

#####

#####: 43126
2000 ## .
© 2000, 2001, 2002, 2003 ## ##### ## ##### ## #####
#####

#####

#####

Version française de Frédéric Praca <frederic.praca@freebsd-fr.org>.

#,.#.: ## ##### ##### ## ##### ## ## ##### ##### ## ##
#####.

#,.#.: ##### ## ##### ## ##### ## ##### #####—
<freebsd-questions@FreeBSD-fr.org> ## ##### ##### # ##
#####.

#####. ##### ## ##### ## ##### ##
#####:

- ## ##### ##### ## ##### ##### ## ##### ## #####.
- ## ##### ##### ## ##### ##### ## # #####.
- ##### ##### ## ##### ## ##### ## ##### ## ##### ## —
#.#.

#####

(## #####) ## ##### ##### (##, ##, ##, #####—
#####, ## ## ## #####) ##### ## ##### #####, ## ##### ##### ## #####—
##:

1. ##### ## ##### ## (## #####) ##### ##### ## ##### #####, #####
#####.
2. ##### ## ##### ## (##### ## ##### ##, ##### ## ##, #####
#####) ##### ##### ## ##### #####, ##### ## #####
##/## ##### ##### ## ##
#####.



#####

#####, #####, ## ## —

##

##, ### ##### ##### ## ##### ## ##
#####. ## ## ##### ## ##### ##—
#####, #####, #####,
#####, #####, ## ##### (#####, ## ## ##—
##, ##### ## ##### ## ## #####; ### ## ##, ##,
#####; ## #####) ##### ## ## ##
#####, ##### ## #####, #####, ## ## (#####
#####) ##### ## ## ## ## ## ##—
#####, ### ## ##### ## ## #####.

#####

#. #####	1
1. #####	5
2. ## #####	7
3. ##### ##	9
4. ##### ## /###/###	11
##. ### #####	13
5. ##### ##	17
5.1. #####	17
5.2. #####	17
5.3. ##### # ##	18
5.4. ##### ## cc	21
5.5. ###	29
5.6. #####	34
5.7. ##### #####	39
5.8. ##### #####	49
6. #####	51
6.1. #####	51
6.2. ##### ##	51
6.3. ##### ##	51
6.4. ### ##### #	55
6.5. #####	55
6.6. ##	57
6.7. ###	57
###. ##	59
7. ##### ##	63
8. ##### ##	65
8.1. ###	65
8.2. ### ##### (####)	68
8.3. #####	68
##. #####	69
9. ##	73
#. ##### #/ (#####/#####)	75
10. ###	79
##. ##### (###)	81
11. ###	85
###. ##	87
12. ###	91
###. #####	93
13. ###	97
##. #####	99
14. #####	103
#. ##	105
15. ###	109
##. #####	111

16. #####	115
16.1. #####	115
16.2. #####	115
16.3. #####	117
16.4. #####	117
16.5. #####	121
17. #####	123
17.1. #####	123
17.2. #####	127
18. ##### (###)	131
18.1. #####	131
19. #####	133
19.1. #####	133
20. #####	135
###. #####	137
21. ###32	141
22. #####	143
23. ###64	145
####. #####	147
24. #####	151
###. #####	153
25. #####	157
##. #####	159
#####	163

#####

8.1. ##### ## ##### 66

8.2. #####(9) ### ## ##### 68

#####

5.1. ## ##### ##### .emacs 42

#.

#####

1. ##### ### #####	5
2. ## ##### ###	7
3. ##### ## #####	9
4. ##### ## /###/###	11

1.

#####

#####—
####, ## ##### ## ##, ## ##### ## #####, ##### ## /###/##,
#####, ##.

! #####
##.

2. ## #####

3. #####

#####

4. ##### ## /
###/###

#####.
/usr/src ## ##### ## #####—
.

#####	#####
bin/	##### ## /bin
contrib/	##### ## ##### ## ##### ##### (#####).
crypto/	##### ## ##
etc/	##### ## /etc
games/	##### ## /usr/games
gnu/	##### ##### ##### ##
include/	##### ## /usr/include
kerberosIV/	##### ## ##### ##
kerberos5/	##### ## ##### 5
lib/	##### ## /usr/lib
libexec/	##### ## /usr/libexec
release/	##### ##### ## ##### ##### ##### ## #####
sbin/	##### ## /sbin
secure/	##### ##
share/	##### ## /sbin
sys/	##### ##
tools/	##### ## ##### ## ## ##### ##
usr.bin/	##### ## /usr/bin
usr.sbin/	##### ## /usr/sbin

##. ###

#####

5. #####	17
5.1. #####	17
5.2. #####	17
5.3. ##### # ##	18
5.4. ##### cc	21
5.5. #####	29
5.6. #####	34
5.7. ##### #####	39
5.8. #####	49
6. #####	51
6.1. #####	51
6.2. #####	51
6.3. #####	51
6.4. ##### #	55
6.5. #####	55
6.6. ##	57
6.7. #####	57

5.3. ##### # ##

##; ##### # ##, ##### ##### # # ##### # ##
#####. #####
##—
#####, ## # ##### # ##### # ## #####. ##
interpréteur, ##### # *compilateur*. ##### # ##
#####, ##

##.

5.3.1. ##

#####, ## ##### # ## # ## ##### # ## # ##—
##. ##
##, ##### # ## # ## # ## # ##

#####. ## # ## # ## # ## # ##, #####
##.

##—
#####, ## # ## # ## # ## # ## # ## # ##. ## # ## # ##—
##—
#####. ## # ## # ## # ## # ## # ## # ## # ##
##, ## # ## # ## # ## # ## # ##. ## # ##, ##

##.

##, ## # ##### # ##### # ## # ## # ## # ##
#####. ## # ## # ##### # ## # #####
##, # #####, #### # ##. ## # ## # ## # ##
(sh, csh) ## # ##### # ## # ##, ## # ##—
(#####) #####
##. ## # ##,

##.

5.3.2. ## ##### # ##### #

##—
##, ##### # ## # ## # ## # ## # ##
#####.

5. ##### ##

#####, #### ## ## ## ## ## ##
#####

pkg_add nom du logiciel

#####. #####, #### ##### #### ##### 2.1.0
!

#####

(###
#####). ##### ## ##
50 #### ##### ## ##### ## #####
80, ##### # ## ## ##—
#####. ## ## #####
#####.

(##—
#####) #### ##### ##### ##### ## #####—
#####

####

50 ##### ## #####—

#####. ##### ##### ## ## ## #####, #### ## ## ## ## ## ##;
#. #### ##### ##
(#####).

, #### #####
#####.

#####.

####

#####; ##### ##### ## ## #####
##.

(##### 5) ## ##### #####.

#####

####. ##### ## ## ##### ##### ##
, ## ##### ##
#####.

#####

##.

####

#####.

####

#####.

#####

#####

5.3.3. ###

#####. #### #####, #### #####
(## ##) ##### ## ##### ## #####.
#####.
##, ##### ## ##### ## #####; #### ##### ##
#####, ##### ##### ## ##### # ##### ## #####
#####¹.

, ## ##### ## ##### ## ##### ## #####. #####—

#####, ##### ##### ## ##### ## #####

! ##### ##### ## ##### ##### ##### ## #####
#####, ##### ##### ## ## ##### ##### ##### ##### ## #####
####, ## ## ## ##### ## ##### ##### ## #####. ## #####

#####.

#####, # ## +-+. # ## +-+ ##### ## ##### —
#####; #####,
####, # ## ##### ## ##### #####, ## ## ## #####
#####. #####, ##### ## ##### ##### #####, —
#####.

—
#####, ##### ## #####
(## ##
#####).
##; ##### ##

¹## ##### # #####, #### ##### ## ##### ##### (##:
#####).


```
#####, ## ##### ## ### ##### ##### #####, ##### ##### ### cc ### ##
##### ### ##### ##### ##### ##### ##### ##### ##### #####
##### ##; ##### #####
```

```
% cc foobar.c
```

```
##### foobar.c ##### ##### ##### #####. ## ##### ##### #####
# #####, ##### ##### ##### #####
```

```
% cc foo.c bar.c
```

```
##### ## ## ##### ## ##### ##### ##### #####. ##### ##
##### ## ## ##### ## ##### ##### ##### #####, ##### #####
## ##### ## ##### ## ##### ## ## # ##### ##### #####
##### ## ## #####.3
```

```
## # # ##### ##### cc, ## ## ## ##### ##### ##### ##
##### ## #####. ##### ##### ##### ## #####, ##### ## #####
##### #####
```

```
-o nom_du_fichier
```

```
## ## ## ##### ## #####. ## ##### ##### #####, cc #####
## ##### a.out.4
```

```
% cc foobar.c          l'exécutable est a.out
% cc -o foobar foobar.c l'exécutable est foobar
```

```
-c
```

```
##### ##### ## #####, ## ## ##. ##### ##### #####
##### ##### ##### ##### ## #####, ## ## ##### ## Makefile.
```

```
% cc -c foobar.c
```

```
#### ## ##### ## fichier objet (### ## #####) ##### foobar.o. #####
#### ##### ##### ##### ##### ##### ##### ##### ##### ## ##-
#####.
```

```
-g
```

```
#### ## ##### ## #####. ##### ## #####-
#### # ##### ##### ##### ##### ##### ##### ## #-
##### ##### ##### # ## #####. ## #####
```

³## ## ## ## ## ##### ##, ## ## ##### ## ## ##### ##### ## ##### ##

⁴## ##### ## ##### ##### ##### ## #####.

5. ##### ##

```
### ##### ##### ##### ## ##### ## ## ## ##### ##
##### ## # ## ##### ## #####, ## ## ## très #####; ## #####
### ## ##### ## ##### ##### ##### ## ##### ##
#### . #####, #### ##### ##### ##### -g ##### ## ## ##
##### ## ##### ## ##### ##### ## # ##### ## ##### #
#### -g ##### ## ##### ## #####.
```

```
% cc -g foobar.c
```

```
#### ## ##### ## ##### ## ##### ## ##### #####. 5
```

-O

```
#### ## ##### ##### ## #####. ## ##### ##### ##-
##### ##### ##### ##### ## ##### ## ##### ##
#### ##### ## #####. #### ##### ## ##### ##### -
O #### ##### ## ##### ##### #####, #### #####
##### ##### ## ##### ##### ##### ## #####. ## #####, ##
##### ## cc ##### ## ## ##### 2.1.0 ##### ## #####
## ##### ## ##### -O2 #### #####.
```

```
##### ## ##### ##### ##### ## ## ## ##-
#### ##### ## #####.
```

```
% cc -O -o foobar foobar.c
```

```
#### ## ##### ## ##### ## foobar.
```

```
### ##### ##### ##### cc # ##### ## ##### ##
##### ## ##### ##### ## #####, ## ##### ##
## ## ## ##### ## ## ##### ##.
```

-Wall

```
##### ## ## ##### ## ## ##### ## cc ##### ## ##.
##### ## ##, ## ##### ## ## ##### cc ##.
```

-ansi

```
##### ## #####, #### ## #####, ## ##### ## #
### cc ## ## #####. ##### ## ##, #### ## ##### ##
##### ## ## ##### ## #####.
```

-pedantic

```
##### toutes ## ##### ## cc ## ## ## ## .
```

⁵#####, #### ##### ## ##### ##### -o ##### ## ## ## #####, ##### ##-
a.out. ##### ## ##### ## ## ## foobar ## ##-
#####!

#####, cc #### ##### ##### ##### ##### ## ##—
 #####. ##### ## ## ##### ##### ## ## ##—
 ##### ## ## ##### ##### ##### ## ##, ## ## #####
 ## ##### ## ## ##### ## ## ##### ## ## ## #####
 ##### ## ## ##### ## ##### ## #####. ## ## ## ##
 ## ## *code portable*.

#####, ## ## ## ## ## ## ## ## ## ## ##—
 #####, ##### ## ## ##### # ##### ##### #####
 ##### ## ## ##### ##### ##### ## ## ## ##
 ##### #####?

```
% cc --Wall --ansi --pedantic -o foobar foobar.c
```

foobar ##### ##### ## foobar.c ## #####
 #####.

-librairie

#####.

 ## ##### ##### ##### ## #. # ##### ## ##
 ## #####, ##### ## ##### ## ## ##### ##
 ##### ## ## #####.

libquelque_chose.a, ## ## ##
 cc ##### -lquelque_chose. ## #####, ## ##### ## #####—
 ##### libm.a, ##### ## cc ## ##### -lm. ## # ##### # ##—
 ##### ## ## ##### ## ## ##### ## ## ##
 ## #####.

```
% cc -o foobar foobar.c -lm
```

foobar.

#++; ##### ##### -lg++, ## -lstdc++ ## #####
 ##### 2.2 ## #####, # ## ## ## cc #####—
 ##### ## ## ##### #++. #####, ##### c++ #####
 ## cc, ## ## ## ## ## ##. c++ ##### ## ## g++ ##
 #####.

```
% cc -o foobar foobar.cc -lg++ Pour FreeBSD 2.1.6 et antérieur
% cc -o foobar foobar.cc -lstdc++ Pour FreeBSD 2.2 et ultérieur
% c++ -o foobar foobar.cc
```

```
##### ## ### ##### ## ##### ## ##### foobar # ##### ## #####
##### #++ foobar.cc. ##### ##, ## ## ##### ##, ## ##### ## #++
## ##### ##### ## .C, .cxx ## .cc, ##### ## ## #####
.cpp (### ##### ##### ##### #####). gcc # ##### ## ## #####
## ## ## ##### # ##### ## ## ##### #####; #####, #####
##### ## ##### ##, ##### ## ##### ##### #####
##### #++ .cpp ## ##### ##### !
```

```
# : ##### ## ##### ## ##### sin() ## ## #####
#####. ### ##### ?
```

```
# : ##### ## ##### ##### ##### sin(), #####  
##### # cc ## ##### ## ##### ##, ##### :
```

```
# : ##### ## ##### ##### ##### # ##### ## #####
-lm. ##### ## ##### ## ##### 2,1 # ## ##### 6.
```

```
int main() {
    float f;

    f = pow(2.1, 6);
    printf("2.1 ^ 6 = %f\n", f);
    return 0;
}
```

```
% cc temp.c --lm
```

```
% ./a.out
2.1 ^ 6 = 1023.000000
```

25

: ##### ## ##### ##### ## ##### #####, ## ##### ##
#####. ##### ## ## ##, ## ##### ## ##
##, ## ## ##### ##### ## ## ## ##
##.

: ##### ##### ## #####?

: ##### ##### ##### ##### ##### math.h. ## #####
#####, ## ##### ##### ##### ## ##### ## ## --
#####!

```
#include <math.h>
#include <stdio.h>

int main() {
...
}
```

#####, ##### :

```
% ./a.out
2.1 ^ 6 = 85.766121
```

##, #####
toujours math.h ## ##### ## ## ##### ##.

: ##### ## ##### foobar.c ## ## ## ##### ## --
foobar. ## #####?

: #####, cc ##### a.out ##### ## ## ## ##
#####. ##### -o *nomfichier*:

```
% cc --o foobar foobar.c
```

: ##, ##### ## ##### foobar, ## ## ## ## ls, ##
foobar # #####, ## ##### ## ##
#####. ##### ## ## ##?

: # #####, ##### ## ##### ## ##
##, ##
##. ##### ## ## ./foobar, ## ## --
foobar ##### ## ##### #, ##
PATH ## ## ## ##
#####

```
bin:/usr/bin:/usr/local/bin:
```

5. ##### ##

#.

: ##### ##### test, ##### ## ## ##### ##### ## #####.
#####?

: ## ##### ### ##### ##### ## ## ##### ##### test ##### /usr/bin ##
##-
#####. ##### #####

```
% ./test
```

!!

: ##### ##### ## ## ##### ##### ## ##### #####, #####
#####. ###
#####?

: ## ### *core dump* ##### ##### #####, ##### ## #####
#####⁶##### #####. #####, ##
#####, ## ##### ## ##### ##
core,

#####.

: #####, ##### ## ##### ##### ?

: ##### gdb ##### ## ##### core (##### [5.6](#), # ##### #).

: ##### ## ##### # ##### ## ##### core, ## # ##### ##### ##-
#####. ##### ## ##### ?

: ##### ##### ##### ## ##### ##### # ##### #####
#####; ##### ## ##### #####
#####.

:

- ##### ##### ## ##### ##, ## #####

```
char *foo = NULL;  
strcpy(foo, "-bang!");
```

- ##### ## ##### ## ## ## ##, ## #####

⁶###: ## ##### ## ##### ## ##### #####

```
char *foo;
strcpy(foo, "-bang!");
```

#####, ###—
 ##### ## ## ## ## ## ##### ## ##### ## ##### ## ##
 ##### ## ## ##### ## ##### ##### ##### ## ## ##
 #####. ## ##### #####, ## ##### ##### ## ##
 ##### ##### ## ##### ## ## ##### ## #####, #####
 ##### ##### #####.

- ##### ##### ## ##### ## ## ## ## ##, ##

```
int bar[20];
bar[27] = 6;
```

- ##### ## ##### ##### ## ## ## ## ##, ##
 #####

```
char *foo = "-Ma chaine";
strcpy(foo, "-bang!");
```

"Ma chaine" ###
 ### ##### ## ##### ##.

- ##### ## ##### ## ## malloc() ## free(), ##

```
char bar[80];
free(bar);
```

##

```
char *foo = malloc(27);
free(foo);
free(foo);
```

#####, #####
 ##### ##### ## ##### #####. ##### ## #####
 ##### ##### ## #####, ## ## ##### ## #####
 ##### ## ## ## ##### ##### ## ## ##
 ## ## ##.

5. ##### ##

: ### #### ##### ## ##### ## ##### ##### ## ##, ## ## ##### ## #####.

#####. ##### ## ?

: ###, ##### ## (#### ## ##### ## ##### ##### ## #####
#####...). #### ## ##### ## ##### ## ##### ## ##-
##.

: ##### ##### ##### ## ##### core ##### ##### ##, ## ## ## ##
#####. ##### ##### ##, ## ##### ##-
?

: ###, ##### ## ##### ##### ## #####,

```
% ps
```

##, ##

```
% kill --ABRT identifiant
```

identifiant ## ##### ## ##### ## #####.

##, ##
#####. ## ##### ##### # ##### ## ##### ##, ## #
#####.

#####, ##### ##### ## ##### core ##### #####
abort(). #### ## ##### ## ##### ## **####(3)**
#####.

core ##### ##### ## ##-
#####, ##### ## ##### ## ## ## #####, #####
gcore. #### ## ##### ## ##### ## **####(1)** #####
#####.

5.5.

5.5.1. ##### ## **make**?

#####, #####

```
% cc fichier1.c fichier2.c
```


5.

```
foo: foo.c
cc -o foo foo.c
```

,

(##### cible),

make #####; #####
foo # #####
foo.c. # foo ##### foo.c, #

foo.c #

(#####) #####
foo # #####
foo #####, make #####

foo.c.

make, # foo # #####
foo.c. # ##### # Makefile# # #--
, # #
make world ##### #--
!

#####--

Makefile
#####:

```
foo: foo.c
cc -o foo foo.c
```

```
install:
cp foo -/home/moi
```

make

% make cible

make #####
make foo ##### Makefile # , make

make, make #####

make #####, make

#####

```
## ##### ## ## ##### ##### # ##### ## #####. ## ##
## ##### ## ## ##### make #####, ## ##### ## ##
##### ## ##### ##### # ##### ## # ## ## ##### Makefile #####
## ##### ## ## #####. ##### ##### ## ##### ## ##
##### ## ## ##### ## ##### ##### ## ##### #####. ## #####
touch ## ## ##### ##### ## ## ## ## ##### ## ## #####.
```

```
##### Makefile# ##### ##### ##### # #####. #####, ## #####
### ##### ##### ##### ## ##### ##### ##### #####
##### ## #####. ## ##### ## ##### ## ##### #####.
##### ## ##### ##### Makefile ##### ## #####:
```

```
#####, ## ##### ##### ##### ## ##### ## ## ##### ##### ## #####
make, ## ##### ##### ## ##### :
```

- ⁸###: ## ##### ##### ##### ## patch.

5. ##### ##

```
##### ## #####
## ##### ## ##### ## ##### ## #####
## ##### ## ##### ## ##### ## ##### ## #####).

6. ## ##### ## ##### ## #####. ## #####, ## ##### ##
##### ## ##### ## ## ##### # ## ##### ## #####
make### ##### Makefile ## ##### ## #####
#### ##### ## #####.

7. ##### ##### ## ##### ## #####. ## ##### ## ----
####, ##### ## ##### #####; ##### ## ##### ## ##
#####, ##### ##### make install. ##### ## ##### ##
## ##### ## ##### ##### ## ## #####; ## ##### ## #####
##### ## ##### ## ##### ## ##### #####, ##### ## #####
##### ## ##### ##### ##### ## ##### ## #####
##### ## #####.
```

#####
!

```
## ##### ##### ## ##### ## ##### # make ## ##### ## Makefile
##### ##### bsd.port.mk. ## ##### ## ##### ## ##### ##
##### ## ##### ## ##### ## ##### ## ##### ## Makefile ## ##
# make ## ##### ## ##### # # ##### (#### ## ##### ##### ##
#### ## #####, ##### ## ##### ## #####) ## ##### ## #####
##### # ##### ## ##### ## ##### ## ##### ## #####
Makefile !

## ##### ##### ## ##### ## ##### Makefile# #####, ## ## ##### /usr/share/
mk ## ## ## ##### ## ##### ## ##### ## ##### ##
## ## ##### ## ##### Makefile# ## ##### ## ##### ## (## ##
#### ## #####, ##### ## ##### ## ##### ## ##### ## ##### !)
```

5.5.4. ##### ## make

```
Make ## ## ##### ## ##### ## ##### ## ##### ## #####
##### ## ## #####. #####, ## # ## ##### ## make
## ##### #####. ## ##### ## #####
##### ## ##### ## ##### ## ##### ## ##### ## ----
##### ## ##### ## ##### ## ##### ## ##### ## #####.
```

```
## ##### ## make ##### ## ##### ## ## ##### ## (#### ## #####);
## # ## ##### ## ##### ## ##### ## /usr/share/doc/psd/12.make. #### ## ##,
#####
```

% zmore paper.ascii.gz

#####

#####.

gmak.
#####.

```
#####  
##### /usr/local/info #####  
#####
```

* **Make:** (make). l'utilitaire GNU Make.

```
## #####. ### ##### ##### ##### #####, ##### ##### info ## #####
##### ##### ## ##### (## #####, ##### C-hi).
```

5.6. #####

5.6.1. ##

```
## #####
##### ## #####
```

```
% gdb nomprog
```

```
##### ## ##### ## ##### ## ##### ## #####. #####  
##### ##:
```

M-x gdb RET nomprog RET

[illegible][illegible]

```
#####, ## ##### ##### ##### ##### #####, ## # ## #
##### ##### #####. #####. ##### ##### ##### #####.
```

```
##### # ##### # ##### # ##### # ##### # gdb ## ##
##### # ##### ##### ##### # ##### # #####.
```

5.6.2. ##### ## ##### ##### ## #####

```
#####
##### _g #####
##### gDb. #####
#####
#####. #####
#####:
```

... (no debugging symbols found) ...

```
##### gdb #####, #####  
-g-
```

```
##### gdb, ##### break main. #####  
#####  
#####  
##### run #####  
#####  
##### main(). (#####  
##### main() #####, ##### !).
```


n. ##
s. ###
f. ####
up ## down #####.

```
##### ## ##### ##### ## ##### ##### ##### ##### ## ##### #####
gdb. ##### ##### (#### ## ##### #####):
```

```
#include <stdio.h>

int buzz(int anint);

main() {
    int i;

    printf("C'est mon programme\n");
    buzz(i);
    return 0;
}

int buzz(int anint) {
    printf("Vous m'avez fourni %d\n", anint);
    return anint;
}
```

```
## ##### ## # 5 ## ## ##### # ### ##### bazz() ### ##### ## ##### ##  
##### ## ##### #####.
```

```
##### ## ##### ## ## #####-
#####
```

```
#### ##### ## ##### ## #####
```

```
% cc -g -o temp temp.c
% ./temp
C'est mon programme
Vous m'avez fourni 4231
```

```
## ##### ## ## ## ##### ! ## ## ##### ## ## ## ## ## !
```

```
% gdb temp
GDB is free software and you are welcome to distribute copies of it
under certain conditions; type -"show copying" to see the conditions.
There is absolutely no warranty for GDB; type -"show warranty" for details.
GDB 4.13 (i386-unknown-freebsd), Copyright 1994 Free Software Foundation, Inc.
(gdb) break main  passe le code d'initialisation
Breakpoint 1 at 0x160f: file temp.c, line 9. gdb met un point d'arrêt sur main()
(gdb) run  Exécute jusqu'à main()
Starting program: -/home/james/tmp/temp  Le programme démarre
```

```
Breakpoint 1, main () at temp.c:9 gdb s'arrête à main()
(gdb) n  Va à la ligne suivante
C'est mon programme  Le programme écrit
(gdb) s  entre dans bazz()
bazz (anint=4231) at temp.c:17 gdb montre la pile
(gdb)
```

```
##### ## #####! ##### ## ## ## 4231? ## ##### ##
## # 5 ##### main()? ##### ## main() ## #####.
```

```
(gdb) up  Remonte la pile des appels
#1 0x1625 in main () at temp.c:11 gdb montre la pile
(gdb) pi  Montre la valeur de i
$1 = 4231 gdb montre 4231
```

```
## ! ## ##### ## ## ##, ## ## ## ## ## ## ## ## ## ##
## #####
```

```
...
main() {
  int i;

  i = 5;
  printf("C'est mon programme\n");
  ...
}
```

```
##### ##### ## ## ## i=5;. ##### ##### ## ## ##
##### ## ##### ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## 4231.
```

5.



####

```
gdb ##### ## ##### ##### ## ## ##### ## #####
##### ## ## ## ## ## up ## down #####
##### ## ## ## ## ##. ##### ## ## ##
##### ## ## ##### ## ## #####, ## ## ## ##
# ##### ## ##### ## ## ##### ## ## ## ##
(## ##### ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ## ##
##### ## ## ## ## ## ## ## ## ##).
```

5.6.3. ##### core

```
## ##### core ## ##### ## ##### ## ##### ## ##—
##### ## ## ## ## ##. ##### # ## ## ## ## ##, ## #####
##### ## ## ## ## ## ## ## ## ## core ## ##### ## ## ##—
##### ## ## ## ## ## ## ## ## ## ## ## ## ## ##. ## ## ##,
##### ## ## ## ## ## ## ## ## ## 4.4##, ## ##### core ## ## ## nomprog.core
##### ## ## ## core, ## ## ## ## ## ## ## ## ## ## ## ## ##
core.
```

```
#### ##### ## ##### core, ##### gdb ## ##### ## ## ## ##. ##### ## ## ## ##
break ## run, #####
```

```
(gdb) core nomprog.core
```

```
## ##### ## ## ## ## ## ## ## ## ## ## ## ## ## core, ## ## ## ## ##
dir /path/to/core/file #####.
```

```
#### ##### ## ## ## ## ## ## ## ## ##:
```

```
% gdb a.out
```

```
GDB is free software and you are welcome to distribute copies of it
under certain conditions; type -"show copying" to see the conditions.
There is absolutely no warranty for GDB; type -"show warranty" for details.
GDB 4.13 (i386-unknown-freebsd), Copyright 1994 Free Software Foundation, Inc.
(gdb) core a.out.core
```

```
Core was generated by `a.out'.
```

```
Program terminated with signal 11, Segmentation fault.
```

```
Cannot access memory at address 0x7020796d.
```

```
#0 0x164a in bazz (anint=0x5) at temp.c:17
```

```
(gdb)
```

```
(gdb) bt
#0  0x164a in bazz (anint=0x5) at temp.c:17
#1  0xefbfd888 in end ()
#2  0x162c in main () at temp.c:11
(gdb)
```

5.6.4. ## ##### ### ## ##### ## #####

```
(gdb) attach identifiant_processus
```

```

...
if ((pid = fork()) < 0) /* _Toujours_ verifier cela */
error();
else if (pid == 0) { /* le fils */
int PauseMode = 1;

while (PauseMode)
    sleep(10); /* Attendre jusqu'a ce que quelqu'un se brache sur nous */
...
} else { /* le pere */
...

```


5.7. #####
#####

#####, ## ##### ## ## ## ## ##### ## ## #####

#####. ⁹ #####, ## ## ##### ## ## ##### ## ##### #####. #####

#####. ## ##### #####. ## ## ## ##
#####.

[illegible]

- ## #####, ##### ## #####
(#####), ##### # ## ## ## ## ## ## ## ## ## ##,
##, ##.

- ⁹##### ## ##### ##### ### ##### #####.

#####

- ### ##### # #####, ### ##### ##### ##### # ## ##### # ##
#####.
- ##### ##### # ##### info ##### ##### # ##### ##—
##, ##### # ##### ## #####.
- ##### ##### # gdb, ### ##### # ##### # ##### # ## #
#####.
- ##### ##### # ## # ##### # ## # ##### # ##### # ## #
#####.

#####.

#####.

C-h t #####
#, ##### #, ##### #
#. (##### #, ##### #
#####).

```
#####  
#####  
#####  
#####  
#####  
#####  
#####  
#####  
#####  
#####
```

```
#####, ## ### ##### ##### ## ##### ### ##### ##### ##
### ### ##### # ### ## ##### ## ##### #####. ## ##### ##
##### ## ##### ## ##### ##, #####, ##### ## ##### ##### ## ## ##
##### ##### ##### ## ## ##### ## ## ##### # ##### ##, #-
##### C-xC-f. ##### ## ## ##### ## ##, ##### # ## ##### ##### ## ##.
```


#####

```
## #####,##### ##### ##### ## #####,##-
## #,##### ## #####,##### replace-s (##### replace-string# ## #####
##### ##### ## ## ##### #####),#####
## ## #####,##### foo (## ##### ## #####),##### ##
#####,##### bar (## ##### ## ##### # foo)####
## #####.##### ## ##### #####-
##### ## #####.
```


• ## ##### ## ##### ## ##### ## ## ## #, #++ ,###,
##.

- % emacsclient nomfichier

```
10##### EDITOR # emacsclient ##  
## #####
```

```
(global-set-key [f5] -'eval-current-buffer)
(global-set-key [f6] -'buffer-menu)
(global-set-key [f7] -'other-window)
(global-set-key [f8] -'find-file)
(global-set-key [f9] -'save-buffer)
(global-set-key [f10] -'next-error)
(global-set-key [f11] -'compile)
(global-set-key [f12] -'grep)
(global-set-key [C-f1] -'compile)
(global-set-key [C-f2] -'grep)
(global-set-key [C-f3] -'next-error)
(global-set-key [C-f4] -'previous-error)
(global-set-key [C-f5] -'display-faces)
(global-set-key [C-f8] -'dired)
(global-set-key [C-f10] -'kill-compilation)

;; Keypad bindings
(global-set-key [up] -"\C-p")
(global-set-key [down] -"\C-n")
(global-set-key [left] -"\C-b")
(global-set-key [right] -"\C-f")
(global-set-key [home] -"\C-a")
(global-set-key [end] -"\C-e")
(global-set-key [prior] -"\M-v")
(global-set-key [next] -"\C-v")
(global-set-key [C-up] -"\M-\C-b")
(global-set-key [C-down] -"\M-\C-f")
(global-set-key [C-left] -"\M-b")
(global-set-key [C-right] -"\M-f")
(global-set-key [C-home] -"\M-<")
(global-set-key [C-end] -"\M->")
(global-set-key [C-prior] -"\M-<")
(global-set-key [C-next] -"\M->")

;; Souris
(global-set-key [mouse-3] -'imenu)

;; Divers
(global-set-key [C-tab] -"\C-q\t") ; Control tab quotes a tab.
(setq backup-by-copying-when-mismatch t)

;; Traite -'y' ou <CR> comme yes, -'n' comme no.
(fset -'yes-or-no-p -'y-or-n-p)
(define-key query-replace-map [return] -'act)
(define-key query-replace-map [?\C-m] -'act)

;; Charge les ajouts
(require -'desktop)
(require -'tar-mode)

;; Diff mode sympa
(autoload -'ediff-buffers -"ediff" -"Intelligent Emacs interface to diff" t)
(autoload -'ediff-files -"ediff" -"Intelligent Emacs interface to diff" t)
```

```

(autoload -'ediff-files-remote -"ediff"
  -"Intelligent Emacs interface to diff")

(if first-time
  (setq auto-mode-alist
    (append -'(("\\.cpp$" -. c++-mode)
      ("\\.hpp$" -. c++-mode)
      ("\\.lsp$" -. lisp-mode)
      ("\\.scm$" -. scheme-mode)
      ("\\.pl$" -. perl-mode)
      -) auto-mode-alist)))

;; Mode de verrouillage automatique de la police de caracteres
(defvar font-lock-auto-mode-list
  (list -'c-mode -'c++-mode -'c++-c-mode -'emacs-lisp-mode -'lisp-mode -'perl-mode -'scheme-
mode)
  -"List of modes to always start in font-lock-mode")

(defvar font-lock-mode-keyword-alist
  -'((c++-c-mode -. c-font-lock-keywords)
    (perl-mode -. perl-font-lock-keywords))
  -"Associations between modes and keywords")

(defun font-lock-auto-mode-select ()
  -"Automatically select font-lock-mode if the current major mode is
in font-lock-auto-mode-list"
  (if (memq major-mode font-lock-auto-mode-list)
    (progn
      (font-lock-mode t))
    -)
  -)

(global-set-key [M-f1] -'font-lock-fontify-buffer)

;; New dabbrev stuff
(require -'new-dabbrev)
(setq dabbrev-always-check-other-buffers t)
(setq dabbrev-abbrev-char-regexp -"\\sw\\\\\\\\\\s_")
(add-hook -'emacs-lisp-mode-hook
  -'(lambda ()
    (set (make-local-variable -'dabbrev-case-fold-search) nil)
    (set (make-local-variable -'dabbrev-case-replace) nil)))
(add-hook -'c-mode-hook
  -'(lambda ()
    (set (make-local-variable -'dabbrev-case-fold-search) nil)
    (set (make-local-variable -'dabbrev-case-replace) nil)))
(add-hook -'text-mode-hook
  -'(lambda ()
    (set (make-local-variable -'dabbrev-case-fold-search) t)
    (set (make-local-variable -'dabbrev-case-replace) t)))

;; mode C++ et C...
(defun my-c++-mode-hook ()

```

5. ##### ##

```
(setq tab-width 4)
(define-key c++-mode-map -"\C-m" -'reindent-then-newline-and-indent)
(define-key c++-mode-map -"\C-ce" -'c-comment-edit)
(setq c++-auto-hungry-initial-state -'none)
(setq c++-delete-function -'backward-delete-char)
(setq c++-tab-always-indent t)
(setq c-indent-level 4)
(setq c-continued-statement-offset 4)
(setq c++-empty-arglist-indent 4))

(defun my-c-mode-hook ()
  (setq tab-width 4)
  (define-key c-mode-map -"\C-m" -'reindent-then-newline-and-indent)
  (define-key c-mode-map -"\C-ce" -'c-comment-edit)
  (setq c-auto-hungry-initial-state -'none)
  (setq c-delete-function -'backward-delete-char)
  (setq c-tab-always-indent t)
  ;; Style d'indentation BSD
  (setq c-indent-level 4)
  (setq c-continued-statement-offset 4)
  (setq c-brace-offset --4)
  (setq c-argdecl-indent 0)
  (setq c-label-offset --4))

;; mode Perl
(defun my-perl-mode-hook ()
  (setq tab-width 4)
  (define-key c++-mode-map -"\C-m" -'reindent-then-newline-and-indent)
  (setq perl-indent-level 4)
  (setq perl-continued-statement-offset 4))

;; mode Scheme...
(defun my-scheme-mode-hook ()
  (define-key scheme-mode-map -"\C-m" -'reindent-then-newline-and-indent))

;; mode Emacs-Lisp...
(defun my-lisp-mode-hook ()
  (define-key lisp-mode-map -"\C-m" -'reindent-then-newline-and-indent)
  (define-key lisp-mode-map -"\C-i" -'lisp-indent-line)
  (define-key lisp-mode-map -"\C-j" -'eval-print-last-sexp))

;; Ajoute tout le reste...
(add-hook -'c++-mode-hook -'my-c++-mode-hook)
(add-hook -'c-mode-hook -'my-c-mode-hook)
(add-hook -'scheme-mode-hook -'my-scheme-mode-hook)
(add-hook -'emacs-lisp-mode-hook -'my-lisp-mode-hook)
(add-hook -'lisp-mode-hook -'my-lisp-mode-hook)
(add-hook -'perl-mode-hook -'my-perl-mode-hook)

;; Le complement a next-error
(defun previous-error (n)
  -"Visit previous compilation error message and corresponding source code."
  (interactive -"p"))
```

```

(next-error (- n)))

;; Divers...
(transient-mark-mode 1)
(setq mark-even-if-inactive t)
(setq visible-bell nil)
(setq next-line-add-newlines nil)
(setq compile-command "make")
(setq suggest-key-bindings nil)
(put 'eval-expression 'disabled nil)
(put 'narrow-to-region 'disabled nil)
(put 'set-goal-column 'disabled nil)

;; Recherche des archives Elisp
(autoload 'format-lisp-code-directory "lispdir" nil t)
(autoload 'lisp-dir-afropos "lispdir" nil t)
(autoload 'lisp-dir-retrieve "lispdir" nil t)
(autoload 'lisp-dir-verify "lispdir" nil t)

;; Mode de verrouillage de police
(defun my-make-face (face colour &optional bold)
  "Create a face from a colour and optionally make it bold"
  (make-face face)
  (copy-face 'default face)
  (set-face-foreground face colour)
  (if bold (make-face-bold face))
  -)

(if (eq window-system 'x)
    (progn
      (my-make-face 'blue "blue")
      (my-make-face 'red "red")
      (my-make-face 'green "dark green")
      (setq font-lock-comment-face 'blue)
      (setq font-lock-string-face 'bold)
      (setq font-lock-type-face 'bold)
      (setq font-lock-keyword-face 'bold)
      (setq font-lock-function-name-face 'red)
      (setq font-lock-doc-string-face 'green)
      (add-hook 'find-file-hooks 'font-lock-auto-mode-select)

      (setq baud-rate 1000000)
      (global-set-key "\C-cmm" 'menu-bar-mode)
      (global-set-key "\C-cms" 'scroll-bar-mode)
      (global-set-key [backspace] 'backward-delete-char)
      ; (global-set-key [delete] 'delete-char)
      (standard-display-european t)
      (load-library "iso-transl"))))

;; X11 ou PC utilisant les ecritures directes a l'ecran
(if window-system
    (progn
      -;; (global-set-key [M-f1] 'hilit-repaint-command)

```


5. ##### ##

```
-;; (global-set-key [M-f2] [?(C-u M-f1])
(setq hilit-mode-enable-list
-'(not text-mode c-mode c++-mode emacs-lisp-mode lisp-mode
scheme-mode)
hilit-auto-highlight nil
hilit-auto-rehighlight -'visible
hilit-inhibit-hooks nil
hilit-inhibit-rebinding t)
(require -'hilit19)
(require -'paren))
(setq baud-rate 2400) ; For slow serial connections
-)

;; Terminal de type TTY
(if (and (not window-system)
(not (equal system-type -'ms-dos)))
(progn
(if first-time
(progn
(keyboard-translate -?(C-h -?(C-?)
(keyboard-translate -?(C-? -?(C-h))))))

;; Sous Unix
(if (not (equal system-type -'ms-dos))
(progn
(if first-time
(server-start))))

;; Add any face changes here
(add-hook -'term-setup-hook -'my-term-setup-hook)
(defun my-term-setup-hook ()
(if (eq window-system -'pc)
(progn
;; (set-face-background -'default -"red")
)))

;; Restaure le -"desktop" -- faire cela le plus tard possible
(if first-time
(progn
(desktop-load-default)
(desktop-read)))

;; Indique que ce fichier a ete lu au moins une fois
(setq first-time nil)

;; Pas besoin de deverminer quoique ce soit maintenant

(setq debug-on-error nil)

;; Tout est fait
(message -"All done, %s%s" (user-login-name) -".")
```

```
##### ## ##### ## #####
##### #####
```

5.7.4. ##### ## ##### ## #####

```
#####, ##### ## ##### ## ##### ##### ##### ##### #####
##### ##### ##### ## ##### .emacs (#, #+++, #####, ##### ## #####),
##### ##### ## ## ##### ##### ##### # ##### # ####, #####
##### ##### ?
```

```
## ##### ##### # ##### ## ## ##### ## ##### ##### ## #####
## ##### #####. ##### ## ##### ##### ##### ## .el, ##-
##### ##### # ##### ##### #. #####, ## ##### ## ## ##### #####
##### ##### ##### ## ##### ## #####
```

```
% find -/usr/ports/lang/whizbang --name -"*.el" --print
```

```
## ## ##### ## ## ##### ## ## ##### ## #####. ## #####
2.1.0#####, ## ##### ## /usr/local/share/emacs/site-lisp11.
```

```
##### ## #####, ## ## ##### ## ## ##### find #####
```

```
/usr/ports/lang/whizbang/work/misc/whizbang.el
```

```
#### #####
```

```
# cp -/usr/ports/lang/whizbang/work/misc/whizbang.el -/usr/local/share/emacs/site-lisp
```

```
#####, ##### ##### ##### ##### ##### ##### ##### #####
##### ##### ## ##### ## ##### ## .wiz. ##### #####
##### ##### ##### .emacs ##### ## ##### #####
##### ##### whizbang.el.
```

```
##### ##### ##### .emacs ## ##### ## ##### ## #####
##### :
```

```
...
(")\.lsp$" -. lisp-mode)
(")\.wiz$" -. whizbang-mode)
(")\.scm$" -. scheme-mode)
...
```

```
#### ##### ## ##### ##### ## ##### whizbang-mode #####
##### ## ##### ## ##### ## .wiz.
```

¹¹### : ## ##### 4.2#####.

5. ##### ##

, ##### ##### #####. #####
whizbang-mode # ##### ## :

```
:: Auto font lock mode
(defvar font-lock-auto-mode-list
  (list -'c-mode -'c++-mode -'c++-c-mode -'emacs-lisp-mode -'whizbang-mode -'lisp-mode -'perl-
mode -'scheme-mode)
  -"List of modes to always start in font-lock-mode")
```

font-lock-mode (## ##### ##
#####) ##### .wiz.

#####. ##### ## ##### ##
.wiz, #####
whizbang-mode hook (#### my-scheme-mode-hook ##### ## auto-indent,
#####).

5.8.

- ##### *Simply Scheme* ### 1994. #### 0#262#08226#8
- ##### *Learning Perl* ##### 1993 #### 1#56592#042#2
- ##### *Lisp (3rd Edition)* #####—
1989 #### 0#201#08319#1
- ##### #. ##### *The Unix Programming Environment* #####—
1984 #### 0#13#937681##
- ##### #. ##### *The C Programming Language (2nd Edition)* ##### 1988 #### 0#13#110362#8
- ##### *The C++ Programming Language* ##### 1991 ####
0#201#53992#6
- #. ##### *Advanced Programming in the Unix Environment* #####
1992 #### 0#201#56317#7
- #. ##### *Unix Network Programming* ##### 1990 ####
0#13#949876#1

6.

#####

#####.

6.1.

##—
##—

#####.

6.2. ##### ##

##. ## ##### ##### ##### ##### ## ##### ## ##—

#####. ## #####

#####.

#####—
#####. ## ##### ## ##### ##### #####
(#### ##### ##), ## #####
#####, ## ##### ## #####. ## ##—
#####, #####, ## ##—
#####.

6.3. ##### ##

(#####) ##### ##### ##
1. ## ##### ## ##### ## 1988 ##
#####. #####, ## #####
#####. ## 17 ##### ## ##### ## 1999, 10 #####
#####. ## ##

##.

#####

(##### : #####) ##### ## #####
#####. ##### ## ##### ##### ## #####—
#####. ##### ##### ##### #####

#####. ## ##### ## ##### ## #####—
#####. ##### #####

#####, ##### ##### ##### ##### ## #####

#####—
#####. 1 ##### ## ##### ## #####
#####, ## ##### ## ##### ## #####

#####, ##### ##### # ## #####—
#####.

#####, ## ##### ##### ##### ## ##### #####
(#####/#####).

#####. ## #####,
#####—
#####.

strcpy(#####, #####)	#####
strcat(#####, #####)	#####
getwd(#####)	#####
gets(####)	#####
[vf]scanf(#####, ...)	#####.
realpath(#####, #####_####[!])	#####
[v]sprintf(####, #####, #####— ###, ...)	#####.

6.3.1.

#####. (##### 4)

```
#include <stdio.h>
```

```
void manipulate(char *buffer) {
    char newbuffer[80];
    strcpy(newbuffer,buffer);
}

int main() {
    char ch,buffer[4096];
    int i=0;

    while ((buffer[i++] = getchar()) != -'\n') { };

    i=1;
    manipulate(buffer);
    i=2;
    printf("La valeur de i est -: %d\n",i);
    return 0;
}
```

[### ##### ###!]

6.3.2. #####

```
#####, ### ##### ##### # # ##### # ##### # ## #####
##### # # #####. ## ##### strcpy # strcat #####
##### # ##### # ##### # # ##### # # #####
##### # ##### # ##### # #. ##### #####
```

#####

#####, #### 6. ### ##### strcpy ## strcat ##### ## ## #####
3.5.

6.3.2.1. ##233;##### ## #####
#####

#####. #####, # #

#####. ##### #####

#/#++.

(#####) ##### ## ## ## ## ## ## ## ## ## ## ## ##
###:/#####.###/#####.### :

#####. ##### ## ## ## ## ¹ # ## ##

#####. ## ## ## ## ## ## ## ## ##

(#####) ## ## ## ##
#####.

##, ##### ## ## ## ## ##
#####_#####() ## #####_#####(). #####_#####() #

#####_#####. ## #####_#####

#####.

#####.

6.3.2.2. ##### ## ## ## ## ## ## ## ## ##
####

#####. ## ## ## ## ## ## ## ## ##
##

¹### : ##### ## ## ## ## ## ## ## ## ##

6.

(strcpy, fscanf, getwd, ##..) ## ##### ## ## ##### ## ##### ##
##.

- #####
- #####
- #####

#####. ## ##### ##### ##### ##### #####
#####. ##

#####. ## ##, ## ##### ##### ##### ##_##### ## ##_#####_####
#####/### ##### ## #####.

6.4. ## ##### ##

6 ##### ## (#####) ##### # ## ##### #####. #
##, ##### ##### ##### ##### ##### ## #####
#####. ## #####, ##### ## ##### #####
##—
#####.

(#### ####) ##### #####
#####. ## ##### ##### ## #####
#####.

(##### ####) ## ## # #### ##
exec() ## ## ##### ##### ## ## ##### #####. ## #####—
seteuid() # ##### ##### ##### ##### #####

#####. ##### ##### ## ##### ##### ## #####
exec(), ## ##### ##### ## ##### #####.

6.5.

chroot(). ## ##### ##### ## ##### ##### ##—
##. ####
#####, ## ##### ##### ##### (#####) ## #####—
#####. ## ##### ##### #####
chdir() #####. ## ##### #####
#####.

#####

#####. #####

#####.

```
## ##### ## ##### chroot() #### ## ### ##### ## ##-  
##### sysctl ## ## #####.#####_#####_. ##### ##  
#### ## ##### # 0, chroot() ##### ## ## ## #####.  
## ## ##### ## ## ##### ## ##### 1, ##### chroot() #####  
##### # # ## ##### ## ## ## ## ## ##  
##### chroot().### ## ## ##, ## #####  
#####.
```

6.5.1. ## ##### ## #####

```
## ##### ## ##### (#####) ##### chroot() ## ##### ## ##### ## #####—
##### ##### ## ##### #####. ## ## #####
## ## ## ##, ##### ##### ##### ## ## ## ## ##
##### ## #####, ## ## ## #####
##### ## #####.
```

```
#####  
##### # #####, #### # ##  
##### # ##### # suser() #####. #####  
# suser() ### # ##### suser_xxx(). #####  
#####  
#####.
```

```
## #####  
## :
```

- ##### ## ##### ##### setuid, seteuid, setgid, setegid, setgroups, seteuid, setregid, setlogin
- ##### ## ##### ##### setrlimit
- ##### ##### ## ##### (#####)
- chroot()
- ##### ## ##### ##### (#####): chflags, fchflags
- ##### ## ##### ##### ##### ##### ##### #####, ## #####, ## #####, ## #####, ## #####, ## ## #####.
- ## ## # ## ##### ##### ##### (##### < 1024)

Jail ### ## ##### ##### ##### ##### ##### ##### ##### ##### ##### ## #####—
#####. #####, ## ##### ## (#—

6.

#####) ##### suser_xxx #####
#####. #####

#####.

6.5.2. #####.1#

#####.

#####. #####
(###_###_###(3)).

6.6.

#####.

#####.

#####.

6.7.

#####.

#####

#####.

##, ## #—
#####. ### ##### ##### ## #####
#—
#####. ##### ## ##### access(2) ##### open(2) ##### #####.
#####. ## #—
seteuid() #####
open() #####. ##### ## #####, ## ##### #####
open() ##### ## ##### ## #—
chmod().

###. ##

#####

7. ##### ## ##### ##### 63

8. ##### ## ## ##### 65

 8.1. ### ##### 65

 8.2. ### ##### ## ##### (#####) 68

 8.3. ##### ##### 68

7. ##### ## #####

#####/###, ### ##### #####, ##### ####-
#####, #####, #####, ### ##### (#####), ## #####
#####, ## #####, ### #####, ### #####, ##.

8. ##### ###

#####

Ce chapitre est maintenu par The FreeBSD SMP Next Generation Project. Envoyez leur directement les commentaires et les suggestions à [liste de diffusion concernant le traitement symétrique multiprocesseurs \(SMP\) sous FreeBSD](#).

```
## ##### ##### ## ##### ##### ##### ##### ##### ##—
##### ##### ## ##### ##### # ##### ## #####. ## ##—
##### ##### ##### ## ##### #####. ### ##### ## #####
##### ##### ##### ## ## ##### ## #####(9) #####. #####
##### ##### ##### ## ##### ##### ##### ##### ##—
##### ##### # #####.
```

8.1. ###

```
## ##### ## ##### ## ##### ##### ##### #####.
#####., ## ##### ## ##### ##### ## #####. ## ##
##### ##### ##### ## ##### ##### , ##### ##### ##### ##
## ## ##### #####. ##### ## ##### , ## ##### #####
#####.
```

```
## ##### ##### ##### #####., ##### ## ##### #####
##### ##### #####. #####., ## ##### ## ##### ##
##### ##### ##### ## #####. ## ##### ##### ##
##### ##### #####., ##### ## #####(9) ##### (#####).
```

```
##### ##### # ##### :
```

```
### ## ## #####
```

```
### ## ## ##### ##### ## ## ## ##### ## #####.
```

```
### #####
```

```
## ## ## ##### ## ## ##### ## mtx_init. ## ## ## ##### ## ## ##—
##### ## ##### ##., ##### ## ##### ## #####
##### ##### ## ##### ## #####.
```

```
####
```

```
## ##### ## ##### ## ##### ##### MTX_*. ## ##### ##
##### ##### ##### ## ##### ##### ##### #####(9).
```

```
MTX_DEF
```

```
## #####
```

```
MTX_SPIN
    ## #####

MTX_COLD
    ## #####. ## #####
    MUTEX_DECLARE, ## ## MTX_COLD ## ##
    mtx_init.

MTX_TOPHALF
    ## #####.

MTX_NORECURSE
    ## #####.
```

```
#####
    ### #####
    #####. #####
    ##### structure name.member name.
```

```
#####
    #####.
```

8.1.

### ## #- #####	#####	####	#####	##### #- #####
#####_####	# ##### #	MTX_SPIN MTX_COLD	_gmonparam, cnt.v_switch, cp_time, curpriority, mtx.mtx_blocked, mtx.mtx_contested, proc.p_contested, proc.p_blocked, proc.p_flag (P_PROFIL ###, P_INMEM, P_SINTR, P_TIMEOUT, P_SWAPINREQ ###, P_INMEN ##), proc.p_nice, proc.p_procq, proc.p_blocked, proc.p_estcpu, proc.p_nativepri, proc.p_priority,	setrunqueue, remrun- queue, mi_switch, chooseproc, sched- clock, resetprio- rity, updatepri, maybe_resched, cpu_switch, cpu_throw

8.

### ## ## #-- #####	### #####	####	#####	##### #-- #####
			proc.p_usrpri, proc.p_rtprio, proc.p_rqindex, proc.p_stats->p_prof, proc.p_stats->p_ru, proc.p_stat, proc.p_cpticks, proc.p_iticks, proc.p_uticks, proc.p_sticks, proc.p_swtime, proc.p_slptime, proc.p_runtime, proc.p_pctcpu, proc.p_oncpu, proc.p_asleep, proc.p_wchan, proc.p_wmesg, proc.p_slpq, proc.p_vmspace (### # ## statclock), pscnt, slpq, itqueuebits, itqueues, rtqueue- bits, rtqueues, queue- bits, queues, id- queuebits, idqueues, switchtime,	
##86###_####	# ##86### #### #	MTX_DEF MTX_COLD	vm86pcb	vm86_bioscall
#####	# ##### #	MTX_DEF MTX_COLD	##### #-- #####	####
#####_####	# ##### #### #	MTX_SPIN	callfree, callw- heel, nextsoftcheck, proc.p_icalout, proc.p_slpcallout, softticks, ticks	

(#### #####)

8.2. ### ##### ## ##### (#### #-
#####)

#####(9) ##### ## ##### ## #-
#####. ### ##### ## ## ##### ##### ## #####
#####.

8.2. #####(9) ##### ##

### ## ## #####	#####		
allproc_lock	allproc zombproc pidhasht- bl proc.p_list proc.p_hash nextpid	proctree_lock	proc.p_children proc.p_sibling

8.3. ##### ##### #####

#-
#####. #####, ##### ## ##### ## #####
#####(9). #####

#-
#####.

- astpending
- mtx.mtx_lock

##. ##### ##
#####

#####

9. ## ##### ##### 73

9. ##

#####

##, ##### ## ##, ##### ## ##, ##### ## ##, ##### ## ##
#####, ##, ##, ##.

#. ##### #/
(#####/#####)

#####

10. ### 79

10.

###, ###, ###2##, ###, #####, #####, ##### # #### ## #####
#####, #####, #####, #####, ###, #####, #####, #####, ###—
#####, ##### ## #####, ####, ##### ##

##. #####
(###)

#####

11. ##### 85

11.

#####, #####, #####, ##### ## #####, ##### ## ##### ##
#####, #####, #####

###. ##

#####

12. ##### 91

12.

#####, ###, ##, ###, ###, #####, ###, #####, #####, ##### (###),
#####,

#####.

#####

#####

13. ### 97

13.

###, ###, ##### ###]

##. ##### ##

#####

14. ##### 103

14.

#####, ###, ####, ##### ## ##### ####, #####, #####, ###

#. ##

#####

15. ### 109

15.

###, #####, ###

##.

#####

#####

16. #####	115
16.1. #####	115
16.2. #####	115
16.3. #####	117
16.4. #####	117
16.5. #####	121
17. ###	123
17.1. #####	123
17.2. ###	127
18. ##### (###) ##	131
18.1. ##	131
19. #####	133
19.1. #####	133
20. #####	135

16.

#####

#####

##—
##(4) ## ##### #####.

16.1.

#####—
#####. ## #####, ## ## #####, ## ## #####
#####,
#####, #####, ## ## ##### ## ## #####. ## ##### ## —

#####. ## # # ##### ## ## ##### ## #####—
(#####) ## ## ##### ## ##### ## #####
#####. ##

#####.

(#####), ##—
#####. ## ##### ##### #####
/dev ## ## ##### ## #####. ##### ## ##
#####, ##### ##### ## #####
—
#####. ## ##### ## ##### ## #####
MAKEDEV.

#####.

16.2. ##### ## ##### ## ##### #

—
#####. ##### ## #####—

#####.

```
##### ## ##### ##### ## #####
# ##
##### ## ## ##### ## ##### ## ##### #####-
##### :
```

- kldload # ##### ## ##### ##### ## ## #####
- kldunload # ##### ## ##### ## #####
- kldstat # ##### ## ##### ##### ## ## #####

```
##### ##### ##### ## #####
```

```
/*
 * Squelette KLD
 * Inspiré de l'article d'Andrew Reiter paru sur Daemonnews
 */

#include <sys/types.h>
#include <sys/module.h>
#include <sys/systm.h> -/* uprintf */
#include <sys/errno.h>
#include <sys/param.h> -/* defines utilise dans kernel.h */
#include <sys/kernel.h> -/* types utilise dans le module d'initialisation */

/*
 * charge le gestionnaire qui traite du chargement et déchargement d'un KLD.
 */

static int
skel_loader(struct module *m, int what, void *arg)
{
    int err = 0;

    switch (what) {
        case MOD_LOAD:          -/* kldload */
            -
            uprintf("Skeleton KLD charge.\n");
            break;
        case MOD_UNLOAD:
            uprintf("Skeleton KLD decharge.\n");
            break;
        default:
            err = EINVAL;
            break;
    }
    return(err);
}

/* Declare ce module au reste du noyau */

DECLARE_MODULE(skeleton, skel_loader, SI_SUB_KLD, SI_ORDER_ANY);
```

```
##### 16. ##### ## ##### ## #-  
##### ## #####
```

16.2.1.

```
##### ## ##### ##### ##### ## ## #####  
##### ##### ## ## ##.
```

```
SRCS=skeleton.c  
KMOD=skeleton  
  
.include <bsd.kmod.mk>
```

```
##### ## ##### make ## ## ##### ##### ##### ## #####  
skeleton.ko ## ## ## ## ## ## ## ## ## :
```

```
#  
kldload -v -./skeleton.ko
```

16.3. ##### ## #####

```
#### ##### ## ##### ##### ##### ##### ##### ## ## #####-  
##### ## #####. ## ##### ##### ## ##### ##### ## #####  
## ##### ## ##### ##### ## ##### ##### ## #####  
##### ## #####. ## ##### /dev/MAKEDEV ##### ## ##### ## #####  
##### ## ##### ##### ##### ##### ## ## ##### #####  
##### ## #####, ## ##### ##### ##### ## ##### ## #####  
##### ## ##### ##### ## ##### mknod
```

16.3.1. ##### ## ##### ##### ##

```
## ##### mknod ##### ##### ##### ## ## ## ##### ##  
#####. ## ##### ## ## ## ## ##### ##### ## #####  
## ## ## #####, ## ##### ## ## ## ##### ## #####
```

16.3.2. ## ##### ##

```
## ##### ## #####, ## #####, ##### ##### # ##### ##  
#### ## ##### ## ##### ##### ## ##### ## #####  
#### ##### ## ##### ## ##### ##### #####, ## ## #-  
##### ##### #####. ##### ## ##### ## ##### ##  
## ##### ## #####.
```

16.4. ##

```
## ##### ## ##### ## ## ##### ## ##### ## #-  
##### ## ##### ## #####. ## ##### ## #####
```

#####

#####.

#####.

```
/*
 * un simple pseudo-périphérique `echo' KLD
 *
 * Murray Stokely
 */

#define MIN(a,b) (((a) < (b)) ? (a) : (b))

#include <sys/types.h>
#include <sys/module.h>
#include <sys/systm.h> -/* uprintf */
#include <sys/errno.h>
#include <sys/param.h> -/* defines utilises dans kernel.h */
#include <sys/kernel.h> -/* types utilises dans me module d'initialisation */
#include <sys/conf.h> -/* cdevsw struct */
#include <sys/uio.h> -/* uio struct */
#include <sys/malloc.h>

#define BUFFERSIZE 256

/* Prototypes des fonctions */
d_open_t    echo_open;
d_close_t   echo_close;
d_read_t    echo_read;
d_write_t   echo_write;

/* Points d'entrée du périphérique Caractère */
static struct cdevsw echo_cdevsw = {
    echo_open,
    echo_close,
    echo_read,
    echo_write,
    noioctl,
    nopoll,
    nommap,
    nostrategy,
    -"echo",
    33,          -/* reserve pour lkms -- -/usr/src/sys/conf/majors */
    nodump,
    nopsize,
    D_TTY,
    --1
};

typedef struct s_echo {
    char msg[BUFFERSIZE];
    int len;
```


16. #####
#####

```
} t_echo;

/* variables */
static dev_t sdev;
static int len;
static int count;
static t_echo *echomsg;

MALLOC_DECLARE(M_ECHOBUF);
MALLOC_DEFINE(M_ECHOBUF, -"echobuffer", -"cache pour le module echo");

/*
 * Cette fonction est appele par les appels systeme kld[un]load(2) pour
 * determiner quelles actions doivent etre faites quand le
 * module est charge ou decharge
 */

static int
echo_loader(struct module *m, int what, void *arg)
{
    int err = 0;

    switch (what) {
    case MOD_LOAD:          /* kldload */
        sdev = make_dev(&echo_cdevsw,
            0,
            UID_ROOT,
            GID_WHEEL,
            0600,
            -"echo");
        /* aloocation de memoire noyau pour l'utilisation de ce module */
        /* malloc(256,M_ECHOBUF,M_WAITOK); */
        MALLOC(echomsg, t_echo *, sizeof(t_echo), M_ECHOBUF, M_WAITOK);
        printf("Peripherique Echo charge.\n");
        break;
    case MOD_UNLOAD:
        destroy_dev(sdev);
        FREE(echomsg, M_ECHOBUF);
        printf("Peripherique Echo decharge.\n");
        break;
    default:
        err = EINVAL;
        break;
    }
    return(err);
}

int
echo_open(dev_t dev, int oflags, int devtype, struct proc *p)
{
    int err = 0;

    uprintf("Peripherique \"echo\" ouvert avec succes.\n");
}
```

```

    return(err);
}

int
echo_close(dev_t dev, int fflag, int devtype, struct proc *p)
{
    uprintf("Fermeture du peripherique \"echo.\\n");
    return(0);
}

/*
 * La fonction read prend juste comme parametre
 * le cache qui a ete sauve par l'appel à echo_write()
 * et le retourne a l'utilisateur pour acces.
 * uio(9)
 */

int
echo_read(dev_t dev, struct uio *uio, int ioflag)
{
    int err = 0;
    int amt;

    /* De quelle taille est cette operation read -? Aussi grande que l'utilisateur le veut,
       ou aussi grande que les donnees restantes */
    amt = MIN(uio->uio_resid, (echomsg->len -- uio->uio_offset > 0) ? echomsg->len -- uio-
    >uio_offset -: 0);
    if ((err = uiomove(echomsg->msg + uio->uio_offset,amt,uio)) != 0) {
        uprintf("uiomove echoue!\\n");
    }

    return err;
}

/*
 * echo_write prend un caractere en entree et le sauve
 * dans le cache pour une utilisation ulterieure.
 */

int
echo_write(dev_t dev, struct uio *uio, int ioflag)
{
    int err = 0;

    /* Copie la chaine d'entree de la memoire de l'utilisateur a la memoire du noyau */
    err = copyin(uio->uio_iov->iiov_base, echomsg->msg, MIN(uio->uio_iov->iiov_len,BUFFERSIZE));

    /* Maintenant nous avons besoin de terminer la chaine par NULL */
    *(echomsg->msg + MIN(uio->uio_iov->iiov_len,BUFFERSIZE)) = 0;
    /* Enregistre la taille */
    echomsg->len = MIN(uio->uio_iov->iiov_len,BUFFERSIZE);

    if (err != 0) {

```

```
##### 16. ##### ## ##-
#####
```

```
    uprintf("Ecriture echouee: mauvaise adresse!\n");
-}

    count++;
    return(err);
}

DEV_MODULE(echo,echo_loader,NULL);
```

```
#### ##### ## #####, ##### ##### ## ##### #####
##### ## ##### ## ## ##### :
```

```
# mknod -/dev/echo c 33 0
```

```
#### ## #####, ##### ##### ## ##### ## #####
##### :
```

```
    # echo --n -"Test Donnees" > -/dev/echo
    # cat -/dev/echo
Test Donnees
```

```
##### ## ## ##.
```

```
#####
```

- ##### (###) ##### # ##### ##-
2000
- ### ## ##### ## ## # ##### ## 2000

16.5.

```
## ##### ##### ##### ##### ## ## #####
## ##### #####. ##### ## #####
##### # ##### ## ##### ## ##### ##(), #####
##### ## ## ##### ## ## ##### ##-
####(2).
```

```
## #####(), #####, ##### ## ## ##, ##..
```


17.

#####

#####—
##.

17.1.

#####—

(##) ##### ## ## ##.

```
/*
 * Simple KLD pour jouer avec les fonctions PCI.
 *
 * Murray Stokely
 */

#define MIN(a,b) (((a) < (b)) ? (a) : (b))

#include <sys/types.h>
#include <sys/module.h>
#include <sys/systm.h> /* uprintf */
#include <sys/errno.h>
#include <sys/param.h> /* defines used in kernel.h */
#include <sys/kernel.h> /* types used in module initialization */
#include <sys/conf.h> /* cdevsw struct */
#include <sys/uio.h> /* uio struct */
#include <sys/malloc.h>
#include <sys/bus.h> /* structs, prototypes for pci bus stuff */

#include <pci/pcivar.h> /* For get_pci macros! */

/* Prototypes des fonctions */
d_open_t mypci_open;
d_close_t mypci_close;
d_read_t mypci_read;
d_write_t mypci_write;

/* Points d'entrée du pilote de périphérique caractère */

static struct cdevsw mypci_cdevsw = {
    mypci_open,
    mypci_close,
    mypci_read,
    mypci_write,
    noioctl,
```

```
    nopoll,
    nommap,
    nostrategy,
    -"mypci",
    36,          /* reserved for lkms -- /usr/src/sys/conf/majors */
    nodump,
    nopsize,
    D_TTY,
    --1
};

/* variables */
static dev_t sdev;

/* Nous sommes plus interresses dans la recherche/attachement
que dans l'ouverture/fermeture/lecture/ecriture a ce point */

int
mypci_open(dev_t dev, int oflags, int devtype, struct proc *p)
{
    int err = 0;

    uprintf("Peripherique \"monpci\" ouvert avec succes.\n");
    return(err);
}

int
mypci_close(dev_t dev, int fflag, int devtype, struct proc *p)
{
    int err=0;

    uprintf("Peripherique \"monpci\" -\"ferme\"\n");
    return(err);
}

int
mypci_read(dev_t dev, struct uio *uio, int ioflag)
{
    int err = 0;

    uprintf("lecture dans monpci!\n");
    return err;
}

int
mypci_write(dev_t dev, struct uio *uio, int ioflag)
{
    int err = 0;

    uprintf("Ecriture dans monpci!\n");
    return(err);
}
```

17. ###

```
/* PCI Support Functions */

/*
 * Retourne la chaine d'identification si ce peripherique est le notre
 */
static int
mypci_probe(device_t dev)
{
    uprintf("MonPCI Probe\n"
        "-ID Fabricant: 0x%x\n"
        "-ID Peripherique -: 0x%x\n",pci_get_vendor(dev),pci_get_device(dev));

    if (pci_get_vendor(dev) == 0x11c1) {
        uprintf("Nous avons le WinModem, recherche reussi!\n");
        return 0;
    }

    return ENXIO;
}

/* La fonction Attach n'est appelée que si
la recherche est reussie*/

static int
mypci_attach(device_t dev)
{
    uprintf("Rattachement de MonPCI pour: ID Peripherique: 0x%x\n",pci_get_vendor(dev));
    sdev = make_dev(&mypci_devsw,
        0,
        UID_ROOT,
        GID_WHEEL,
        0600,
        "-monpci");
    uprintf("Peripherique Monpci charge.\n");
    return ENXIO;
}

/* Detach le peripherique. */

static int
mypci_detach(device_t dev)
{
    uprintf("Monpci detache!\n");
    return 0;
}

/* Appele lors de l'arret du systeme apres sync. */

static int
mypci_shutdown(device_t dev)
{
    uprintf("Monpci arrete!\n");
    return 0;
}
```

```

}

/*
 * routine de suspension du peripherique
 */
static int
mypci_suspend(device_t dev)
{
    uprintf("Monpci suspendu!\n");
    return 0;
}

/*
 * routine de reprise du peripherique
 */

static int
mypci_resume(device_t dev)
{
    uprintf("Monpci resume!\n");
    return 0;
}

static device_method_t mypci_methods[] = {
    /* Interface Peripherique*/
    DEVMETHOD(device_probe, mypci_probe),
    DEVMETHOD(device_attach, mypci_attach),
    DEVMETHOD(device_detach, mypci_detach),
    DEVMETHOD(device_shutdown, mypci_shutdown),
    DEVMETHOD(device_suspend, mypci_suspend),
    DEVMETHOD(device_resume, mypci_resume),

    { 0, 0 - }
};

static driver_t mypci_driver = {
    "monpci",
    mypci_methods,
    0,
    /* sizeof(struct mypci_softc), */
};

static devclass_t mypci_devclass;

DRIVER_MODULE(mypci, pci, mypci_driver, mypci_devclass, 0, 0);

```

#####

- **### #####**
- **### #####, ##### ## ### #####, ## ##.**

17. ###

17.2. ###

#####. #####
(###, ###, ###, ###, ###) ## #####

(#####
#####).

17.2.1.

registres d'adresse de base (#####

bus_alloc_resource().

attach() ##### :

```
sc->bar0id = 0x10;  
sc->bar0res = bus_alloc_resource(dev, SYS_RES_MEMORY, &(sc->bar0id),  
0, ~0, 1, RF_ACTIVE);  
if (sc->bar0res == NULL) {  
    uprintf("Allocation memoire du registre PCI de base 0 echouee!\n");  
    error = ENXIO;  
    goto fail1;  
-}  
  
sc->bar1id = 0x14;  
sc->bar1res = bus_alloc_resource(dev, SYS_RES_MEMORY, &(sc->bar1id),  
0, ~0, 1, RF_ACTIVE);  
if (sc->bar1res == NULL) {  
    uprintf("Allocation memoire du registre PCI de base 1 echouee!\n");  
    error = ENXIO;  
    goto fail2;  
-}  
sc->bar0_bt = rman_get_bustag(sc->bar0res);  
sc->bar0_bh = rman_get_bushandle(sc->bar0res);  
sc->bar1_bt = rman_get_bustag(sc->bar1res);  
sc->bar1_bh = rman_get_bushandle(sc->bar1res);
```


#####.

bus_space_*. #####

:

uint16_t

#####

```
board_read(struct ni_softc *sc, uint16_t address) {
    return bus_space_read_2(sc->bar1_bt, sc->bar1_bh, address);
}
```

#####, ### ##### ##### ## :

```
void
board_write(struct ni_softc *sc, uint16_t address, uint16_t value) {
    bus_space_write_2(sc->bar1_bt, sc->bar1_bh, address, value);
}
```

8###, 16### ## 32### ## #####
bus_space_{read|write}_{1|2|4} ## #####.

17.2.2. ###

#####. ##### ##
#####, ## #####
##.

#####, ## ##### attach() ## #####.

/* Recupere la ressource IRQ */

```
sc->irqid = 0x0;
sc->irqres = bus_alloc_resource(dev, SYS_RES_IRQ, &(sc->irqid),
    0, ~0, 1, RF_SHAREABLE -| RF_ACTIVE);
if (sc->irqres == NULL) {
    uprintf("Allocation IRQ echouee!\n");
    error = ENXIO;
    goto fail3;
}
```

/* Maintenant nous choisissons notre gestionnaire d'interruption */

```
error = bus_setup_intr(dev, sc->irqres, INTR_TYPE_MISC,
    my_handler, sc, &(sc->handler));
if (error) {
    printf("Ne peut regler l'IRQ\n");
    goto fail4;
}
```

```
sc->irq_bt = rman_get_bustag(sc->irqres);
sc->irq_bh = rman_get_bushandle(sc->irqres);
```

17.2.3.

##, ### #####
#####.

17. ###

#####. #####, ## # ## vtophys() #####.

```
#include <vm/vm.h>
#include <vm/pmap.h>

#define vtophys(virtual_address) (...)
```

##, ## ## ## ##
vtobus().

```
#if defined(__alpha__)
#define vtobus(va)    alpha_XXX_dmamap((vm_offset_t)va)
#else
#define vtobus(va)    vtophys(va)
#endif
```

17.2.4. #####

attach(). #####

#####.

18. #####
(###)

18.1. ## #####

#####

19.

###

Ce chapitre a été écrit par Nick Hibma. Les modifications pour le manuel par Murray Stokely.

19.1.

#####.

20.

#####.

###.

#####

21. ###32	141
22. #####	143
23. ###64	145

21. ###32

#86 #####.

22.

#####.

#####, ##### ## #####, ##### ## #####—
##.

#####/#####.

23. ###64

##64 #### #####.

#####.

#####

24. ##### 151

24.

#####

###. ### #####
#####

#####

25. ##### 157

25.

#####, ###4, ###

##.

#####

..... 163

#####

- [1] ##### # ##### # #####. ##### © 1998 ##### #####—
#####, ###.. 1#55860#428#6. ##### #####, ###.. *Computer
Organization and Design*. ### ##### / #####. 1#2.
- [2] #. #####. ##### © 1993 ##### #####, ###..
0#201#56317#7. ##### #####, ###.. *Advanced Programming in the
Unix Environment*. 1#2.
- [3] ##### # #####, ##### # #####, ## # #—
#####. ##### © 1996 ##### #####, ###..
0#201#54979#4. ##### #####, ###.. *The Design and
Implementation of the 4.4 BSD Operating System*. 1#2.
- [4] #####. Phrack 49; "Smashing the Stack for Fun and Profit".
- [5] #####, ##, ## # #. *StackGuard; Automatic Adaptive Detec-
tion and Prevention of Buffer-Overflow Attacks*.
- [6] ##### # # #. *strcpy and strcat -- consistent, safe string copy and
concatenation..*

