xdb : eXtra DéBogueur symbolique

mise en route :

$cc -g *nomfich.c*	$cc -g *nomfich.c* -o *objet*
$xdb [*-options*]	$xdb [*-options*] *objet*

avec pour principales options : -R restitution état enregistré par **ss** (voir ci-après)
-P capture d'un processus en cours

les principales commandes (en bref) (pour plus de détails voir page suivante) :

aide :	aide en ligne de xdb (voir aussi le **manuel** de xdb)	h		help
afficher :	variable ou expression, adresse	p *var*		p *var?*
	précédente, suivante	p -		p +
	liste paramètres	l		
	tracer pile d'appels fonctions	t		T
fenêtre source :	voir	v		V
	déplacement	+		-
	déplacement suivant pile d'appels	up	down	top
	recherche avant, arrière	/		?
	répète dernière recherche	n		N
	toggle case sensitivity (MAJ/min)	tc		
points d'arrêts :	ajouter un point d'arrêt	b		
	ajouter en début procédure, toutes	bb		bp
	ajouter en fin procédure, toutes	bx		bpx
	tracer appels procédure, toutes	bt		bpt
	liste points d'arrêts	lb		
	détruire un point d'arrêt, tous	db [n^o]		db *
	détruire tous ceux associés aux procédures		dp	
	tous les basculer (actif/inactif)	tb		
	sauve état dans un fichier (récupération avec option -R)	ss		
exécuter :	(run) avec, sans arguments	r		R
	(step) pas à pas **entrant** dans fonctions	s		
	(step) pas à pas **S**ans entrer dans fonctions	S		
	continue	c		C
	goto ligne	g		
	(kill) tuer processus lancé par xdb	k		

quitter le débogueur pour terminer : q

affichage suivant format explicitement indiqué :

p *variable* \[*n*\]*f*[*t*]	affichage d'une variable
p + \[*n*\]*f*[*t*] **p -** \[*n*\]*f*[*t*]	affichage de l'élément suivant/précédent (tableau)
p *expression* *f*[*t*]	affichage d'une expression.
p *variable*? \[*n*\]*f*[*t*]	affichage d'une adresse

avec :

n = nombre d' objets (dans le cas d'un tableau)

t = taille de chaque objet

b	1 octet (byte)
s	2 octets (short)
l	4 octets (long)
D	8 octets (double)
L	16 octets (long double)

(par défaut : taille appropriée au type de la variable)

f = format d'affichage

a	ascii jusqu'à octet nul (chaîne)
s	string (chaîne)
c,b	caractère, (byte) octet décimal
d,D	décimal (entier, entier long)
x,X	hexadécimal (entier, entier long)
e,E	exposanté (float, double)
f,F	fixé (float, double)
g,G	général (float, double)

commandes d' observation des fichiers sources :

v [*n°ligne*]	affiche fichier source autour de la ligne spécifiée. [défaut : ligne courante]
V [*profondeur*]	affiche fichier source autour de la ligne courante en remontant dans la pile d'appels des fonctions jusqu'à la profondeur indiquée. [défaut : 0]
top	affiche fichier source au niveau du sommet de la pile d'appels.
up [*nb*] **down** [*nb*]	remonte ou descend d e *nb* niveaux dans la pile d'appels. [défaut : 1]
+ [*nb*] **-** [*nb*]	remonte ou descend de *nb* lignes dans le fichier source. [défaut : 1]
/ [*chaîne*] **?** [*chaîne*]	cherche en avant ou en arrière la *chaîne* dans le fichier source.
n **N**	répète la dernière recherche (en changeant de direction dans le 2è cas).

points d'arrêts :

b [*n°ligne*] [*nbre*] [*commandes*] — place un point d'arrêt au *n°ligne* indiqué. Le point d'arrêt sera ignoré *nbre* fois avant l'arrêt. Les *commandes* seront exécutées au moment de l'arrêt.

bb / **bx** / **bt** [*prof*] [*nbre*] [*commandes*] — place point d'arrêt { en début / en fin / pour tracer } procédure en remontant à la profondeur *prof* dans pile d'appels.

bp / **bpx** / **bpt** [*commandes*] — idem pour <u>toutes</u> les procédures.

commandes d'exécution contrôlée du programme :

t	trace de la pile d'appels de fonctions : indique les appels successifs de fonctions (celles en cours d'exécution).
r [*arguments*]	exécute (**r**un) le programme avec les arguments indiqués. [défaut : les arguments précédemment utilisés]
R	exécute (**r**un) le programme sans arguments.
s [*nb*]	exécute *nb* lignes (**s**tep by step) [défaut : **1**]
S [*nb*]	--- idem ---; mais ne s'arrête pas dans les fonctions appelées.
c [*n°ligne*]	**c**ontinue exécution après un arrêt (point d'arrêt ou signal interruption). Si c'est un signal qui a été la cause de l'arrêt, il est remis à zéro (<u>non reçu</u>). Eventuellement un point d'arrêt temporaire est placé au *n°ligne* indiqué.
C [*n°ligne*]	--- idem ---; mais si c'est un signal d'interruption qui a provoqué l'arrêt, il sera reçu par le programme en cours de déboguage.
g [*n°ligne*]	lance l'exécution (**g**o) à la ligne indiquée.
k	tue (kill) le processus (programme en cours de déboguage) lancé par **xdb** lors d'une commande **r**, **R**, ou **g**.