PostScript au Laboratoire

Pierre JUILLOT

C.R.N. Strasbourg

❄ ❄
❄

13 Décembre 1996

http://crnal4.in2p3.fr/~juillot

Adobe et PostScript

➪ 1982: Richard Warnock et Chuck Gerschke fondent “Adobe Systems Inc.”

➪ 1985: la Apple Laserwriter (première imprimante PostScript) et le langage PostScript résultat:
☞ du succès des Personal Computers, dont le Macintosh
☞ des premières imprimantes laser bon marché (Canon)
☞ de la baisse du prix des mémoires (256 kb)

➪ PostScript devient vite un standard pour la typographie industrielle
➪ puis, les grands constructeurs “adoptent” PostScript: Apple, Digital, IBM, HP, Sun, NexT . . . (bon gré, mal gré)

➪ Octobre 1996: plus de 300 périphériques d’impression contiennent le langage PostScript

➪ au laboratoire, les applications graphiques sont standardisées: X-Windows (X11) sur l’écran et PostScript pour l’impression

Qu'est-ce que le langage PostScript?

❶ PostScript est un **langage de description de page** (PDL); c'est une façon puissante de représenter le contenu d'une page imprimée: texte, graphique (traits, courbes), images digitisées (photographies) ...

❷ un **langage de programmation** de haut niveau.

➪ PostScript décrit une **page entière** de manière **globale** et non plus en tant que suite de lignes, comme dans les imprimantes classiques antérieures
un document PostScript est donc une suite de pages individuelles

➪ PostScript fait appel à des **piles** (comme le langage Forth) avec une notation postfixée (les arguments précèdent les opérateurs qui les utilisent)

➪ le calcul se fait sans les symboles classiques:
+ − * / =

➪ il est **interprété** (non compilé): un programme PostScript reste sous sa forme source jusqu'à son impression et est donc portable dans ∀ environnement
☞ PostScript est **indépendant** du **type** d'imprimante et de sa **résolution**
☞ les instructions sont exécutées immédiatement et ne demandent pas de grands buffers

Ce que n'est PAS PostScript

• un processeur de texte
• "quelque chose" après SCRIPT (SCRIPT/VS est un processeur de texted'IBM)
• un format conçu pour l'échange de documents

➪ **un fichier PostScript est un programme**

Qu'est-ce que un fichier PostScript? c'est un fichier ASCII qui contient un programme PostScript

➪ le software est **indépendant** du **type de matériel**: et de sa **résolution**: le **même fichier** peut être imprimé sur $\forall$ type d'imprimante, depuis une deskjet à 150 dpi jusqu'à une photocomposeuse (matériel de professionnel) à 2540 dpi.

➪ ce même fichier peut être visualisé sur un écran, à l'aide d'interprèteurs software (`ghostscript, ghostview`)

➪ il est destiné à la visualisation, à l'impression ou à être inclus dans un autre document PostScript (PostScript Encapsulé)

➪ PostScript n'intervient dans la vie d'un document qu'à son **étape ultime**; un fichier PostScript décrit un document **finalisé.** A partir d'un fichier PostScript seul, il est pratiquement **impossible** de revenir en arrière pour y changer des phrases, inverser des paragraphes . . .

➪ un fichier PostScript est indépendant de tout système d'exploitation

➪ il se transfére d'une machine à une autre par réseau (`ftp ascii`)

➪ on peut le lire et le modifier avec un éditeur

En tant que fichier ASCII, les lignes du fichier, terminées par <NL>, contiennent de mots (entités syntaxiques) séparés par des caractères espace ␣

Du point de vue PostScript,

➪ il n'y a **aucune contrainte** sur la longueur des lignes du fichier

➪ PostScript **impose** une seule convention: les deux premiers caractères du fichier **doivent** être: `%!`

(cf. `#!/bin/sh` au début des scripts en shell Unix)

➪ PostScript **recommande** l'inclusion de **métacommentaires** dans le fichier

Les attitudes possibles envers PostScript

➪ **l'utiliser de façon transparente:** envoyer un fichier en impression en espérant que tout ira sans problème

➪ **essayer de comprendre** pour pouvoir faire face à toute situation délicate
ex. comment interpréter le “dump” du contenu des piles que produisent les imprimantes modernes (ex: OCÉ-6730), en cas de problème d'impression?

➪ **apprendre les bases du langage** pour

☞ améliorer une application, ex.
- remplacer un texte anglais en français
- superposer du texte sur une image

☞ intervenir dans un fichier dont on ignore l'histoire, ex.
- épaissir les traits d'une image pour en faire un transparent
- enlever les couleurs d'une image pour l'inclure dans un article

➪ réaliser des sorties PostScript dans **ses propres programmes** ex. se passer des formulaires pré-imprimés en programmant le code PostScript qui les génère dans l'application même

Exemple de situation délicate

```
Message-ID: <B49F@cernvm.cern.ch>
From: templon@studbolt.mit.edu (Jeff Templon)
Newsgroups: cern.heplib,cern.printing
Subject: Problem Printing GARFIELD manual
Date: Fri, 19 May 95 15:18:47 GMT
Expires: Wed, 31 May 95
Sender: SERVUS@cernvm
Path: cernvm!servus
Keywords:
Lines: 31


Hi,

I picked up the postscript version of the GARFIELD manual from
the CN Division's web page, which ultimately led to

    http://consult.cern.ch/writeup/garfield

However if I try to print this, the laser printer chews on it
for quite some time, and then quits (or finishes) without
ever having printed anything.  I even tried viewing the
document under ghostview (which worked) and asking ghostview
to print selected pages for me, but the same thing happened
- five minutes thinking by the laser printer, but nothing ever
coming out.

Can someone please help?  Why don't you guys just use TeX/LaTeX
and dvips to make the documents, they work almost everywhere!

                        JT
```

Error Status

Product name: Oce 6730/6845 PS
Version 2011.23, Revision 7
Release 3.0, Firmware Level 22

Error name: typecheck
Offending operator: ifelse

Operand Stack	Execution Stack	Dictionary Stack
top	top	top
{	{	md
/pm	//def	userdict
save	}	shareddict
def	{	systemdict
mx3	//end	bottom
setmatrix	$error	
newpath	/_length	
0	//undef	
0	$error	
moveto	/init	
ct	//get	
dup	{	
39	handleerror	
get	}	
0	//if	
exch	$error	
getinterval	/dostop	
cvx	//get	
exec	{	
mtx	//stop	
astore	}	
setmatrix	//if	
mx1	}	
astore	{	
pop	//stop	
mx2	}	
astore	//@exec	
pop	~filestream~	
3a	{	
asto	//_disableinterrupts	
pnsh	}	
neg	//@stopped	
0	{	
rlineto	//_disableinterrupts	
pnlh	{	
pnlv	{	
Truncated Stack Listing	Truncated Stack Listing	

Program Continuation:
}{pnlv @1 gt
{@2 @1 moveto pnsh 0 rlineto pnlh pnsh add pnlv lineto 0 pnsv rlineto
pnsh neg 0 rlineto @2 @1 pnsv add lineto}{pnlh pnlv moveto pnsh 0 rlineto

Un “dump” de l’imprimante OCÉ-6730

Produire un fichier PostScript depuis une application Fortran

```
      PROGRAM POST

*   open metafile
      LUNPS=10
      OPEN(UNIT=LUNPS,FILE='post.ps'
     + ,FORM='FORMATTED',STATUS='UNKNOWN')

*   fill metafile
      WRITE(LUNPS,100,ERR=999)
  100 FORMAT('%!')
      GRAY=0.8
      WRITE(LUNPS,103,ERR=999)  GRAY
  103 FORMAT('clippath ',F3.1,' setgray fill showpage')

*   close metafile
      CLOSE(LUNPS)
      STOP

*   error
  999 PRINT * , 'Error in writing metafile'
      END
```

```
%!
clippath  .8 setgray fill showpage
```

Le même fichier PostScript depuis une application C

```
/*ooooooooooooooooooooooooooooooooooooooooooooo
 Une sortie Postscript depuis un programme C
ooooooooooooooooooooooooooooooooooooooooooooo*/
#include <stdio.h>
#define filename "post.ps"
#define gray 0.8

FILE *fp;

main()
{

/* open metafile */
fp=fopen(filename,"w");
if(!fp) {
      printf("Error in writing post.ps metafile\n");
           exit(1);
}

/* fill metafile */
fprintf(fp,"! \n");
fprintf(fp,"clippath %3.1f setgray fill showpage\n",gray);

/* close metafile */
fclose(fp);
}
```

```
%!
clippath  .8 setgray fill showpage
```

Les fichiers PostScript“sympathiques”(?!)
en tant qu’étape finale d’une application, on peut toujours recréer le fichier PostScript depuis cette application:

➩ **traitement de texte**: Word, LaTeX . . . ou **PAO** FrameMaker, XPress . . .

➩ **dessins**: Adobe Illustrator ou venant d’un scanner (Le transfert des fichiers Macintosh hors de leur monde s’accompagne de difficultés)

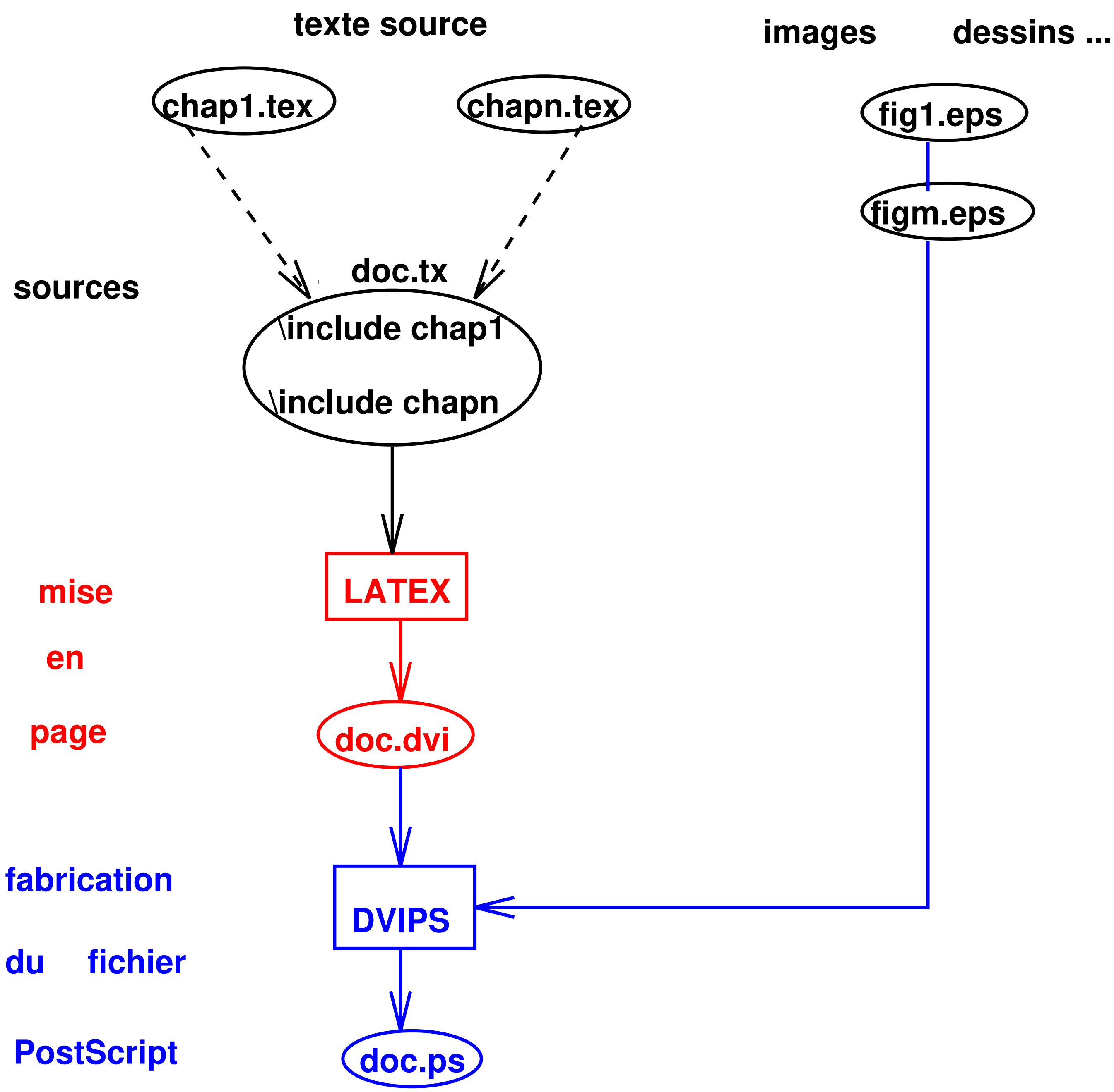
➩ **logiciels spécifiques:** CAO, EAO . . .
PAW pour le physicien: métafichiers PostScript (-111) ou PostScript Encapsulé (-113)

➩ **conversion** depuis d’autres formats:
ex. conversion d’un fichier texte en PostScript:
`lephp6 123: a2ps <post.c >toto.ps` ou
conversion d’une image GIF, TIF . . . `(xv)`

➩ **recopie d’écran**: `XtoPS, xwpick`

➩ résultat de **WWW**:
`netscape, mosaic`

Un cas sympathique: LATEX



Les fichiers PostScript que l'on ne produit pas soi-même

➩ fichiers échangés par **mail**

☞ comme le fichier est ASCII, il ne nécessite PAS de manipulation du genre `uuencode/uudecode`, `binhex/hexbin`, ...

☞ mais, se méfier des "initiatives" du mailer

• supprimer les lignes fabriquées jusqu'à la première ligne PostScript:

`%!` ou `%!PS-Adobe`

• le mailer IBM-VM introduit un ␣ en col. 1

•certains mailers tronquent à 72 col., . . .

➩ fichiers échangés **par réseau**:

le passage par réseau peut altérer le fichier, par ex. en tronquant les lignes

➩ fichiers obtenus depuis **un serveur de fichiers** PostScript (ex. CERN Preprint Server) ou depuis une machine qui accepte les connexions **ftp** (anonymous)

Si l'on n'est **pas satisfait** du résultat, il faut **intervenir dans le fichier PostScript lui-même**!

Que peut-on faire avec un fichier PostScript?

➪ **le visualiser** à l'écran

➪ **l'imprimer** soi-même sur papier, transparents, ou par un professionnel sur plaques de linotypes . . .

➪ **l'échanger par réseau** entre différentes machines

➪ **l'inclure,** en tant que dessin, figure, . . . dans un autre document (PostScript Encapsulé)

➪ **l'éditer et le modifier** dans la mesure du possible (ce cours)

Une imprimante PostScript est
❶ **un ordinateur** classique
sans écran, ni clavier
➪ **hardware**:
• un processeur, ex. RISC Intel i960 à 25 Mhz
• une mémoire centrale avec mémoire cache
(1 page A4 à 600 dpi = 600 dpi $\times$ 600 dpi $\times$ 10” $\times$ 8” / 8 = 3.6 MBytes)
• éventuellement, un disque interne
• plusieurs plusieurs protocoles de connexion: réseau, série (bidirectionnelle), parallèlle, Apple Talk

➪ **software**
• un interprèteur
• les “objets” PostScript: piles, dictionnaires
• une *Font machinery* (génératrice de $$$$)
mais pas de système d'exploitation, help, ...
$\Rightarrow$ un *Raster Image Processor (RIP)*

❷ **une imprimante laser** classique
tambour, dessin par laser, toner, cuisson ...

HP LaserJet 4MV Printer

Resolution	600 dpi with microfine toner
Resolution Technology	Resolution Enhancement technology
Print Speed	Up to 16 ppm
languages	**Adobe PostScript Level 2** and Enhanced HP PCL 5
Auto Language Switching	Yes
Memory Standard:	12 MB
Maximum:	44 MB
Memory Technology	Memory Enhancement technology
Media Capacity Standard:	350 sheets
	Optional: 850 sheets
	250 sheets output, face down
Media Sizes	Letter Legal Executive Tabloid (11 x 17) A3 A4 B3 B4
Media Types	Plain paper
	Envelopes
	Transparencies
	Labels
Scalable Typefaces	**35 Adobe Type 1**
	35 Intellifont
	10 TrueType (TM)
Interfaces	HP JetDirect card for Ethernet/LocalTalk
	HP Bi-Tronics parallel
Auto I/O Switching	Yes

Syntaxe: les objets du langage

Syntaxe	signification	examples
%	commentaire	% ignore ce qui suit
1.234	nombres	5 3.14 6.023e23
$< \ldots >$	nombre hexadécimal	$<$ 901fa3 $>$
(. . .)	chaîne de caractères	(X/STRA)
/any	nom littéral	/pi /abc /Abc
any	nom exécutable	pi abc Abc
{ . . . }	procédure	{add 2 div }
[. . .]	tableau	[12 /pi (X/STRA)]

%% métacommentaire:

%%BoundingBox: 0 0 623 842

Comment reconnaître un fichier PostScript?

➪ les deux premiers caractères sont `%!`

➪ un script pour reconnaître les fichiers PostScript

```
lephp9 97: more PS_or_not
#!/bin/ksh
#  check if a file is a PostScript file or not
#                          P. Juillot 11-06-1996
if [ $# -eq 0 ] ;  then
 echo  "Give file name to be printed :"
 read f
else
  f=$1
 fi
#    test existence of the file
if [ ! -f $f ] ; then
 echo  file $f does not exist!! ; exit 1
fi

#  is the file a PostScript file ($?=0) ot not ($?=1)?
head -n 1 $f | grep "^%!"
ret_code=`expr $¿  # return code of test i.e 0 or 1
zero=`expr 0`       # this is 0
if [ $ret_code = $zero ] ; then
   echo "file $f is a PostScript file"
else
   echo "file $f is NOT a PostScript file"
fi
```

➪ la première ligne *peut* contenir d'autres informations après les `%!` obligatoires:

`%!PS-Adobe-3.0` indique que le fichier PostScript contient des **métacommentaires**

`%!PS-Adobe-3.0 EPSF-`
indique un fichier PostScript Encapsulé

Mémoire PostScript et fichiers E/S

➪ la mémoire PostScript s'organise autour de **3 piles** de type LIFO

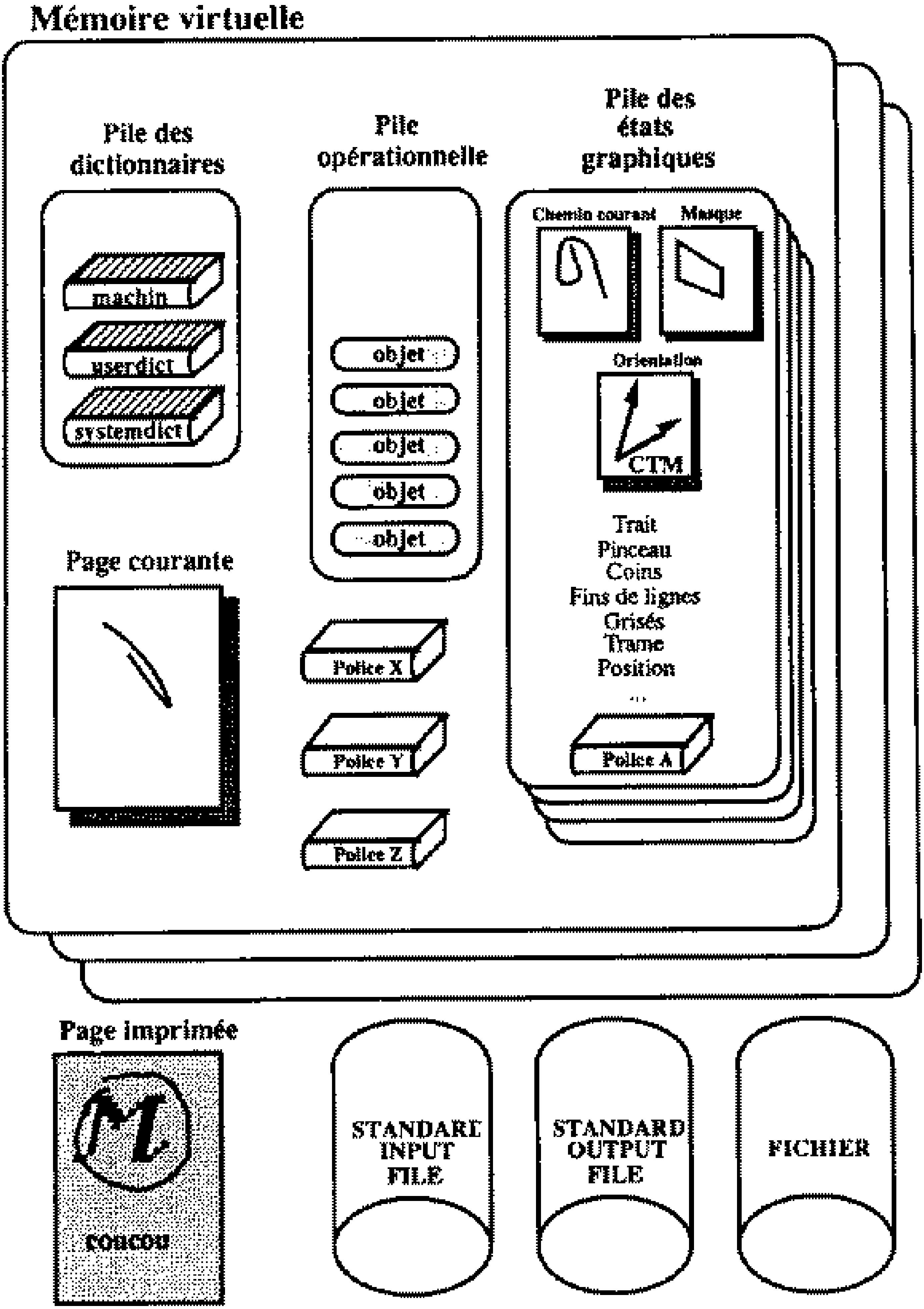
❶ **la pile opérationelle** (Operand stack) sert à l'interprètation du programme

❷ **la pile des dictionnaires** (Dictionary stack)

❸ **la pile des états graphiques** (une 4ème pile d'exécution (Execution stack) est utilisée de façon interne)

➪ **les fichiers d'entrée sortie:** (cf. UNIX et le C)

- **%stdin** (standart input file): le programme à interpréter
- **%stdout** (standart output file): retour des messages d'erreur, des prints de debugs,...



L'organisation de la mémoire

La pile opérationelle est la partie de la mémoire qui assure **l'interprétation** du programme; analogue à une pile de livres

➩ le programme est envoyé dans la pile et les **opérateurs** agissent sur les objets déjà présents dans la pile: ⇒le langage est post fixé

`stack` est un opérateur qui visualise le contenu de la pile rien y changer

ex. **empilement** et **addition** d'entiers:

```
GS>
GS>47 5 stack
5
47
GS<2>12 stack
12
5
47
GS<3>add stack
17
47
GS<2>
```

Les dictionnaires

➩ sont caractéristiques de PostScript et sont nécessaires pour sauvegarder une information: quand on “fait” a=2, on passe implicitement par un dictionnaire

➩ un dictionnaire est une suite de paires:
- le premier élément est le nom (littéral)
- le second est l’objet associé;

c’est donc une table où des noms (les clés) sont associés à des objets (leurs valeurs)

```
/a 2 def
```

➩ les **opérateurs** sont les commandes prédéfinies PostScript (environ 300 en Niveau 1 et 400 en Niveau 2); ils sont contenus dans le dictionnaire `systemdict`

➩ il y a toujours **deux dictionnaires**: `systemdict` et `statusdict` qui contient les opérateurs du “hardware”

➩ en redéfinissant un même opérateur dans des dictionnaires différents, on peut complètement modifier sa fonctionnalité (ex disabler l’opérateur `showpage`)

Que trouve-t-on principalement sur une page?

➪ **du texte** décrit par des chaînes de caractères dans une grande variété de polices de caractères, dans n’importe quelle position, orientation ou échelle

➪ **des figures géométriques** fabriquées généralement par un logiciel de dessin, sont construites avec des opérateurs graphiques, qui décrivent
-des lignes droites,
-des courbes de toute taille, orientation et épaisseur
-des régions pleines, de toute taille, forme et couleur

➪ **des images échantillonnées** de photographies numérisées, résultant de la digitisation par un scanner, à n’importe quelle échelle ou orientation

Les principes fondamentaux de PostScript

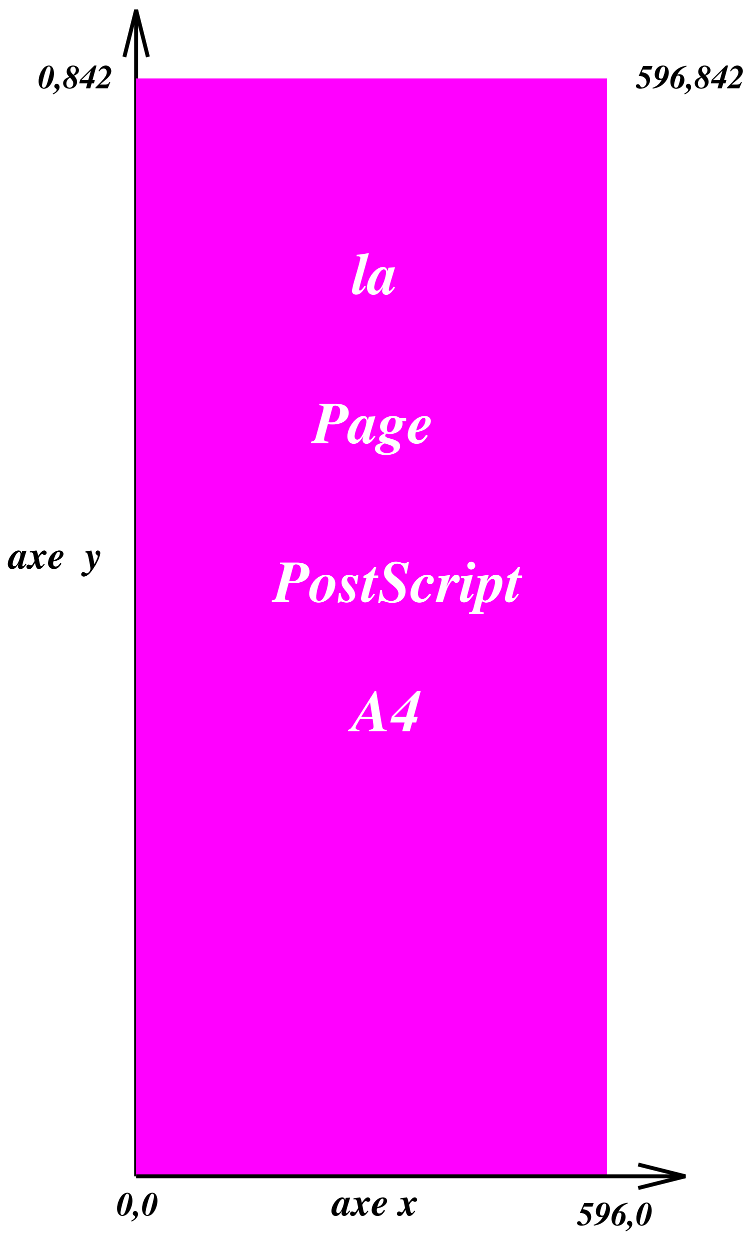
❶ la notion de **point courant**

toute entité graphique nécessite un point de départ et le déplace à la fin de son exécution: *“une tortue Logo levant et baissant la plume d’un traceur imaginaire”*

❷ **la superposition**

chaque entité graphique: ligne, surface, texte . . . se **superpose** à toutes celles déposées avant elle et en **masque** la partie qu’elle recouvre éventuellement

La page PostScript



** Système d'axes orthonormés:*

axe x: bord inférieur
axe y: bord gauche

**origine:*

coin bas gauche

**orientation des angles:*

mathématique

** unités PostScript:*

1 unité = 1/72 inch

**pas de limitation sur x et y:*

$-\infty$, $+\infty$

Texte

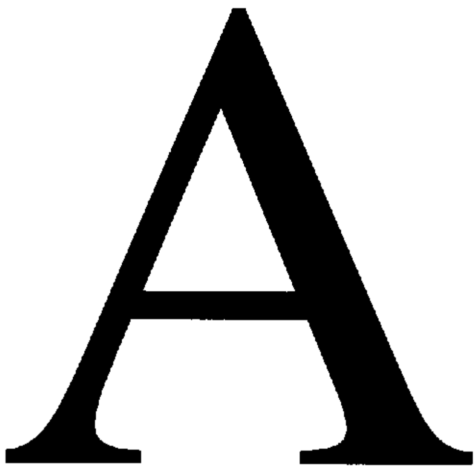
```
%!PS-Adobe-2.0
%%Title: file ex1.ps
%%CreationDate:Thu Nov 14 21:50:33 MET 1996
%%EndComments

% define orchid  as current color
.90  .49  .93 setrgbcolor

/NewCenturySchlbk-Roman findfont
100  scalefont
setfont

120 400 moveto
(X/STRA) show

showpage
```

X/STRA

Polices “bitmap” et Polices vectorielles

Police bitmap agrandie

➪ définie par un ensemble de pixels
➪ Type 3

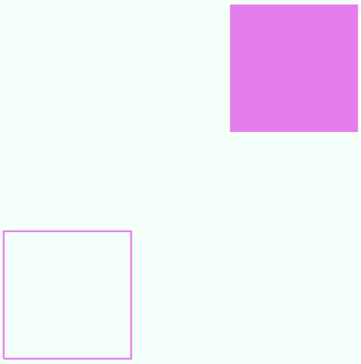


Police vectorielle

➪ les contours sont définis par des “calculs”:
Type 1: courbes de Bézier (3 pts) et **True Type**: splines (4 pts)

Chemins graphiques

```
%!PS-Adobe-2.0
%%Title: file ex2.ps
%%CreationDate:Thu Nov 14 22:10:27 MET 1996
%%EndComments

/MintCream {.96 1.00 .98 setrgbcolor} def
/orchid {.90 .49 .93 setrgbcolor} def

MintCream
clippath fill

orchid
newpath 72 72 moveto
  144 72  lineto 144 144 lineto 72 144 lineto
  72 72 lineto stroke

newpath 200 200  moveto
  272 200  lineto 272 272 lineto 200 272 lineto
  closepath fill
showpage
```



```
%!PS-Adobe-2.0
%%Title:carte_de_france.ps
%%Creator: f2ps
%%CreationDate: 2-AUG-1994 09:44:24
%%For: wolf@itesec (Christophe Wolfhugel) mod. P Ju
%%Pages: 1
%%BoundingBox: 33 121 579 671
%%EndComments
%%EndProlog
24.000000 674.000000 translate
0.900000 -0.900000 scale
1.000 setlinewidth
%
% Exemple de Polyline
%
newpath 341 4 moveto 311 12 lineto 303 17 lineto 304 51 lineto 308 57
lineto 303 56 lineto 302 62 lineto 311 71 lineto 302 64 lineto 299 72
lineto 288 83 lineto 263 90 lineto 240 103 lineto 236 115 lineto 237 120
lineto 252 118 lineto 223 132 lineto 215 128 lineto 191 122 lineto 182 124
lineto 181 130 lineto 178 119 lineto 174 113 lineto 176 106 lineto 175 104
 lineto 167 104 lineto 165 107 lineto 159 106 lineto 156 103 lineto 144 101
 lineto 146 107 lineto 147 108 lineto 147 116 lineto 149 123 lineto 158 135
 lineto 160 148 lineto 160 162 lineto 163 173 lineto 170 175 lineto 171 179
 lineto 166 179 lineto 149 180 lineto 147 172 lineto 142 184 lineto 137 178
 lineto 133 185 lineto 129 175 lineto 110 178 lineto 110 184 lineto 106 182
 lineto 91 156 lineto 72 157 lineto 67 168 lineto 56 165 lineto 55 172
 lineto 49 170 lineto 49 166 lineto 21 170 lineto 19 179 lineto 12 191
 lineto 19 190 lineto 31 186 lineto 27 190 lineto 34 190 lineto 37 198
lineto 32 196 lineto 25 194 lineto 21 192 lineto 20 194 lineto 18 195
 lineto 21 197 lineto 22 201 lineto 25 198 lineto 38 207 lineto 33 209
lineto 28 207 lineto 10 211
 lineto 10 212 lineto 18 215 lineto 27 221 lineto 27 227 lineto 28 231
 lineto 37 231 lineto 40 222 lineto 43 227 lineto 45 226 lineto 45 221
lineto 51 231 lineto 60 231 lineto 67 235 lineto 69 239 lineto 82 244
 lineto 85 253 lineto 87 253 lineto 85 246 lineto 90 248 lineto 90 244
lineto 93 244 lineto 94 246 lineto 103 243 lineto 103 248 lineto 95 249
 lineto 99 255 lineto 107 254 lineto 107 252 lineto 118 256 lineto 112 255
 lineto 114 260 lineto 111 262
 lineto 113 268 lineto 123 272 lineto 128 269 lineto 140 268 lineto 153 276
lineto 143 275 lineto 139 270 lineto 128 271 lineto 126 279 lineto 137 286
 lineto 128 298 lineto 144 317 lineto 146 325 lineto 162 336 lineto 177 337
lineto 177 341 lineto 172 347 lineto 182 357 lineto 179 361 lineto 177 370
 lineto 177 376 lineto 172 376 lineto 171 383 lineto 189 392 lineto 198 405
 lineto 202 426 lineto 207 428 lineto 210 436 lineto 206 431 lineto 206 449
 lineto 203 430 lineto 201 429
 lineto 192 406 lineto 180 391 lineto 176 395 lineto 170 455 lineto 175 448
 lineto 182 453 lineto 180 457 lineto 173 454 lineto 170 462 lineto 158 521
 lineto 148 536 lineto 139 544 lineto 139 547 lineto 160 556 lineto 155 567
lineto 159 570 lineto 165 564 lineto 176 571 lineto 192 574 lineto 207 587
 lineto 217 581 lineto 234 592 lineto 264 595 lineto 264 595 lineto 316 599
 lineto 337 615 lineto 347 609 lineto 367 614 lineto 387 606 lineto 400 608
 lineto 391 598 lineto 390 568
 lineto 400 545 lineto 412 547 lineto 440 528 lineto 444 534 lineto 466 535
 lineto 466 541 lineto 495 540 lineto 505 539 lineto 506 549 lineto 534 558
 lineto 571 549 lineto 573 532 lineto 582 532 lineto 602 506 lineto 614 499
 lineto 616 473 lineto 605 476 lineto
 582 467 lineto 578 447 lineto 588 439 lineto 584 430 lineto 568 427
 lineto 573 412 lineto 584 402 lineto 585 385 lineto 567 363 lineto 576 354
 lineto 565 336 lineto 567 326 lineto 556 316 lineto 542 318 lineto 536 324
 lineto 537 333 lineto 528 343 lineto 524 340 lineto 524 333 lineto 528 332
 lineto 531 324 lineto 528 322 lineto 528 308 lineto 544 284 lineto 567 254
 lineto 560 251 lineto 568 242 lineto 576 250 lineto 594 234 lineto 591 203
```

```
 lineto 596 165 lineto 614 135 lineto 574 124
 lineto 542 127 lineto 535 107 lineto 502 108 lineto 481 107 lineto 455 91
 lineto 450 66 lineto 443 66 lineto 438 78 lineto 421 77 lineto 420 58
lineto 412 55 lineto 398 56 lineto 393 46 lineto 379 43 lineto 371 24
 lineto 359 28 lineto 348 20 lineto 348 9 lineto 341 4 lineto stroke
showpage
```

La France vue de PostScript

Les dictionnaires toujours présents:

❶ `systemdict` contient les opérateurs du langage PostScript, indépendants de l'imprimante

❷ `systemdict` contient `FontDirectory`, le dictionnaire des fontes présentes dans l'imprimante (35)

❸ `statusdict` contient les opérateurs relatifs au contrôle et au pilotage de l'imprimante: bacs A4 / A3, recto/verso, connexion, copies multiples, mode manuel ...

❹ `serverdict` pour des opérations "permanentes"

➩ un fichier `PPD`, fourni par Adobe, contient "le PostScript" spécifique à chaque type de matériel

PostScript et PostScript Encapsulé

❶ un fichier PostScript Encapsulé contient la description d' **une seule page**, qui contient généralement **une illustration** c.à.d. histogramme, diagramme, dessin, image, ...

❷ l'illustration a son bord bas gauche qui commence en (0,0)

➩ un fichier PostScript Encapsulé est destiné à être physiquement inclus dans un autre document PostScript

➩ la syntaxe d'un fichier PostScript Encapsulé est rigoureusement identique à celle d'un fichier PostScript, **à quelques exceptions près**, destinées à ne pas déranger l'environnement dans lequel il s'insère:

☞ la syntaxe recommandée pour sa première ligne est

`%!PS-Adobe-3.0 EPSF-`

☞ **il doit** contenir un métacommentaire de type

`%%BoundingBox`

☞ il peut ne **pas** contenir de `showpage`

☞ il ne doit pas contenir d'opérateurs **d'initialisation globale**, comme `initgraphics`, `erasepage ...`

Et si on n’a pas d’imprimante PostScript?

➪ ex. lecture d’un fichier PostScript

opérateur `run`

```
lephp6 55: gs
Initializing... done.
GS>
GS>(latex.ps) run
Loading Helvetica-Bold font from /usr/local/lib/ghostscript/font
>>showpage, press <return> to continue<<
```

➪ liste des “drivers” contenus dans `ghostscript`

opérateur `devicenames`

```
crnal4.in2p3.fr> gs
Initializing... done.
GS>
GS>devicenames ==
[/x11 /bj10e /bj200 /cdeskjet /cdjcolor /cdjmono /cdj500
/cdj550 /declj250 /deskjet /dfaxhigh /dfaxlow /djet500
/djet500c /epson /eps9high /epsonc /ibmpro /jetp3852
/laserjet /la50 /la75 /lbp8 /ln03 /lj250 /ljet2p /ljet3
/ljetplus /m8510 /necp6 /oki182 /gif8 /gifmono /pbm
/pbmraw /pgm /pgmraw /ppm /ppmraw /bit]
GS>
```

➪ impression sur une imprimante non PostScript dont le “driver” existe :

```
crnhp2 82:
crnhp2 82: more gsdj
#!/bin/sh
# to print PostScript files using ghostscript on a
# deskjet 500 (dj) or the printer specified via the
# environment variable LPOPTS
exec ghostscript -q -dNOPAUSE -sPAPERSIZE=a4 -sDEVICE=djet500 \
     -sOutputFile="|lp -ddj -or $LPOPTS " $* quit.ps
crnhp2 83:
```

L'état graphique et la pile des états graphiques

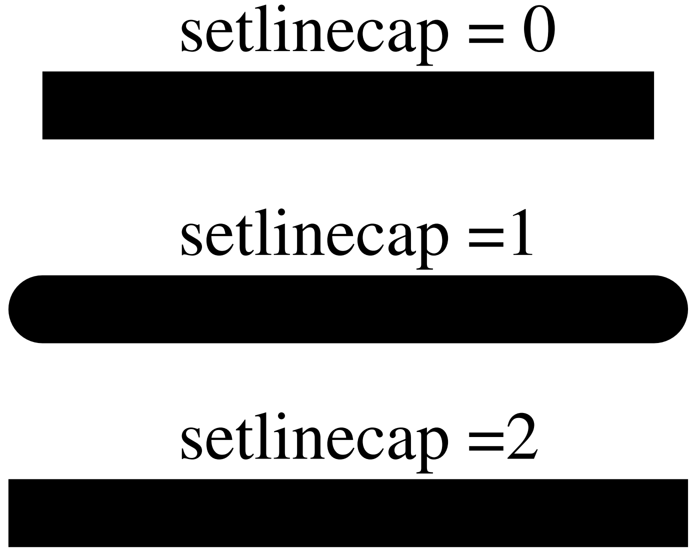
l'état graphiquecontient des quantités courantes: chemin graphique, la valeur du gris `setgray` couleur, fonte, épaisseur de ligne, les différents pointillés, la façon dont les lignes se terminent, se joignent ...

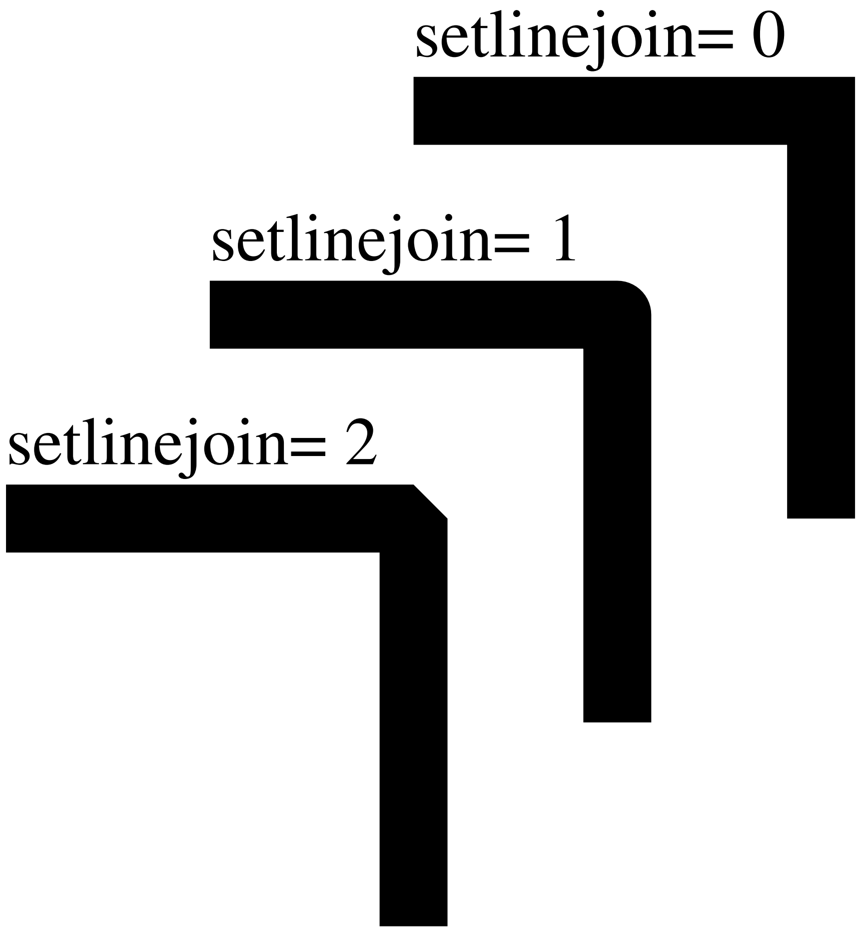
➪ dès que la valeur d'une des quantités de l'état graphique varie, on a affaire à un nouvel état graphique

➪ PostScript permet de définir un nombre illimité d'états graphiques par **translation, rotation, changement d'échelle**

➪ chaque état graphique peut être sauvé dans **la pile des états graphiques** (par l'opérateur `gsave`)

➪ on restaure un état graphique, en dépilant les états graphiques au-dessus de lui dans la pile des états graphiques, par l'opérateur `grestore`


setlinecap = 0
setlinecap =1
setlinecap =2


setlinejoin= 0
setlinejoin= 1
setlinejoin= 2

X/STRA

Le “clipping”

```
%!PS-Adobe-2.0
%%Title: ex4.ps
%%CreationDate: Dec  9 20:36:12 MET 1996
%%EndComments
/cm { 72 mul 2.54 div } def
/fill_fond  {  % expects 7 arguments
% this procedure fill_fond  fills an area full of a text;
%    1-4: the limits of the area, i.e.
%       llx, lly, urx, ury as for a standard BoundingBox
%    5: the name of the font
%    6: the size of the font
%    7: the string with which to fill
%  I can pile (but I prefer to define it here) a small quantity
%      quantity delta to separate the string both in x and y
  /delta 0.1 cm def
   /ury exch def /urx exch def /lly exch def /llx exch def
   /ps exch def  /fname exch def /str_fond exch def
%    set the font and compute the string length in this font
    fname findfont ps scalefont setfont
   /str_len str_fond stringwidth pop def
%   compute the size of area in in x and y: Xtot, Ytot
   /Xtot urx llx sub def /Ytot ury lly sub def
%   how many strings in x? Xtot/string length +1
   /nx Xtot str_len div cvi 1 add def
%  how many strings in y? Ytot/ ps +1
   /ny Ytot ps div cvi 1 add def
% looping  on x and y, the origins will contain delta
/ycurr lly def          % y starts at lly
1 1 ny {
/xcurr llx def          % x starts at llx
1 1 nx {
% go to origin, print string, increment x-origin in the x-loop
xcurr ycurr moveto str_fond show /xcurr xcurr str_len add delta
% increment y-origin in the y-loop
/ycurr ycurr ps add delta add def
 } for } def
gsave 0.5 setgray
(Groupe Strasbourgeois des Utilisateurs d’UNIX: X/STRA)
    /Helvetica 10 0 0 612 772          fill_fond
grestore showpage
```

X/STRA

X/STRA