
return 1
else
beep 99,99
endif
until 0
```

10 La portabilité de cette routine pour un Organiseur II deux lignes n'est liée qu'aux commandes et fonctions d'affichage (at 1,4 à remplacer par 1,2 et at 15,4 par 15,2)

```
trialpha:
local der%,ref$(255),ref%,lpos%
at 1,4
print"Tri en cours..."
last
der%=pos
if count>0
while der%<>0
position der%
ref%=pos
ref$=upper$(a.a$)
do
if upper$(a.a$)<ref$
ref$=upper$(a.a$)
ref%=pos
endif
lpos%=pos
back
until pos=1 and lpos%=1
position ref%
at 15,4
print"<";mid$(gen$(100+int((abs(der%-count)/count)*100),3),2-
,2);"%>"
```

Suite du programme trialpha

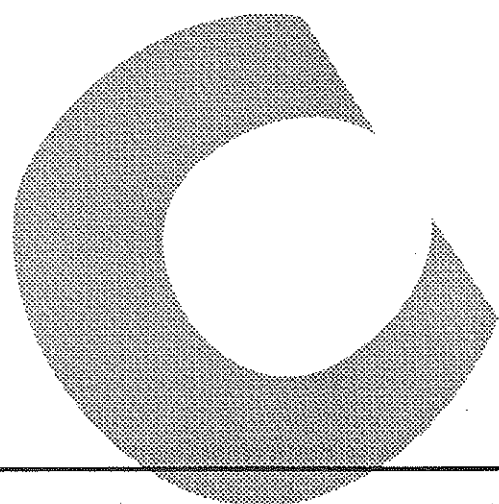
```
update
der%=der%-1
endwh
endif
beep 300,300
beep 200,800
beep 300,300
```

Comment réaliser un tri multi-rubriques ?
Il vous suffit, dans la routine TRIALPHA, de remplacer toutes les chaînes a.a\$ par la concaténation des différentes rubriques constituant le tri.
Exemple : Vous voulez trier sur la 2ème rubrique et sur la 5ème rubrique de votre fichier. Il vous faut remplacer 'a.a\$' par 'a.b\$+a.e\$'.

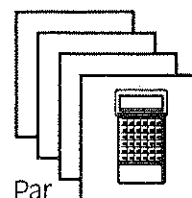
Comment réaliser un tri numérique ?
Il vous suffit de remplacer la routine TRIALPHA par la routine listée ci-dessous :

```
trin:
local der%,rubnum,ref%,lpos%
at 1,4
print"Tri en cours..."
last
der%=pos
if count>0
while der%<>0
position der%
ref%=pos
rubnum=val(a.a$)
do
if val(a.a$)>rubnum
rubnum=val(a.a$)
ref%=pos
endif
lpos%=pos
back
until pos=1 and lpos%=1
position ref%
update
der%=der%-1
endwh
endif
beep 300,300
beep 200,800
beep 300,300
```

Il est important de noter que les algorithmes de tri donnés ci-dessus ne sont qu'indicatifs et que, selon les applications, d'autres peuvent s'avérer plus rapides et plus performants.



Course de voitures



Par
Alexandre
Bouillot

Voici un petit programme pour vous "défouler" sur votre XP ou votre CM avant de prendre la route pour rentrer de vacances. Si vous avez un LZ ou un LZ 64, deux possibilités s'offrent à vous : adapter le programme à un écran 4 lignes ou choisir XTRANS à la place de TRANS.

```
CARS:
LOCAL VIE%, TO%, LARG%, X%, AX%, DIR%, POS%, VI%, TMP%
LOCAL CARSS$(3), VI$(3), ROUTE$(16)
LOCAL PO
REM
GDU: (1, 14, 31, 31, 14, 14, 31, 31, 14)
AT 1, 2: PRINT "APPUYEZ SUR ON";
VIEW (1, "CONTROLE DU VEHICULE: FLECHE DROITE ET GAUCHE.
VITESSE: FLECHES HAUT ET BAS. ARRET DU JEU: TOUCHE
MODE.PAUSE: TOUCHE ON/CLEAR.")
DO
PO=0
VIE%=3
REM VIE% : NOMBRE DE VIES
CARSS$=CHR$(1)
LARG%=9
REM LARG% : LARGEUR DE LA ROUTE (EN CARACTERES)
AX%=8-LARG%/2
X%=8-LARG%/2
POS%=8
VI%=0
```

Suite du programme CARS

```
AX%=8-LARG%/2
X%=8-LARG%/2
POS%=8
VI%=0
ROUTE$="*" + REPT$(" ", LARG%-2) + "*"
CLS
PRINT REPT$(" ", 32);
DO
REM
REM TIRAGE DIR. ROUTE
REM
DIR%=RND*3
IF DIR%=0 AND X%<=16-LARG%
X%=X%+1
ELSEIF DIR%=2 AND X%>1
X%=X%-1
ENDIF
REM
REM AFFICHAGE ROUTE
REM
AT AX%, 1
PRINT ROUTE$
AT X%, 2
PRINT ROUTE$
REM
VI$="0" + FIX$(VI%+1, 0, 2)
AT 16, 2 : PRINT RIGHT$(VI$, 1)
AT 16, 1 : PRINT LEFT$(RIGHT$(VI$, 2), 1);
REM
REM AFFICHAGE VOITURE
REM
AT POS%, 2
PRINT CARSS$
REM SAISIE POS. VOITURE
TMP%=100
REM TMP% : VALEUR POUR LA BOUCLE D'ATTENTE
REM PLUS LA VALEUR EST GRANDE, PLUS LE JEU EST LENT
REM
WHILE TMP%>VI%*5
TMP%=TMP%-1
ENDWH
REM
TO%=KEY
IF TO%=5
POS%=POS%-1
ELSEIF TO%=6
POS%=POS%+1
ELSEIF TO%=4 AND VI%>0
VI%=VI%-1
ELSEIF TO%=3 AND VI%<9
VI%=VI%+1
ENDIF
REM
REM
AT POS%, 2
PRINT CARSS$
REM
```

suite du programme cars

```

REM AFFICHAGE VOITURE
REM
  AT POS %, 2
  PRINT CARS$
REM
REM COLLISION?
REM
  IF POS%<=X%-1 OR POS%>=X%-1+LARG%
    CLS
    PRINT "***** CRASH*****";
    GET
    VIE%=VIE%-1
    POS%=X%+LARG%/2
    CLS
    PRINT REPT$(" ",32);
  ENDIF
REM
REM FIN BOUCLE
REM
  AX%=X%
  PO=PO+1+V1%
  UNTIL VIE%=0 OR TO%=2
REM
CLS
PRINT "DISTANCE: ";PO
GET
DIR%=MENU("REJOUER,QUITTER")
UNTIL TO%=2 OR DIR%=2 OR DIR%=0

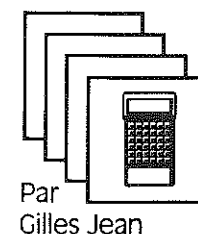
```

```

gdu : (x%, a%, b%, c%, d%, e%, f%,g%, h%)
rem
rem cette procédure permet de dessiner les caractères
rem nécessaire pour le bon déroulement de notre jeu.
rem
rem Pour adapter le jeux sur une machine LZ ou LZ64
rem utilisez la fonction UDG.
rem
POKEB $180, 64+x%*8
POKEB $181, a%
POKEB $181, b%
POKEB $181, c%
POKEB $181, d%
POKEB $181, e%
POKEB $181, f%
POKEB $181, g%
POKEB $181, h%

```

Bonne route à tous les amateurs Organiseur II.



Par
Gilles Jean

Une aide sur la base de données

Que ce soit avec les outils de développement sur PC et compatible ou avec un Organiseur II (tous modèles confondus), nous ne disposons pas de commande ou de fonction permettant de connaître le fichier en cours en mémoire à un moment M. En effet, la complexité de certaines applications nécessite cette information. L'information est stockée en mémoire à l'adresse \$b1 et retourne une information numérique. A correspond à 0, B à 1, C à 2 et D à 3 comme le montrent les petits programmes ci-dessous :

```

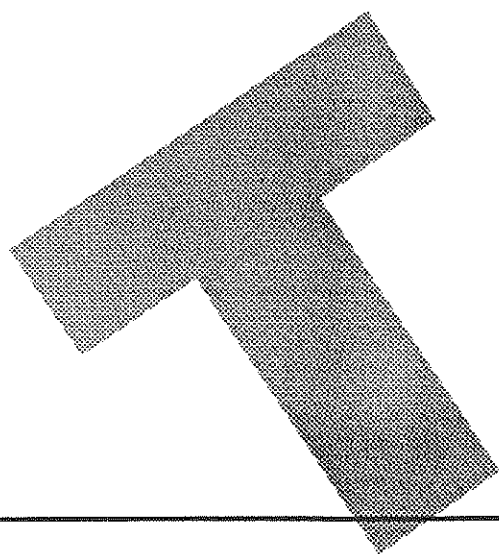
FILE:
trap create"a:xyz",a,a$
trap open"a:xyz",a,a$
print filbuf:
trap close
trap delete"a:xyz"
trap create"a:xyz",b,a$
trap open"a:xyz",b,a$
print filbuf:
trap close :get

```

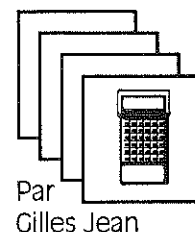
```

FILBUF:
return peekb($b1)

```



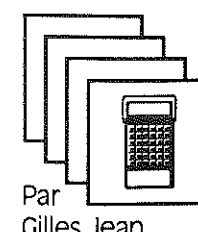
TTRN & RTRN



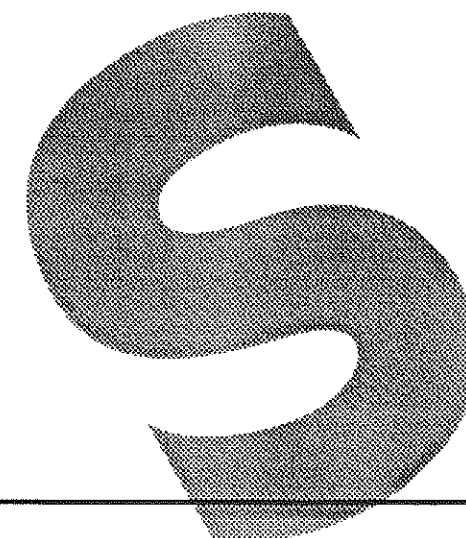
Par
Gilles Jean

16 Sachant que ces deux paramètres de la liaison Comms-Link (RS232) permettent la transcription d'un caractère en émission/réception avec une machine distante, on peut par exemple convertir une tabulation en une virgule. La tabulation étant le séparateur de champs (rubrique) d'un enregistrement, cela permettrait d'obtenir un fichier texte délimité par une virgule. Autre hypothèse de travail, nous pouvons convertir le séparateur de décimales qui est le point (.) sur l'Organiseur II en virgule (,). En effet, certains logiciels utilisent l'un ou l'autre comme séparateur de décimales. Ces options permettent aussi la suppression d'un caractère comme par exemple la possibilité de retirer des guillemets d'un fichier issue d'une base de données (Omnis 5, Dbase et d'autres) afin de le récupérer sur l'Organiseur II débarrassé de tout formatage. Vous pouvez, pour de plus amples renseignements, consulter votre manuel d'utilisation de Comms-Link. L'exemple ci-dessous permet de choisir son propre séparateur de champs pour exporter un fichier vers un PC/Mac.

```
EXP:
local a%(2),sep$(1)
a%(1)=$3f17 :a%(2)=$3900 :usr(addr(a%)),0)
print"Séparateur ?", :trap input sep$
if err :stop :else
lset: (9600,0,8,1,1,0,0,chr$(13)+chr$(10),chr$(26),"",chr$(13-)+chr$(10),chr$(26),
chr$(9)+sep$,0,2)
xtsend: ("a:main", "main", 0)
endif
```



Par
Gilles Jean



SHIFT EXE & SHIFT SPACE

17 Le clavier des Organiseur II LZ et LZ64 peut être étendu des combinaisons de touches suivantes :

SHIFT EXE : permet par exemple dans l'option de recherche du module de la base de données d'accéder à l'enregistrement précédant celui en cours en mémoire.

SHIFT SPACE : permet par exemple dans l'option de sauvegarde du module de la base de données de parvenir à la ligne suivante de celle en cours.

Il existe en effet une variable système (KBB_SPEC) unique à cette gamme de produit (cela ne fonctionnera donc pas sur un modèle deux lignes) permettant de détecter la pression simultanée de ces touches et d'en affecter une valeur ASCII récupérable dans vos applications. Cette variable ne peut prendre que quatre valeurs :

La valeur 1 active Shift EXE et retourne la valeur 14.

La valeur 2 active Shift SPACE et retourne la valeur 15.

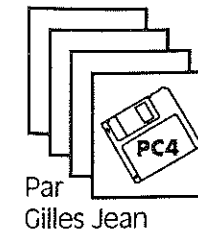
La valeur 7 active simultanément Shift EXE et Shift SPACE en retournant respectivement 14 et 15.

La valeur 0 retourne en mode standard.

Les autres valeurs étant réservées par le constructeur, nous vous conseillons beaucoup d'attention dans la rédaction de vos futures applications afin de ne pas encourir des phénomènes étranges.

```
TEST:
local k%
rem *****
rem Gestion du shift exe & Shift space *
rem *****
rem k%=buffer clavier utilisateur *
rem *****
escape off
rem *****
rem Shift Exe renvoie valeur Ascii =14 *
rem Exe renvoie valeur Ascii =13 *
rem *****
pokeb $2185,7
do
k%=get
if k%=1
rem *****
rem Sortie de la boucle de test *
rem *****
break
else
print k%
endif
until 0
rem *****
rem redispotion à la norme standard *
rem *****
pokeb $2185,0
```

```
TEST2:
local d%
rem *****
rem d%=buffer clavier utilisateur *
rem *****
pokeb $2185,7
rem *****
rem Ouverture du fichier *
rem *****
trap open"a:main",a,a$
if count
first
do :d%=disp(-1,"")
if d%=1
break
elseif d%=13
next
elseif d%=14
back
endif
if eof
first
endif
until 0
endif
rem *****
rem Fermeture du fichier *
rem *****
trap close
```



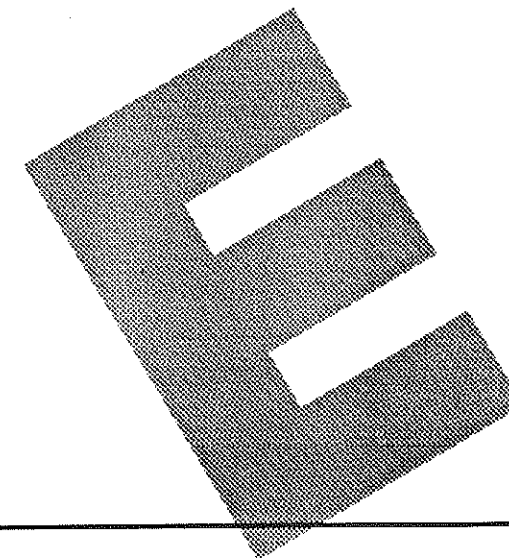
Par
Gilles Jean



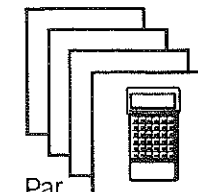
Une procédure d'entrée

Le programme listé ci-dessous vous permet de faire saisir à l'utilisateur un mot de passe (BOSS) et de contrôler la validité de la saisie. Si vous disposez d'un écran couleur, vous pourrez, en mixant les couleurs, masquer l'affichage du code en utilisant les commandes papier et encre.

```
proc start
local boucle,mp$
note ce programme fonctionne sur PC4 et XCHANGE *
mode 0 :éponge
écris car(218);repro(car(196),78);car(191);
écris car(179);repro(car(32),78);car(179);
écris car(192);repro(car(196),78);car(217);
écris au 20,0;repro(car(196),80);
écris au 21,0;repro(car(32),20);"Veuillez saisir le
mot de passe
- Merci"
écris au 8,32;car(218);repro(car(196),14);car(191)
écris au 9,32;car(179);" MOT DE PASSE ";car(179)
écris au 10,32;car(192);repro(car(196),14);car(217)
tantque boucle<3 :que boucle=boucle+1
écris au 12,32;car(27)+car(65);"===>";
saisis mp$
si majus(mp$)="BOSS"
note appel de votre procédure
stoppe
sinon :écris car(7) :fsi
ftantque
éponge :mode 1
fproc
```



Errare humanum est



Par
Gilles Jean

Dernières versions

MC

Système	1.29
Mobile	TR1.2
Mclink	1.29

ORGANISEUR

CM	3.3
XP	3.6
POS200	3.6
POS250	3.6
POS350	3.6
POS296	3.6

LZ	4.6
LZ64	4.6
POS 432	4.6
POS 464	4.6
Dévelop.	3.01
Diary link	1.0
CL PC	2.1
CL MAC	2.1

Log.2 lignes	
Tableur	2.0
Topfinance	1.3
Filepak	1.4
Barpak	0.9
Formulator	1.0
Travel Pak	1.5
Maths Pak	1.0

Log. 4 lignes	
Tableur	2.0
Topfinance	2.2
Filepak	2.0
Prakpak	1.0

PC4

PC4	1.1
-----	-----

XCHANGE

Xchange	1.37
---------	------

20

Deux procédures n'ont pu être imprimées dans l'article traitant de la compression de caractères du précédent Bulletin Technique (Avril/Mai/Juin 1990). Nous nous en excusons vivement auprès de tous nos lecteurs.

```
comp%: (char%)  
local newchar%  
if char%=9  
newchar%=60  
elseif char%>=32 and char%<=90  
newchar%=char%-31  
elseif char%>=97 and char%<=122  
newchar%=char%-63  
endif  
return newchar%
```

```
shift1% : (byte%,bits%)  
return (byte%*2**bits%)-256*int(byte%/((2**(8-bits%))))
```

Le contenu du présent document composé sur Macintosh II et QuarkXPress donne l'information la plus complète et la plus exacte possible au moment de la publication, mais n'est garanti ni quant à la complétude, ni quant à l'exactitude. **Awave, Omnis, Quartz, Xchange, Archive, Abacus, Quill, Easel, PC4, Organiseur II, Mobile computer, QuarkXPress, Apple II e, Macintosh, ImageWriter, LaserWriter, IBM** sont des marques déposées.
Réalisation : Odile Lecoq