

Exemples de logiciels SimH 20-Jun-2006

AVIS DE DROIT D'AUTEUR

L'avis de copyright suivant s'applique au code source, au code binaire et à la documentation du SIMH : Code

original publié en 1993-2006, écrit par Robert M Supnik.

Copyright (c) 1993-2006, Robert M Supnik

L'autorisation est accordée par la présente, gratuitement, à toute personne obtenant une copie de ce logiciel et des fichiers de documentation associés (le "Logiciel"), de traiter le Logiciel sans restriction, y compris sans limitation les droits d'utiliser, de copier, de modifier, de fusionner, de publier, de distribuer, d'accorder des sous-licences et/ou de vendre des copies du Logiciel, et d'autoriser les personnes à qui le Logiciel est fourni à faire de même, sous réserve des conditions suivantes :

L'avis de droit d'auteur ci-dessus et cet avis d'autorisation doivent être inclus dans toutes les copies ou parties substantielles du logiciel.

LE LOGICIEL EST FOURNI "TEL QUEL", SANS GARANTIE D'AUCUNE SORTE, EXPRESSE OU IMPLICITE, Y COMPRIS, MAIS SANS S'Y LIMITER, LES GARANTIES DE QUALITÉ MARCHANDE, D'ADÉQUATION À UN USAGE PARTICULIER ET D'ABSENCE DE CONTREFAÇON. EN AUCUN CAS, ROBERT M SUPNIK NE POURRA ÊTRE TENU RESPONSABLE D'UNE QUELCONQUE RÉCLAMATION, D'UN QUELCONQUE DOMMAGE OU D'UNE QUELCONQUE RESPONSABILITÉ, QUE CE SOIT DANS LE CADRE D'UNE ACTION CONTRACTUELLE, DÉLICTELLE OU AUTRE, DÉCOULANT DU LOGICIEL, DE SON UTILISATION OU D'AUTRES OPÉRATIONS LIÉES AU LOGICIEL.

Sauf dans les cas prévus par la présente notice, le nom de Robert M Supnik ne peut être utilisé à des fins publicitaires ou autres pour promouvoir la vente, l'utilisation ou d'autres opérations relatives à ce logiciel sans l'autorisation écrite préalable de Robert M Supnik.

1	PDP-8.....	3
1.1	ESI-X.....	3
1.2	FOCAL69	3
1.3	PDP-8 OS/8.....	3
1.4	PDP-8 TSS/8.....	4
2	PDP-11	5
2.1	UNIX V5, V6, V7.....	5
2.1.1	UNIX V5	5
2.1.2	UNIX V6	5
2.1.3	UNIX V7	5
2.2	RT-11	6
2.2.1	RT-11 V4.....	6
2.2.2	RT-11 V5.3.....	6
3	Nova et Eclipse RDOS.....	6
4	PDP-1.....	7
4.1	PDP-1 LISP.....	7
4.2	PDP-1 DDT.....	7
5	PDP-7.....	7
5.1	SIM8	7
5.2	DECsys	8
6	PDP-9/PDP-15	8
6.1	FOCAL	8
6.2	Système logiciel avancé/Moniteur de clavier.....	8
6.3	Système logiciel avancé/arrière-plan forestier.....	9
6.4	DOS-15	10
7	IBM 1401	11
7.1	Carte unique "Koans.....	11
7.2	Bande de diagnostic	11
7.3	SPS.....	12
8	HP2116 16K BASIC.....	13
9	PDP-10 TOPS-10 7.03, TOPS-10 7.04, TOPS-20 V4.1.....	14
10	Interdata 32b Systems	14
10.1	Interdata 7/32 UNIX.....	14
10.1.1	UNIX V6	14
10.1.2	UNIX V7	14

Ce mémorandum présente les exemples de progiciels disponibles pour fonctionner avec les simulateurs du SIMH. Un grand nombre de ces logiciels sont disponibles sous des licences d'utilisation limitée ; veuillez lire les conditions de licence incluses avec le logiciel.

1 PDP-8

1.1 ESI-X

ESI-X est un programme interactif de calcul technique. Il peut exécuter à la fois des commandes immédiates et des programmes stockés (comme BASIC). ESI-X est fourni à la fois sous forme de source et d'image papier au format chargeur binaire. Pour plus d'informations, voir la documentation fournie avec le programme. Mes remerciements à Dave Waks, qui a écrit le programme, et à Paul Pierce et Tim Litt, qui ont récupéré la source sur son support d'archivage.

Pour charger et exécuter ESI-X :

```
sim> load esix.bin sim>
run 5400
TYPE 2+2.
      2+2 4=
```

1.2 FOCAL69

FOCAL69 est un programme interactif pour les calculs techniques. Il peut exécuter à la fois des commandes immédiates et des programmes stockés (comme BASIC). FOCAL69 est fourni sous la forme d'une image papier au format chargeur binaire. Pour charger et exécuter FOCAL69 :

```
sim> load focal69.bin
sim> run 200
*TYPE 2+2
=      4.000*
```

1.3 PDP-8 OS/8

OS/8 est le système d'exploitation du PDP-8 basé sur la mémoire de masse. Il fournit un environnement de développement et d'exécution de programmes en assembleur, BASIC et FORTRAN. OS/8 est fourni sous licence, tel quel, sans frais, par Digital Equipment Corporation, pour un usage non commercial uniquement. Veuillez lire le contrat de licence ci-joint pour en connaître tous les termes et conditions. Ce contrat de licence doit être reproduit avec toute copie des images disques OS/8. Mes remerciements à Doug Jones de l'Université de l'Iowa, qui a fourni les images disque. et à Digital Equipment Corporation, qui a fourni la licence. Pour démarrer et exécuter

OS/8 :

```
sim> att rx0 os8_rx.dsk sim>
att rx1 os8f4_rx.dsk sim> boot
rx0

DA dd-mmm-yy
.
```

Notez que OS/8 ne reconnaît que les caractères majuscules. La première disquette (lecteur 0) est la disquette système ; elle contient également BASIC. La deuxième disquette (lecteur 1) contient FORTRAN.

1.4 PDP-8 TSS/8

TSS/8 est le système de temps partagé du PDP-8. Il fournit un environnement de développement et d'exécution de programmes en assembleur, BASIC et FORTRAN. TSS/8 est fourni tel , sans frais, par Digital Equipment Corporation, pour un usage non commercial uniquement. Mes remerciements à John Wilson de Dbit Inc, qui a fourni l'image du disque et la source de la bande d'initialisation. Note : votre environnement doit disposer d'un second Teletype en état de marche ; en d'autres termes, vous ne pouvez pas utiliser TSS si votre système hôte ne prend pas en charge la bibliothèque de sockets SIMH.

Pour charger et exécuter TSS/8 :

- Chargez la bande d'amorçage en papier :

```
sim> load tss8_init.bin
```

- Monter l'image disque TSS/8 du RF08 :

```
sim> attach rf tss8_rf.dsk
```

- Attribuez un port TCP/IP à l'auditeur Telnet pour les terminaux supplémentaires :

```
sim> attach ttix <port #> 4000 fonctionne typiquement
```

- Lancer le bootstrap :

```
sim> run 24200
```

- TSS/8 démarre et passe par la boîte de dialogue de démarrage.

```
CHARGEMENT, VIDAGE, DÉMARRAGE, ETC. DÉMARRAGE
MONTH-DAY-YEAR : mm:dd:yy          numérique, aa dans l'intervalle [74:85]
HR:MIN - hh:mm                     numérique, format 24 heures
(tapez cr pour attirer l'attention)
```

.

- et est maintenant prêt à être connecté. La liste des comptes et des mots de passe :

```
PPN      Mot de
passe [0,1]
        VH3M
[0,2]    LXHE
[0,3]    SHUG
[77,77]
[1,10]   WBCN
[20,1]   DT
[20,2]   PT
[20,3]   TSS8
[20,4]   EDIT
[20,5]   4EME
[1,50]   JERK
```

- Connectez-vous en utilisant l'un des comptes existants. La commande login n'est pas prise en compte :

```
LOGIN 2 LXHE                               compte de bibliothèque privilégié
```

```
TSS/8.24 JOB 01 [00,02] K00                23:23:06
SYSTEM IS DOWN, INC.
```

- Le système est maintenant prêt à recevoir des commandes. Pour obtenir la liste des répertoires :

```
R CAT
```

Les autres utilisateurs peuvent se connecter en se connectant, à partir d'un client Telnet, à localhost sur le port spécifié dans la commande `ATTACH TTIX`.

2 PDP-11

2.1 UNIX V5, V6, V7

UNIX a été développé pour la première fois sur le PDP-7 ; il a été utilisé pour la première fois à grande échelle sur le PDP-11. UNIX fournit un environnement de développement et d'exécution de programmes en assembleur et en C. UNIX V5, V6, V7 pour le PDP-11 est fourni sous licence, tel quel, sans frais, par Caldera Corporation, pour un usage non commercial uniquement. Veuillez lire l'accord de licence ci-joint pour en connaître les termes et conditions. Cette licence doit être reproduite avec toute copie des images disques UNIX V5, V6, V7. Mes remerciements à PUPS, la PDP-11 UNIX Preservation Society of Australia, qui a fourni les images disques, et à Caldera, qui a fourni la licence.

2.1.1 UNIX V5

UNIX V5 est contenu dans une seule image disque RK05. Pour démarrer UNIX :

```
sim> set cpu u18
sim> att rk0 unix_v5_rk.dsk sim>
boot rk
@unix login :
root #ls -l
```

2.1.2 UNIX V6

UNIX V6 est contenu dans quatre images disque RK05. Pour démarrer UNIX :

```
sim> set cpu u18
sim> att rk0 unix0_v6_rk.dsk
sim> att rk1 unix1_v6_rk.dsk
sim> att rk2 unix2_v6_rk.dsk
sim> att rk3 unix3_v6_rk.dsk
sim> boot rk0
@unix login :
root # ls -l
```

2.1.3 UNIX V7

UNIX V7 est contenu dans une seule image disque RL02. Pour démarrer UNIX :

```
sim> set cpu u18 sim>
set r10 RL02
sim> att r10 unix_v7_rl.dsk sim>
boot r10
```

```
@boot
New Boot, les appareils connus sont hp ht rk rl rp tm vt
: rl(0,0)rl2unix #
```

Une image plus petite est contenue dans une seule image disque RK05. Pour démarrer UNIX :

```
sim> set cpu u18
sim> att rk0 unix_v7_rk.dsk sim>
boot rk0
@boot
New Boot, les appareils connus sont hp ht rk rl rp tm vt
: rk(0,0)rkunix #
STTY -LCASE #
```

2.2 RT-11

RT-11 est le système d'exploitation mono-utilisateur du PDP-11. Il fournit un environnement de développement et d'exécution de programmes en assembleur, BASIC et FORTRAN. RT-11 est fourni sous licence, tel quel, sans frais, par Mentec Corporation, pour une utilisation non commerciale UNIQUEMENT SUR CE SIMULATEUR. Veuillez lire l'accord de licence ci-joint pour connaître l'ensemble des termes et conditions. Cet accord de licence doit être reproduit avec toute copie de l'image disque RT-11. Mes remerciements à John Wilson, un collectionneur privé, qui a fourni l'image disque de RT-11 V4 ; à Megan Gentry, de Digital Equipment Corporation, qui a fourni l'image disque de RT-11 V5.3 ; et à Mentec Corporation, qui a fourni la licence.

2.2.1 RT-11 V4

RT-11 est contenu dans une seule image disque RK05. Pour démarrer et exécuter RT-11 :

```
sim> att rk0 rtv4_rk.dsk sim>
boot rk0
```

Pour les disques des séries RL, HK, RM et RP, RT-11 s'attend à trouver une table de blocs défectueux du fabricant dans la dernière piste du disque. Par conséquent, l'initialisation d'un nouveau disque (tous les zéros) échoue, car il n'y a pas de table de blocs défectueux valide. Pour créer une table de blocs défectueux minimale, utilisez la commande `SET <unit> BADBLOCK`.

2.2.2 RT-11 V5.3

RT-11 est contenu dans une seule image disque RL02. Pour démarrer et exécuter RT-11 :

```
sim> set rl0 rl02
sim> att rl0 rtv53_rl.dsk sim>
boot rl0
```

Il s'agit d'un kit de distribution RT-11 complet. Il attend de l'utilisateur qu'il copie le pack de distribution et génère un nouveau système. Pour ce faire, il faut monter un pack vierge sur RL1. Lorsqu'un pack vierge est attaché au simulateur, une table de blocs défectueux doit être créée avec la commande `SET <unit> BADBLOCK`.

3 RDOS Nova et Eclipse

RDOS est le système d'exploitation de stockage de masse en temps réel de Nova. Il fournit un environnement de développement et d'exécution de programmes en assembleur, BASIC et FORTRAN. RDOS est fourni sous licence, tel quel, sans frais, par Data General Corporation, pour une utilisation non commerciale uniquement. Veuillez lire l'accord de licence ci-joint pour en connaître les termes et conditions. Cet accord de licence doit être reproduit avec toute copie de l'image disque RDOS. Mes remerciements à Carl Friend, un collectionneur privé, qui a fourni l'image disque, et à Data General Corporation, qui a fourni la licence.

Pour démarrer et exécuter RDOS pour le Nova :

```
sim> att dp0 rdos_d31.dsk sim>
set tti dasher
sim> boot dp0 FILENAME ?
(cr)
DATE (mm/dd/yy) ? xx/yy/zz
TIME (hh:mm:ss) ? hh:mm:ss R
liste/e
```

Pour démarrer et exécuter RDOS pour l'Eclipse :

```
sim> att dp0 zrdos75.dsk sim>
set tti dasher
sim> boot dp0 FILENAME ?
(cr)
DATE (mm/dd/yy) ? xx/yy/zz
TIME (hh:mm:ss) ? hh:mm:ss R
liste/e
```

4 PDP-1

4.1 PDP-1 LISP

PDP-1 LISP est un interprète interactif pour le langage Lisp. Il peut exécuter à la fois des commandes interactives et des programmes stockés. Les instructions de démarrage de LISP sont compliquées ; la documentation incluse dans le programme fournit des informations sur le chargement et l'utilisation de LISP. Je remercie Peter Deutsch, qui a écrit le programme, Gordon Greene, qui l'a tapé à partir d'une liste imprimée, et Paul McJones, qui a participé processus de débogage final.

4.2 PDP-1 DDT

PDP-1 DDT est un outil de débogage interactif pour le PDP-1. Il fournit des capacités de débogage symbolique pour les programmes PDP-1. La documentation incluse dans le programme fournit des informations sur le chargement et l'utilisation de DDT. Je remercie Derek Peschel, qui a transcrit et débogué DDT.

5 PDP-7

5.1 SIM8

PDP-7 SIM8 est un simulateur de PDP-8 pour le PDP-7. Il implémente un PDP-8/I 8K avec clavier, téléimprimeur, lecteur, perforateur et imprimante de ligne. Il fournit un environnement de console interactive pour le contrôle et le débogage du PDP-8.

PDP-8 simulé. Pour plus d'informations, voir la documentation incluse dans le programme. Mes remerciements à Dave Waks, qui a écrit le programme, et à Paul Pierce et Tim Litt, qui ont récupéré la source sur son support d'archivage.

Pour charger et exécuter SIM8 :

```
sim> load sim8.rim sim>
set tti fdx sim> run
AC/ 0000
```

5.2 DECSys

PDP-7 DECSys était le premier système d'exploitation de Digital Equipment Corporation basé sur la mémoire de masse. Conçu pour un PDP-7 de 8KW avec deux ou trois lecteurs DECTape, il offrait un environnement de développement interactif pour les programmes en Fortran et en langage d'assemblage. Je remercie le professeur Harlan Lefevre, de l'université de l'Oregon, dont la préservation minutieuse d'un PDP-7 en parfait état de marche, et de ses logiciels, pendant près de quarante ans, a rendu possible la récupération de ce logiciel.

Pour charger et exécuter DECSys :

```
sim> att -e dt2 decsys.dtp sim>
att dt3 scratch.dtp sim> load
decsys.rim 17640 sim> run
```

GA

6 PDP-9/PDP-15

6.1 FOCAL

FOCAL15 est un programme interactif pour les calculs techniques. Il peut exécuter à la fois des commandes immédiates et des programmes stockés (comme BASIC). FOCAL15 est fourni sous la forme d'une image papier au format chargeur binaire. Je remercie Al Kossow, qui a fourni l'image binaire. Pour charger et exécuter FOCAL15 :

```
sim> load focal15.bin
sim> run
*TYPE FSQT(2), !
=      1.4142
*
```

6.2 Système logiciel avancé / moniteur de clavier

Le moniteur clavier du système logiciel avancé est le moniteur de mémoire de masse le plus simple pour le PDP-15. Il offre des capacités de développement et d'exécution de programmes pour un seul utilisateur. Mes remerciements à David Gesswein, qui a fourni les images DECTape de ADSS/KM. Pour charger et exécuter ADSS/KM-15 :

- Sur le PDP-9 (uniquement), initialiser le mode extension sur on :

```
sim> d extm_init 1
```


- Chargez la bande d'amorçage en papier dans la mémoire supérieure :

```
sim> load dec-15u.rim 77637
```

Vous devez spécifier l'adresse de chargement.

- Vérifier que le bootstrap s'est correctement chargé :

```
sim> ex pc PC
:      077646
```

- Modifier l'imprimante de ligne par défaut pour qu'elle soit LP09 plutôt que Type 647 (PDP-9) ou LP15 (PDP-15) :

```
sim> set lpt disabled
sim> set lp9 enabled
```

- Monter l'image DECtape de l'Advanced Software System sur l'unité DECtape 0 :

```
sim> attach dt adss15_32k.dtp
```

- Lancer le bootstrap :

```
sim> run
```

- Le DECtape démarre et imprime

```
KMS9-15 V5B000
$
```

et est maintenant prêt à recevoir des ordres. Les ordres reconnus sont les suivants :

```
D      répertorier le répertoire des périphériques du système
I      liste des commandes disponibles
R      liste des affectations des appareils
SCOM   liste des systèmes de communication région
```

- Pour exécuter Focal, affectez l'emplacement DAT 10 inutilisé, puis chargez Focal :

```
$A LPA0 10
$GLOAD
CHARGEUR V5B000
>_FOCAL<altmode= control-[> FOCAL
V9A
*TYPE 2+2, !
4.0000
*
```

6.3 Système de logiciel avancé/contexte forestier

L'Advanced Software System/Foreground Background est un moniteur en temps réel pour le PDP-15. Il permet l'exécution de programmes en avant-plan et en arrière-plan. Mes remerciements à David Gesswein, qui a fourni les images DECtape de ADSS/FB.

Remarque : votre environnement doit disposer d'un second Teletype opérationnel ; en d'autres termes, vous ne pouvez pas exécuter Foreground/Background si votre système hôte ne prend pas en charge la bibliothèque de sockets SIMH.

Pour charger et exécuter ADSS/FB :

- Chargez la bande d'amorçage en papier dans la mémoire supérieure :

```
sim> load dec-15u.rim 77637
```

Vous *devez* spécifier l'adresse de chargement.

- Vérifier que le bootstrap s'est correctement chargé :

```
sim> ex pc PC
:      077646
```

- Monter l'image DECTape avant-plan/arrière-plan sur l'unité DECTape 0 :

```
sim> attach dt fb15_32k.dtp
```

- Attribuez un port TCP/IP à l'auditeur Telnet pour le deuxième terminal :

```
sim> assigner ttil <port #> 4000 fonctionne typiquement
```

- Lancez un client Telnet qui jouera le rôle de second terminal et se connectera à localhost sur le port spécifié.

- Lancer le bootstrap :

```
sim> run
```

- Le DECTape démarre et le Foreground/Background Monitor s'imprime sur le deuxième terminal.

```
F9/15 V4A
$
```

et est maintenant prêt à recevoir des ordres. Les ordres reconnus sont les suivants :

```
D 0      répertorier le répertoire des périphériques du système
R        liste des affectations des appareils
```

- Pour activer l'arrière-plan, chargez IDLE au premier plan :

```
$A DTA0 -4
$GLOAD FGLOAD
V2A
>_IDLE<altmode= control-[>
```

- Le terminal d'arrière-plan répond par

```
B9/15 V4A
$
```

et est maintenant prêt à recevoir des commandes.

6.4 DOS-15

DOS-15 est une version plus élaborée d'ADSS-15 utilisant le RF15 ou le RP15 comme périphérique système. Mes remerciements à Hans Pufal, qui a récupéré DOS-15 à partir d'une cassette DEC et a déduit comment reconstruire des images de disques à partir de cassettes DEC. Pour charger et exécuter DOS-15 :

- Charger le bootstrap de la bande papier DOS-15 dans la mémoire supérieure :

```
sim> load rfsboot.rim 77637
```

Vous *devez* spécifier l'adresse de chargement.

- Monter l'image disque DOS-15 RF sur le RF15 :

```
sim> att rf dosv2a_4p.rf
```

- Lancer le bootstrap :

```
sim> run
```

- Le disque RF démarre et imprime

```
DOS-15 V2A
SAISIR LA DATE (MM/JJ/AA) -
```

- Entrez la date (l'année doit être comprise entre 70 et 99). Le DOS-15 imprime son invite et est maintenant prêt à recevoir des commandes. Les commandes reconnues sont les suivantes :

```
I          informations sur le système
S          configuration du système
R          assignation des périphériques du système
```

7 IBM 1401

7.1 Carte unique "Koans"

L'une des formes d'art de l'IBM 1401 consistait à faire tenir des programmes utiles sur une simple carte perforée. Trois exemples sont inclus :

```
i1401_ctolp.cd      imprime un jeu de cartes sur l'imprimante à lignes
i1401_ctopu.cd      copie un jeu de cartes sur le perforateur de cartes
i1401_hello.cd      imprime "HELLO WORLD" sur l'imprimante de ligne et
s'arrête
```

Pour utiliser les cartes de reproduction, il suffit de les insérer au début d'un fichier texte, en les terminant par une nouvelle ligne. Attachez le fichier modifié au lecteur de cartes, attachez un fichier vierge au périphérique de sortie et démarrez le lecteur de cartes.

7.2 Bande de diagnostic

Le logiciel et le texte ont été fournis par Charles Owens.

Cette bande 1401 Diagnostics est une bande amorçable contenant une série de diagnostics 1401 datant d'environ 1962. La console 1407 Inquiry n'est pas utilisée ; toutes les commandes se font par le panneau avant.

Pour l'exécuter dans le simulateur, il faut l'attacher de la manière suivante :

```
sim> attach mt1 1401diag.mt sim>
attach lpt errorlist.txt sim> boot
mt1
```

Le simulateur s'arrête avec IS= 433. À ce , vous pouvez définir des options à l'aide des commutateurs de détection et de la mémoire.

```
D 1252 "1"          Permet d'imprimer les en-têtes pour chaque essai.
                    Dans le cas contraire, aucune impression n'aura lieu, sauf en
                    cas d'erreur.

D SSB 1             Boucle si une erreur est détectée.

D SSC 1             Imprime tous les cas de test et pas
                    seulement les erreurs. D SSD 1 Répéter l'exécution du test
                    encore et encore.

D SSE 1             Arrêter si une erreur est détectée, sinon continuer.
```

Lorsque vous continuez à partir de cet arrêt (utilisez C pour CON), le simulateur s'arrête à 3001. Entrez à nouveau C et la bande passera par une série de diagnostics de base de l'unité centrale.

7.3 SPS

Le logiciel et le texte ont été fournis par Charles Owens.

sps1.obj et sps2.obj sont les jeux de cartes objets du "Symbolic Programming System", un assembleur primitif pour le 1401, antérieur à l'Autocoder, plus connu et plus fonctionnel.

Pour utiliser SPS, écrivez un programme SPS à l'aide de votre éditeur préféré (deux exemples sont fournis, hello.sps et diaglist.sps). Les paquets SPS ne sont pas en format libre, mais les opérandes doivent être placés en colonnes :

```
1 - 5      Nombre de lignes (optionnel)
6 -        Count (nombre de caractères lors de la définition d'une
           constante).
8 - 13     Label (six caractères doivent commencer par un alphabet).
14 - 16    Opcode : Exemples :
           A= Add
           B= Branche (doit être d-mod pour conditionnel) BWZ
           = Branche si mot-symbole ou zone
           C= Comparer
           CC= Carriage Control (imprimante) CS =
           CLear Storage (mémoire)
           CU= Unité de contrôle (par ex.
           bande) CW = Clear Workmark (effacer
           le signet)
           D= Diviser
           DC= Définir la constante (pas de mot-symbole)
           DCW= Définir une constante (commence en 24, longueur en 6-7)
           END = Fin du programme
           LCA= Charger les
           caractères H = Halte
           M= Multiplier
           MCE= Déplacer et éditer
           MCS= Déplacer et supprimer les zéros
           MCW = Déplacer les caractères
           MN= Déplacement
           numérique MZ =
           Déplacement de zone
```

```

ORG= Définir le point
d'origine P = Carte perforée
R= Carte de lecture
S= Soustraire
SS= Sélection de
l'empileur SW= Définir
le mot-symbole W =
Ligne d'écriture
ZA= Zéro et addition
ZS= Bande de zéro et de
soustraction :
MCW %UX YYY W      Écriture d'une bande à partir de l'adresse YYY sans mots
repères)
LCA %UX YYY W      Écriture de la bande, unité X à partir de l'adresse
YYY avec mots-clés) MCW %UX YYY R Lecture de la bande à partir de l'adresse
YYY sans mots-clés)
LCA %UX YYY R      Lecture de la bande, unité X à partir de l'adresse
YYY avec marques de mots) CU %UX M      Écriture d'une marque sur la
bande
CU %UX E           Saut de bande et bande vierge
CU %UX B           Retour arrière de l'enregistrement
CU %UX R           Rembobinage de la bande
CU %UX U           Rembobiner et débobiner la bande
17 - 22           Adresse pour Opérande A (étiquette ou adresse réelle à 4
chiffres)
23 - 23           vierge,+ ou - pour ajuster l'opérande A d'une constante
24 - 26           Nombre à 3 chiffres pour ajuster l'opérande A si 23 est+ ou -
27 - 27           Index ( ?)
28 - 33           Adresse de l'opérande B
34 - 34           Blanc,+ ou -
35 - 37           Numéro à 3 chiffres de l'ajustement de l'adresse B si 34 est+ ou
-
38 - 38           Index ( ?)
39 - 39           Modificateur D pour cette instruction. Notes :
/= Comparer, c'est inégaliser
S= Branchement si comparaison égale
T= B est inférieur à A
U= B est supérieur à A L =
Erreur de lecture de la
bande
40 - 55           K= Fin de bobine de la bande
Commentaires

```

Le jeu de cartes SPS doit commencer par une opération ORG pour indiquer où commence le programme dans le stockage et se terminer par une carte END, avec un opérande A facultatif indiquant où commencer l'exécution.

Pour assembler un programme SPS, placez la source SPS entre les paquets SPS1.OBJ et SPS2.OBJ, et une autre copie de la même source après le paquet SPS2.OBJ (SPS est un assembleur à deux passes). SPS imprime un listing sur le LPT et perforé un jeu d'objets sur le CDP, prêt à être exécuté.

Un script de commande UNIX permettant d'assembler un jeu de cartes SPS sans difficulté se trouve dans "sps". Pour utiliser ce script, entrez "sps nomduprogramme". Le script crée programme.lst pour le listing et programme.obj pour le jeu d'objets. Les utilisateurs de Windows n'ont pas de chance, pour l'instant.

8 HP2116 16K BASIC

HP BASIC est une implémentation de BASIC centrée sur les bandes de papier pour un HP2116 de 16KW. Les numéros de périphériques correspondent aux paramètres par défaut du simulateur :

```

PTR    =      10
TTY    =      11
PTP    =      12

```

Le programme est un BASIC complet mais précoce et présente une exigence inhabituelle : tous les programmes doivent inclure une instruction END valide pour s'exécuter correctement. Je remercie Jeff Moffatt de m'avoir fourni ce programme.

Pour charger et exécuter BASIC :

```
sim> load basic1.abs sim>
run 100
PRÊT
10 PRINT SQR(2)
20 END RUN
1.41421
```

9 PDP-10 TOPS-10 7.03, TOPS-10 7.04, TOPS-20 V4.1

TOPS-10 était le principal système d'exploitation en temps partagé pour le PDP-10. TOPS-20 était une alternative populaire dérivée du système BBN TENEX. Les bandes d'installation et de distribution de TOPS-10 7.03, TOPS-10 7.04 et TOPS-20 4.1 sont disponibles à l'adresse <http://pdp-10.trailing-edge.com>.

10 Interdata 32b Systems

10.1 Interdata 7/32 UNIX

Interdata UNIX V6 a été le premier portage d'UNIX à partir de son environnement d'origine sur le PDP-11. Ce portage a été réalisé par Richard Miller et ses collègues de l'Université de Wollongong, en Australie, en 1976-77. UNIX V6 et V7 pour l'Interdata 7/32 est fourni sous licence, tel quel, sans frais, par Caldera Corporation, pour une utilisation non commerciale uniquement. Veuillez lire l'accord de licence ci-joint pour en connaître les termes et conditions. Cette licence doit être reproduite avec toute copie des images disques UNIX V6 et V7. Mes remerciements à Richard Miller, sans les conseils, les connaissances et la patience duquel la recreation d'Interdata UNIX n'aurait pas été possible, ainsi qu'à Caldera, qui a fourni la licence.

10.1.1 UNIX V6

UNIX V6 est contenu dans une seule image disque de 10 Mo. Pour démarrer UNIX V6 :

```
sim> set ttp ena sim>
set pas dev=12 sim> set
mt dev=85
sim> att -e dp0 iu6_dp0.dsk sim>
boot dp0
?login unix :
root # ls -l
```

10.1.2 UNIX V7

UNIX V7 est contenu dans deux images disque de 10 Mo. Pour démarrer UNIX V7 :

```
sim> set ttp ena sim>
set pas dev=12
```

```
sim> set mt dev=85
sim> att -e dp0 iu7_dp0.dsk
sim> att -e dp1 iu7_dp1.dsk
sim> boot dp0
Botte
: dsk(1,0)unix #
```

Tapez ^D pour activer le mode multiutilisateur. Le mot de passe root est root.