#

# Copyright (c) 1994 by Xerox Corporation. All rights reserved.

# Copyright (c) 1996 by Silicon Graphics. All rights reserved.

# Copyright (c) 1998 by Fergus Henderson. All rights reserved.

# Copyright (c) 2000-2010 by Hewlett-Packard Company. All rights reserved.

##

# THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED

# OR IMPLIED. ANY USE IS AT YOUR OWN RISK.

##

# Permission is hereby granted to use or copy this program

# for any purpose, provided the above notices are retained on all copies.

# Permission to modify the code and to distribute modified code is granted,

# provided the above notices are retained, and a notice that the code was

# modified is included with the above copyright notice.

##

#

# get cmake and run:

# cmake -G "Visual Studio 8 2005"

# in the same dir as this file

# this will generate gc.sln

#

SET(CMAKE\_LEGACY\_CYGWIN\_WIN32 0) # Remove when CMake >= 2.8.4 is required

PROJECT(gc)

CMAKE\_MINIMUM\_REQUIRED(VERSION 2.6)

ADD\_DEFINITIONS("-D\_CRT\_SECURE\_NO\_DEPRECATE

-DALL\_INTERIOR\_POINTERS")

IF(APPLE)

IF("${CMAKE\_OSX\_ARCHITECTURES}" STREQUAL "")

SET(CMAKE\_OSX\_ARCHITECTURES "ppc;i386;x86\_64" CACHE STRING "Build architectures for Mac OS X" FORCE)

ENDIF()

ENDIF(APPLE)

#LIBATOMIC #TODO

#ADD\_LIBRARY(atomic\_ops STATIC )

#SET\_TARGET\_PROPERTIES(atomic\_ops PROPERTIES COMPILE\_FLAGS -DNO\_DEBUGGING)

#LIBGC

INCLUDE\_DIRECTORIES(include)

INCLUDE\_DIRECTORIES(libatomic\_ops/src)

SET(SRC alloc.c reclaim.c allchblk.c misc.c mach\_dep.c os\_dep.c

mark\_rts.c headers.c mark.c obj\_map.c blacklst.c finalize.c

new\_hblk.c dbg\_mlc.c malloc.c stubborn.c dyn\_load.c

typd\_mlc.c ptr\_chck.c gc\_cpp.cc mallocx.c checksums.c

thread\_local\_alloc.c)

SET(LIBS)

OPTION(enable\_threads "TODO" NO)

IF(enable\_threads)

FIND\_PACKAGE(Threads REQUIRED)

MESSAGE("Thread Model: ${CMAKE\_THREAD\_LIBS\_INIT}" )

INCLUDE\_DIRECTORIES(${Threads\_INCLUDE\_DIR})

SET(LIBS ${LIBS} ${Threads\_LIBRARIES})

ENDIF(enable\_threads)

OPTION(enable\_parallel\_mark "Parallelize marking and free list construction" NO)

#IF(Threads\_FOUND)

# ADD\_DEFINITIONS("")

#ELSE

# MESSAGE("Parallel mark requires enable\_threads ON" )

#ENDIF(Threads\_FOUND)

IF(enable\_parallel\_mark)

ENDIF(enable\_parallel\_mark)

OPTION(enable\_cplusplus "install C++ support" NO)

SET(\_HOST ${CMAKE\_HOST\_SYSTEM\_PROCESSOR}--${CMAKE\_SYSTEM}) #FIXME missing the vendor field.Use lowercase

STRING(TOLOWER ${\_HOST} HOST)

MESSAGE("HOST = ${HOST}")

#Thread Detection. Relying on cmake for lib an includes.

#TODO check cmake detection

IF(CMAKE\_USE\_PTHREADS\_INIT)

SET(SRC ${SRC} pthread\_start.c pthread\_support.c pthread\_stop\_world.c)

IF( HOST MATCHES x86-.\*-linux.\*|ia64-.\*-linux.\*|i586-.\*-linux.\*|i686-.\*-linux.\*|x86\_64-.\*-linux.\*|alpha-.\*-linux.\*|sparc.\*-.\*-linux.\*)

ADD\_DEFINITIONS("-DGC\_LINUX\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ENDIF()

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

MESSAGE("Explicit GC\_INIT() calls may be required.")

ENDIF()

IF ( HOST MATCHES .\*-.\*-linux.\*)

ADD\_DEFINITIONS("-DGC\_LINUX\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

ENDIF()

IF ( HOST MATCHES .\*-.\*-aix.\*)

ADD\_DEFINITIONS("-DGC\_AIX\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

ENDIF()

IF ( HOST MATCHES .\*-.\*-hpux11.\*)

MESSAGE("Only HP/UX 11 POSIX threads are supported.")

ADD\_DEFINITIONS("-DGC\_HPUX\_THREADS")

ADD\_DEFINITIONS("-D\_POSIX\_C\_SOURCE=199506L") #TODO test -DVAR=value. Alternative is COMPILE\_DEFINITIONS property

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ENDIF()

MESSAGE("Explicit GC\_INIT() calls may be required.")

ADD\_DEFINITIONS("-D\_REENTRANT") #TODO

ENDIF()

IF ( HOST MATCHES .\*-.\*-hpux10.\*)

MESSAGE("Only HP/UX 11 POSIX threads are supported.")

ENDIF()

IF ( HOST MATCHES .\*-.\*-openbsd.\*)

ADD\_DEFINITIONS("-DGC\_OPENBSD\_THREADS")

ENDIF()

IF ( HOST MATCHES .\*-.\*-freebsd.\*)

MESSAGE("FreeBSD does not yet fully support threads with Boehm GC.")

ADD\_DEFINITIONS("-DGC\_FREEBSD\_THREADS")

ENDIF()

IF ( HOST MATCHES .\*-.\*-kfreebsd.\*-gnu)

ADD\_DEFINITIONS("-DGC\_FREEBSD\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ENDIF()

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

ADD\_DEFINITIONS("-DUSE\_COMPILER\_TLS")

ENDIF()

IF ( HOST MATCHES .\*-.\*-gnu.\*)

ADD\_DEFINITIONS("-DGC\_GNU\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

ENDIF()

IF ( HOST MATCHES .\*-.\*-netbsd.\*)

MESSAGE("Only on NetBSD 2.0 or later.")

ADD\_DEFINITIONS("-DGC\_NETBSD\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

ADD\_DEFINITIONS("-D\_PTHREADS")

ENDIF()

IF ( HOST MATCHES .\*-.\*-solaris.\*)

ADD\_DEFINITIONS("-DGC\_SOLARIS\_THREADS")

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

#TODO

# if test "$GCC" != yes; then

# CFLAGS="$CFLAGS -O"

# need\_atomic\_ops\_asm=true

# fi

ENDIF()

IF ( HOST MATCHES .\*-.\*-irix.\*)

ADD\_DEFINITIONS("-DGC\_IRIX\_THREADS")

ENDIF()

IF ( HOST MATCHES .\*-.\*-cygwin.\*)

ADD\_DEFINITIONS("-DGC\_THREADS")

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ENDIF()

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

#TODO

# win32\_threads=true

ENDIF()

IF ( HOST MATCHES .\*-.\*-darwin.\*)

ADD\_DEFINITIONS("-DGC\_DARWIN\_THREADS")

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

MESSAGE("Explicit GC\_INIT() calls may be required.")

SET(SRC ${SRC} darwin\_stop\_world.c)

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ENDIF()

#TODO

#darwin\_threads=true

ENDIF()

IF ( HOST MATCHES .\*-.\*-osf\*)

ADD\_DEFINITIONS("-DGC\_OSF1\_THREADS")

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

MESSAGE("Explicit GC\_INIT() calls may be required.")

# May want to enable it in other cases, too.

# Measurements haven't yet been done.

ENDIF()

ENDIF()

IF ( HOST MATCHES .\*-.\*-linux.\*)

ADD\_DEFINITIONS("-DGC\_LINUX\_THREADS")

ADD\_DEFINITIONS("-D\_REENTRANT")

ENDIF()

ENDIF(CMAKE\_USE\_PTHREADS\_INIT)

IF(CMAKE\_USE\_WIN32\_THREADS\_INIT)

ADD\_DEFINITIONS("-DGC\_THREADS")

#win32\_threads=true TODO

IF (${enable\_parallel\_mark})

ADD\_DEFINITIONS("-DPARALLEL\_MARK")

ADD\_DEFINITIONS("-DTHREAD\_LOCAL\_ALLOC")

ENDIF()

ADD\_DEFINITIONS("-DEMPTY\_GETENV\_RESULTS") #TODO test

SET(SRC ${SRC} win32\_threads.c)

ENDIF(CMAKE\_USE\_WIN32\_THREADS\_INIT)

OPTION(enable\_gcj\_support "Support for gcj" NO)

IF(enable\_gcj\_support)

ADD\_DEFINITIONS("-DGC\_GCJ\_SUPPORT")

ENDIF(enable\_gcj\_support)

ADD\_LIBRARY( gc-lib STATIC ${SRC})

SET\_TARGET\_PROPERTIES(gc-lib PROPERTIES

COMPILE\_DEFINITIONS GC\_NOT\_DLL)

#TODO TARGET\_LINK\_LIBRARIES(... ... ${LIBS})

ADD\_LIBRARY( gcmt-lib STATIC ${SRC})

SET\_TARGET\_PROPERTIES(gcmt-lib PROPERTIES

COMPILE\_DEFINITIONS GC\_NOT\_DLL)

ADD\_LIBRARY( gcmt-dll SHARED ${SRC})

IF(WIN32)

ADD\_EXECUTABLE(cord cord/cordbscs.c cord/cordxtra.c

cord/tests/de.c cord/tests/de\_win.c)

SET\_TARGET\_PROPERTIES(cord PROPERTIES WIN32\_EXECUTABLE TRUE)

SET\_TARGET\_PROPERTIES(cord PROPERTIES

COMPILE\_DEFINITIONS GC\_NOT\_DLL)

TARGET\_LINK\_LIBRARIES(cord gc-lib)

TARGET\_LINK\_LIBRARIES(cord gdi32)

ENDIF(WIN32)

ADD\_SUBDIRECTORY(tests)