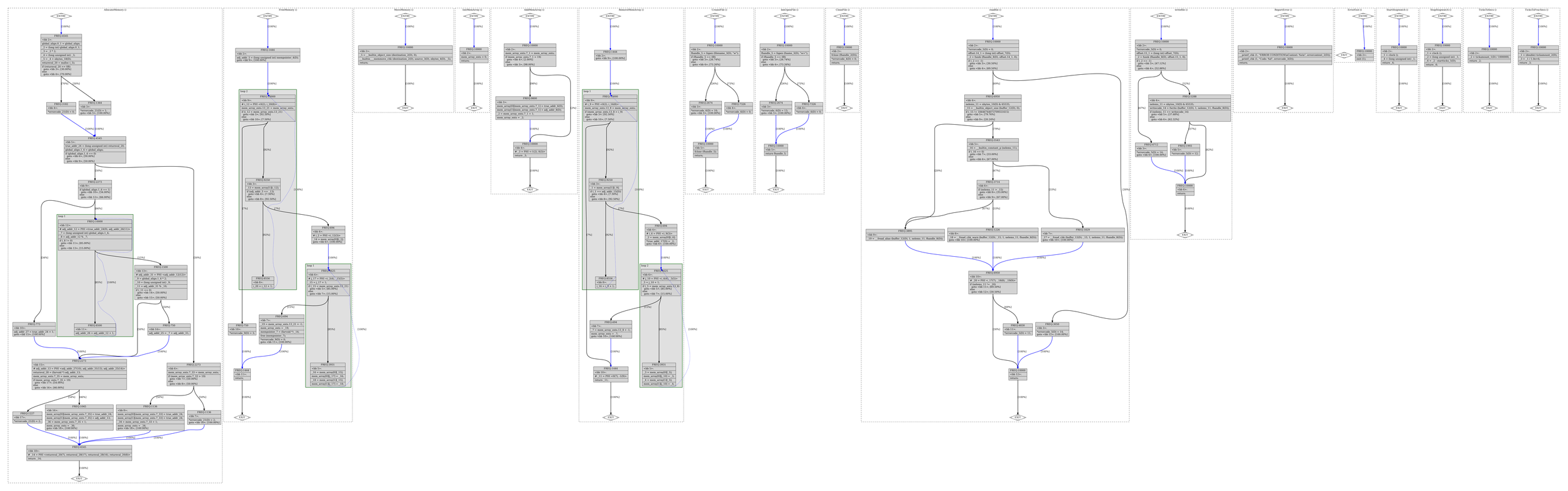
AllocateMemory ()
ENTRY
100%
FREQ:4545
%bb 2=:
global_align.0_1 = global_align;
_2 = (long int) global_align.0_1;
_3 = _2 & 7;
_4 = (long unsigned int) _3;
_5 = _4 + sbytes_32(0);
_6 = malloc (_5);
returntemp_29 = malloc (_5);
if (returntemp_29 == 0B)
goto %bb 3=; [30.00%]
else
goto %bb 4=; [70.00%]
70%
30%
FREQ:1542
%bb 3=:
*errorcode_31(0) = 3;
goto %bb 5=; [100.00%]
100%
FREQ:1364
%bb 4=:
*errorcode_31(0) = 0;
goto %bb 5=; [100.00%]
100%
FREQ:4545
%bb 5=:
true_addr_24 = (long unsigned int) returntemp_29;
global_align.1_6 = global_align;
if (global_align.1_6 == 0)
goto %bb 6=; [50.00%]
else
goto %bb 9=; [50.00%]
50%
50%
FREQ:2273
%bb 6=:
if (global_align.1_6 == 1)
goto %bb 10=; [48.00%]
else
goto %bb 12=; [52.00%]
52%
48%
loop 1
FREQ:19000
%bb 12=:
# adj_addr_11 = PHI <true_addr_24(6), adj_addr_26(11)>
_7 = (long unsigned int) global_align.1_6;
_8 = adj_addr_11 % _7;
if (_8 != 0)
goto %bb 11=; [85.00%]
else
goto %bb 13=; [15.00%]
85%
15%
FREQ:1500
%bb 13=:
# adj_addr_31 = PHI <adj_addr_12(12)>
_9 = global_align.1_6 * 2;
_10 = (long unsigned int) _9;
_11 = adj_addr_31 % _10;
if (_11 == 0)
goto %bb 14=; [50.00%]
else
goto %bb 15=; [50.00%]
50%
50%
FREQ:8560
%bb 11=:
adj_addr_26 = adj_addr_12 + 1;
FREQ:750
%bb 14=:
adj_addr_25 = _7 + adj_addr_31;
FREQ:773
%bb 10=:
if (adj_addr_31 == true_addr_24 + 1)
goto %bb 15=; [100.00%]
100%
FREQ:2273
%bb 15=:
# adj_addr_13 = PHI <adj_addr_27(10), adj_addr_31(13), adj_addr_25(14)>
returntemp_28 = (farvoid *) adj_addr_13;
_12 = _4 + 8;
mem_array_extn.7_35 = mem_array_extn;
if (mem_array_extn.7_35 > 19)
goto %bb 16=; [66.00%]
else
goto %bb 18=; [34.00%]
66%
34%
FREQ:2273
%bb 9=:
mem_array_extn.7_33 = mem_array_extn;
if (mem_array_extn.7_33 > 19)
goto %bb 7=; [50.00%]
else
goto %bb 8=; [50.00%]
50%
50%
FREQ:1041
%bb 16=:
mem_array.0[mem_array_extn.7_35] = true_addr_24;
mem_array1.1[mem_array_extn.7_35] = adj_addr_13;
_26 = mem_array_extn.7_35 + 1;
mem_array_extn = _26;
goto %bb 18=; [100.00%]
100%
FREQ:1136
%bb 8=:
mem_array.0[mem_array_extn.7_33] = true_addr_24;
mem_array1.1[mem_array_extn.7_33] = true_addr_24;
_25 = mem_array_extn.7_33 + 1;
mem_array_extn = _25;
goto %bb 18=; [100.00%]
100%
FREQ:1136
%bb 7=:
*errorcode_32(0) = 0;
goto %bb 18=; [100.00%]
100%
FREQ:4185
%bb 18=:
_34 = PHI <returntemp_28(7), returntemp_28(17), returntemp_28(16), returntemp_29(8)>
return _34;
100%
EXIT
FreeMemory ()
ENTRY
100%
FREQ:1444
%bb 2=:
if(ptr_addr_1 == (long unsigned int) mempointer_4(0))
goto %bb 9=; [100.00%]
100%
loop 2
FREQ:19000
%bb 9=:
# i_12 = PHI <0(2), i_20(8)>
mem_array_extn.17_13 = mem_array_extn;
if (i_12 < mem_array_extn.17_13)
goto %bb 10=; [92.50%]
else
goto %bb 11=; [7.50%]
92%
8%
FREQ:9250
%bb 10=:
_15 = mem_array1.0[i_12];
if (_15 == ptr_addr_1)
goto %bb 6=; [7.00%]
else
goto %bb 8=; [93.00%]
93%
7%
FREQ:8558
%bb 8=:
i_20 = i_12 + 1;
FREQ:694
%bb 6=:
_19 = mem_array_extn.13_21 + -1;
mem_array_extn = _19;
returntemp_7 = (farvoid *) _14;
free (mempointer_7);
*errorcode_3(0) = 0;
goto %bb 13=; [100.00%]
100%
FREQ:750
%bb 11=:
*errorcode_3(0) = 3;
FREQ:1444
%bb 13=:
return;
100%
EXIT
FREQ:694
%bb 4=:
# i_17 = PHI <i_20(5), i_15(3)>
_14 = mem_array1.0[i_17];
if (i_17 < mem_array_extn.13_21)
goto %bb 5=; [92.50%]
else
goto %bb 7=; [7.50%]
93%
7%
FREQ:9225
%bb 5=:
# i_17 = PHI <i_20(5), i_15(3)>
_15 = i_17 + 1;
if (_15 < mem_array_extn.13_21)
goto %bb 5=; [92.50%]
else
goto %bb 7=; [7.50%]
93%
7%
FREQ:2831
%bb 5=:
_16 = mem_array1[8_15];
mem_array1[8_17] = _16;
_18 = mem_array1[8_15];
mem_array1[8_17] = _18;
100%
MoveMemory ()
ENTRY
100%
FREQ:10000
%bb 2=:
_5 = __builtin__object_size (destination_2(D), 0);
__builtin__memmove_chk (destination_2(D), source_3(D), sbytes_4(D), _5);
return;
100%
EXIT
InitMemArray ()
ENTRY
100%
FREQ:10000
%bb 2=:
mem_array_extn = 0;
return;
100%
EXIT
AddMemArray ()
ENTRY
100%
FREQ:10000
%bb 2=:
mem_array_extn.7_1 = mem_array_extn;
if (mem_array_extn.7_1 > 19)
goto %bb 8=; [2.00%]
else
goto %bb 3=; [98.00%]
2%
98%
FREQ:10000
%bb 3=:
mem_array.0[mem_array_extn.7_1] = true_addr_4(D);
mem_array1[mem_array_extn.7_1] = adj_addr_5(D);
_2 = mem_array_extn.7_1 + 1;
mem_array_extn = _2;
100%
FREQ:10000
%bb 4=:
_3 = PHI <+1(2), 0(3)>
return _3;
100%
EXIT
RemoveMemArray ()
ENTRY
100%
FREQ:1444
%bb 2=:
goto %bb 9=; [100.00%]
100%
loop 1
FREQ:19000
%bb 9=:
# i_9 = PHI <0(2), i_5(8)>
mem_array_extn.13_9 = mem_array_extn;
if (mem_array_extn.13_9 > i_9)
goto %bb 10=; [92.50%]
else
goto %bb 11=; [7.50%]
93%
7%
FREQ:9250
%bb 10=:
_7 = mem_array1[i_9];
if (_7 == ptr_addr_1(D))
goto %bb 6=; [7.00%]
else
goto %bb 8=; [93.00%]
93%
7%
FREQ:8558
%bb 8=:
i_5 = i_9 + 1;
FREQ:694
%bb 6=:
_10 = mem_array_extn.13_9 + -1;
mem_array_extn = _10;
goto %bb 13=; [100.00%]
100%
FREQ:1444
%bb 13=:
_11 = PHI <0(7), 1(5)>
return _11;
100%
EXIT
FREQ:694
%bb 4=:
# i_6 = PHI <i_9(3)>
_1 = mem_array1[i_6];
*true_addr_17(D) = _1;
goto %bb 6=; [100.00%]
FREQ:9225
%bb 6=:
# _12 = PHI <i_6(4), _5(5)>
_5 = _12 + 1;
if (_5 < mem_array_extn_12.8)
goto %bb 5=; [92.50%]
else
goto %bb 7=; [7.50%]
93%
7%
FREQ:2831
%bb 5=:
_3 = mem_array1[8_5];
mem_array1[8_12] = _3;
_4 = mem_array1[8_5];
mem_array1[8_5] = _4;
100%
CreateFile ()
ENTRY
100%
FREQ:10000
%bb 2=:
fhandle_3 = fopen (filename_3(D), "w");
if (fhandle_3 == 0B)
goto %bb 3=; [25.74%]
else
goto %bb 4=; [74.26%]
26%
74%
FREQ:2674
%bb 3=:
*errorcode_4(D) = 3;
goto %bb 5=; [100.00%]
100%
FREQ:7326
%bb 4=:
*errorcode_4(D) = 0;
100%
FREQ:10000
%bb 5=:
fclose (fhandle_3);
return;
100%
EXIT
InitOpenFile ()
ENTRY
100%
FREQ:10000
%bb 2=:
fhandle_3 = fopen (filename_3(D), "w+");
if (fhandle_3 == 0B)
goto %bb 3=; [25.74%]
else
goto %bb 4=; [74.26%]
26%
74%
FREQ:2674
%bb 3=:
*errorcode_4(D) = 3;
goto %bb 5=; [100.00%]
100%
FREQ:7326
%bb 4=:
*errorcode_4(D) = 0;
100%
FREQ:10000
%bb 5=:
return fhandle_3;
100%
EXIT
CloseFile ()
ENTRY
100%
FREQ:10000
%bb 2=:
fclose (fhandle_2(D));
*errorcode_4(D) = 0;
return;
100%
EXIT
readfile ()
ENTRY
100%
FREQ:10000
%bb 2=:
*errorcode_5(D) = 0;
offset.14_1 = (long int) offset_7(D);
_2 = fseek (fhandle_8(D), offset.14_1, 0);
if (_2 == -1)
goto %bb 3=; [20.59%]
else
goto %bb 4=; [79.41%]
21%
79%
FREQ:4950
%bb 4=:
nelems_11 = nbytes_10(D) & 65535;
_11 = __builtin_object_size (buffer_12(D), 1);
if (_11 != 18446744073709551615)
goto %bb 5=; [79.76%]
else
goto %bb 6=; [20.24%]
80%
20%
FREQ:5563
%bb 5=:
_16 = __builtin_constant_p (nelems_11);
if (_16 == 0)
goto %bb 7=; [13.00%]
else
goto %bb 6=; [87.00%]
13%
87%
FREQ:3714
%bb 6=:
if (nelems_11 != 1)
goto %bb 8=; [33.00%]
else
goto %bb 9=; [67.00%]
33%
67%
FREQ:2895
%bb 9=:
_19 = __fread_alias (buffer_12(D), 1, nelems_11, fhandle_8(D));
FREQ:3236
%bb 8=:
_18 = __fread_chk_warn (buffer_12(D), 18, 1, nelems_11, fhandle_8(D));
goto %bb 10=; [100.00%]
100%
FREQ:3229
%bb 7=:
_17 = __fread_chk (buffer_12(D), 1, nelems_11, fhandle_8(D));
goto %bb 10=; [100.00%]
100%
FREQ:4950
%bb 10=:
# _20 = PHI <_17(7), _18(8), _19(9)>
if (nelems_11 != _20)
goto %bb 11=; [85.59%]
else
goto %bb 12=; [30.59%]
86%
14%
FREQ:4830
%bb 11=:
*errorcode_5(D) = 11;
FREQ:3050
%bb 12=:
*errorcode_5(D) = 14;
goto %bb 27=; [100.00%]
100%
FREQ:10000
%bb 27=:
return;
100%
EXIT
writefile ()
ENTRY
100%
FREQ:10000
%bb 2=:
*errorcode_5(D) = 0;
offset.13_1 = (long int) offset_7(D);
_2 = fseek (fhandle_8(D), offset.13_1, 0);
if (_2 == -1)
goto %bb 3=; [47.12%]
else
goto %bb 4=; [52.88%]
47%
53%
FREQ:4752
%bb 3=:
*errorcode_5(D) = 14;
goto %bb 6=; [100.00%]
100%
FREQ:1288
%bb 4=:
nelems_11 = nbytes_10(D) & 65535;
writewrite_14 = fwrite (buffer_12(D), 1, nelems_11, fhandle_8(D));
if (nelems_11 == writewrite_14)
goto %bb 5=; [17.48%]
else
goto %bb 6=; [82.52%]
17%
83%
FREQ:1991
%bb 5=:
*errorcode_5(D) = 0;
FREQ:10000
%bb 6=:
return;
100%
EXIT
ReportError ()
ENTRY
100%
FREQ:10000
%bb 3=:
__printf_chk (1, "ERROR CONDITION\nCustomer: %s\n\n", errorcustomer_2(D));
__printf_chk (1, "Code: %d\", errorcode_3(D));
return;
100%
EXIT
ErrorExit ()
ENTRY
100%
FREQ:10000
%bb 2=:
exit (1);
StartSbupwatch ()
ENTRY
100%
FREQ:10000
%bb 2=:
_1 = clock ();
_2 = (long unsigned int) _1;
return _2;
100%
EXIT
StopSbupwatch ()
ENTRY
100%
FREQ:10000
%bb 2=:
_1 = clock ();
_2 = (long unsigned int) _1;
_3 = _2 - startclock_3(D);
return _3;
100%
EXIT
TickToSecs ()
ENTRY
100%
FREQ:10000
%bb 2=:
_1 = (double) tickamount_2(D);
_2 = _1 / 1.0e+6;
return _2;
100%
EXIT
TickToFSecSecs ()
ENTRY
100%
FREQ:10000
%bb 2=:
_1 = (double) tickamount_2(D);
_2 = _1 / 1.0e+9;
return _2;
100%
EXIT