・メモリプロファイル http://namazu.org/~satoru/misc/c.html#memory-profilers
・dmalloc document http://dmalloc.com/docs/5.4.2/online/

■ Bash 用コマンド

・function dmalloc { eval `command dmalloc -b $*`; }
・dmalloc -l log -i 100 low
・CFLAGS="-ggdb -DDMALLOC -DUSE_DMALLOC" LIBS=-ldmalloc ./configure

■ Dmalloc with GDB

・add "function dmalloc { eval `command dmalloc -b $*`; }" to .bashrc
・add following commands to .gdbinit

```
define dmalloc
  #
  #echo Enter dmalloc options:
  #shell read arg; dmalloc -g $arg > /tmp/dmalloc-gdb
  #
  shell dmalloc -g -l log -i 100 low > /tmp/dmalloc-gdb
  source /tmp/dmalloc-gdb
  shell rm -f /tmp/dmalloc-gdb
  show env DMALLOC_OPTIONS
  # the following will not work if no symboltable is loaded, but that
  # doesn't matter.
  break dmalloc_error
end
```

・run the GDB, type "dmalloc"

■ How to compile for debug

・CFLAGS="-Wall -ggdb -DDMALLOC -DUSE_DMALLOC" LIBS="-ldmalloc" ./configure
・make

■ gint_cmp_shuffle を使うと SegV になる。

・比較関数 gint_cmp だと正常に動く。

```
#include <glib.h>
#define A 16

gint gint_cmp(const gint *a, const gint *b, gpointer user_data)
{
  if (*a < *b)
    return (-1);
  else if (*a > *b)
    return (1);
  return (0);
}

gint gint_cmp_shuffle(const gint *a, const gint *b, gpointer user_data)
{
  return g_random_int_range(-1, 2);
}

int main(int argc, char *argv[])
{
  int i;
  int table[A];
  for (i = 0; i < A; i++) table[i] = i;
  for (i = 0; i < 128; i++) {
```

```
    g_print("%d\n", i);
    //g_qsort_with_data(table, A, sizeof(int), (GCompareDataFunc)gint_cmp, NULL);
    g_qsort_with_data(table, A, sizeof(int), (GCompareDataFunc)gint_cmp_shuffle, NULL);
  }
  return 0;
}
```

・stdlib.h の q_sort ならどちらでも正常に動く．

```
#include <stdlib.h>
#include <glib.h>
#define A 16

int int_cmp(const int *a, const int *b)
{
  if (*a < *b)
    return (-1);
  else if (*a > *b)
    return (1);
  return (0);
}

int int_cmp_shuffle(const int *a, const int *b)
{
  return g_random_int_range(-1, 2);
}

int main(int argc, char *argv[])
{
  int i;
  int table[A];
  for (i = 0; i < A; i++) table[i] = i;
  for (i = 0; i < 128; i++) {
    g_print("%d\n", i);
    //qsort(table, A, sizeof(int), (int(*)(const void*, const void*))int_cmp);
    qsort(table, A, sizeof(int), (int(*)(const void*, const void*)) int_cmp_shuffle);
  }
  return 0;
}
```

・g_qsort_with_data の比較関数はランダムに大小を返してはいけないらしい。

http://bugzilla.gnome.org/show_bug.cgi?id=313127

・stdlib.h の qsort が動くのはどうしてだろう。

・タグジャンプは M-. (M-x find-tag)
・元に戻るには M-* (M-x pop-tag-mark)

・Ruby に関するメモに "Ruby: " という目印をつけておけば Emacs で M-x occur を実行して Ruby: を検索すれば Ruby 関係のメモをさっと一覧表示できる。

■ XML to HTML transformation

・OpenSP インストール ,OpenJade インストール ,DSSSL,DTD インストール ,catalog 編集 , 環境変数 SGML_CATALOG_FILES を export.
・openjade -t xml -i html -d /usr/share/sgml/dsssl/docbook/html/ldp.dsl#html /usr/share/sgml/dsssl/docbook/dtds/decls/xml.dcl rh.xml 2> log
・単一 HTML 用コマンド
・openjade -t xml -i html -V nochunks -d /usr/share/sgml/dsssl/docbook/html/ldp.dsl#html /usr/share/sgml/dsssl/docbook/dtds/decls/xml.dcl rh.xml > output.html 2> log
・Makefile

```
   FLAGS=-t      xml      -i      html      -d      /usr/share/sgml/dsssl/docbook/html/ldp.dsl\#html
/usr/share/sgml/dsssl/docbook/dtds/decls/xml.dcl
 JADE=openjade
 default: rh.html
         @echo Completed.

 clean:
         @rm -f *~
         @rm -f *.html

 rh.html: rh.xml
         @$(JADE) $(FLAGS) rh.xml 2> log
```