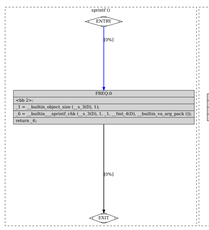
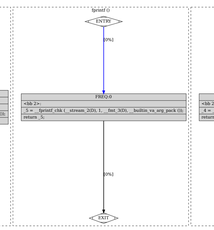
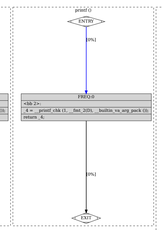
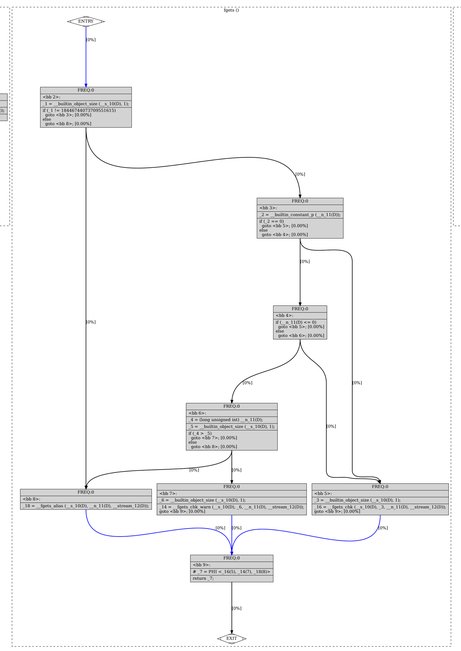

sprintf ()
ENTRY
19%
FREQ:0
<bb 2>:
_1 = __builtin_object_size (_s, 3);
_6 = __builtin__sprintf_chk (_s, 3, _1, __fmt, 42);
__builtin_va_arg_pack ();
return _6;
19%
EXIT

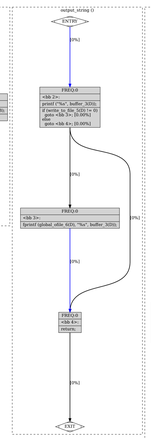

fprintf ()
ENTRY
19%
FREQ:0
<bb 2>:
_5 = __fprintf_chk (__stream, 2CD), 1, __fmt, 3CD), __builtin_va_arg_pack ();
return _5;
19%
EXIT

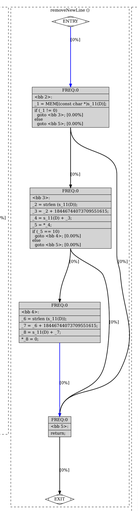

printf ()
ENTRY
19%
FREQ:0
<bb 2>:
_4 = __printf_chk (1, __fmt, 2CD), __builtin_va_arg_pack ();
return _4;
19%
EXIT

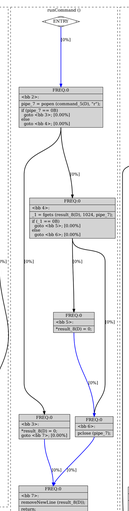

fgets ()
ENTRY
19%
FREQ:0
<bb 2>:
_1 = __builtin_object_size (__s, 1)D);
if (_1 != 18446744073709551615)
goto <bb 3>; [0.00%]
goto <bb 8>; [0.00%]
FREQ:0
<bb 3>:
_2 = __builtin_constant_p (__n, 1)D);
if (_2 == 0)
goto <bb 5>; [0.00%]
goto <bb 4>; [0.00%]
FREQ:0
<bb 4>:
if (__n == 1)D)
goto <bb 5>; [0.00%]
else
goto <bb 6>; [0.00%]
FREQ:0
<bb 6>:
_4 = (long unsigned int) __n_1)D);
_5 = __builtin_object_size (__s, 1)D);
if (_4 > _5)
goto <bb 7>; [0.00%]
else
goto <bb 8>; [0.00%]
FREQ:0
<bb 8>:
_10 = __fgets_alias (__s, 3)D), __n_1)D), __stream, 12)D));
FREQ:0
<bb 7>:
_6 = __builtin_object_size (__s, 0)D), 1);
_14 = __fgets_chk_warn (__s, 3)D), _6, __n_1)D), __stream, 12)D));
FREQ:0
<bb 5>:
_3 = __builtin_object_size (__s, 0)D), 1);
_16 = __fgets_chk (__s, 3)D), _3, __n_1)D), __stream, 12)D));
FREQ:0
<bb 9>:
# _7 = PHI <_16(5), _14(7), _10(8)>
return _7;
19%
EXIT

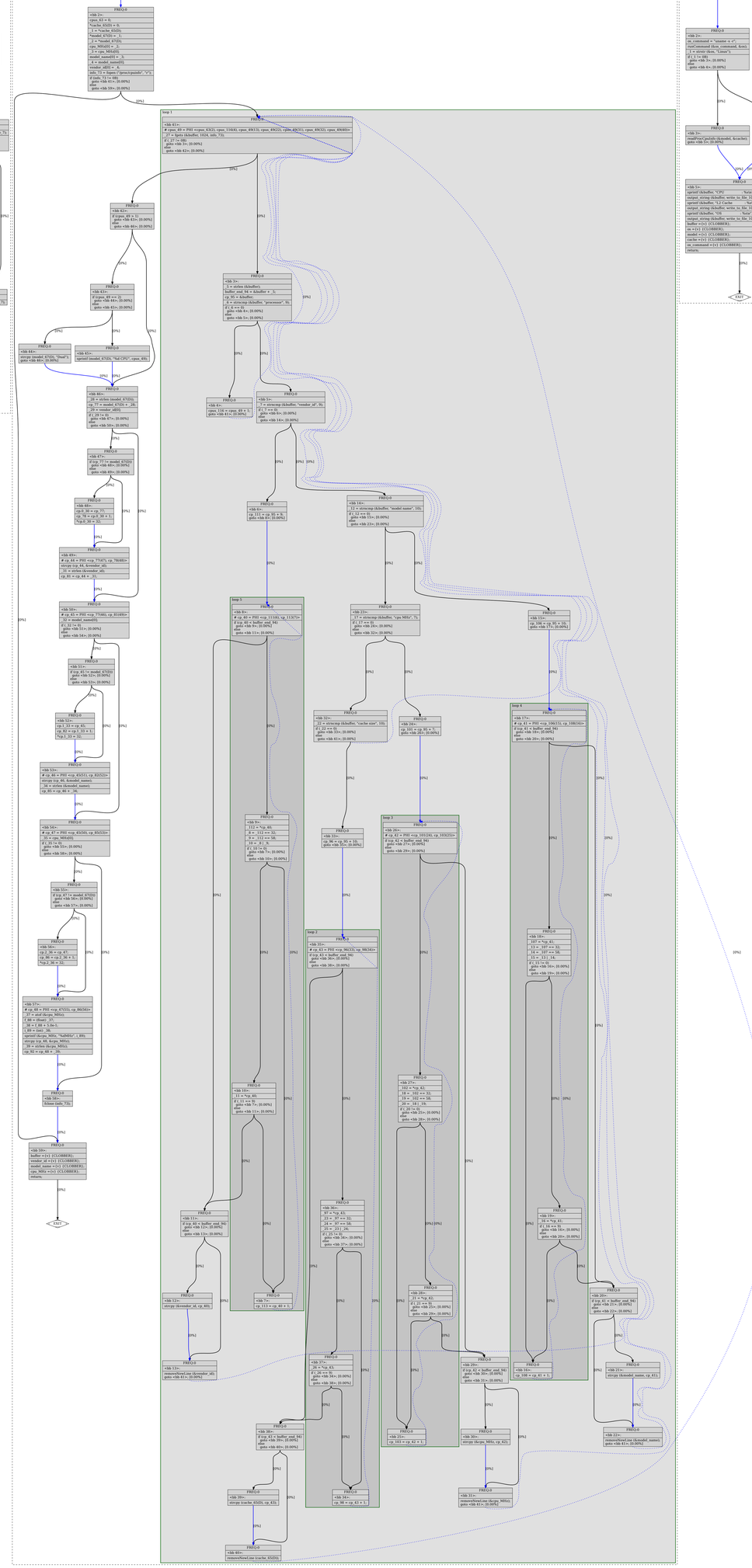
strcpy ()
ENTRY
19%
FREQ:0
<bb 2>:
_1 = __builtin_object_size (__dest, 3CD), 1);
_6 = __builtin___strcpy_chk (__dest, 3CD), __src, 4CD), _1);
return _6;
19%
EXIT

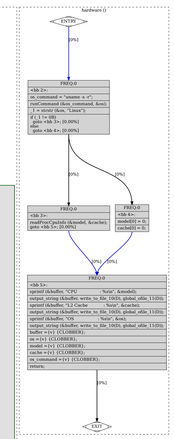
atof ()
ENTRY
19%
FREQ:0
<bb 2>:
_4 = strtod (__nptr, 2CD), 0B);
return _4;
19%
EXIT


output_string ()
ENTRY
19%
FREQ:0
<bb 2>:
printf ("%s", buffer_3CD));
if (strlen_to_file_5CD) != 0)
goto <bb 3>; [0.00%]
goto <bb 4>; [0.00%]
FREQ:0
<bb 3>:
fprintf (global_stfile_6CD), "%s", buffer_3CD));
FREQ:0
<bb 4>:
return;
19%
EXIT


removeNewLine ()
ENTRY
19%
FREQ:0
<bb 2>:
printf ("%s", string_11CD);
if (strlen (string_11CD)) > 0)
goto <bb 3>; [0.00%]
goto <bb 4>; [0.00%]
FREQ:0
<bb 3>:
_2 = strlen (s_11CD);
_3 == _2 + 18446744073709551615;
_4 = s_11CD + _3;
_5 = *_4;
if (_5 == 10)
goto <bb 5>; [0.00%]
else
goto <bb 5>; [0.00%]
FREQ:0
<bb 4>:
_6 = strlen (s_11CD);
_7 == _6 + 18446744073709551615;
_8 = s_11CD + _7;
*_8 = 0;
FREQ:0
<bb 5>:
return;
19%
EXIT


runCommand ()
ENTRY
19%
FREQ:0
<bb 2>:
pipe_7 = popen (command_5CD), "r");
if (pipe_7 == 0B)
goto <bb 3>; [0.00%]
goto <bb 4>; [0.00%]
FREQ:0
<bb 4>:
fread (result_8CD), 1024, pipe_7);
if (_1 == 0B)
goto <bb 5>; [0.00%]
goto <bb 6>; [0.00%]
FREQ:0
<bb 5>:
*result_8CD) = 0;
goto <bb 7>; [0.00%]
FREQ:0
<bb 6>:
pclose (pipe_7);
FREQ:0
<bb 7>:
removeNewLine (result_8CD);
return;
19%
EXIT


readProcCpuInfo ()
ENTRY
19%
FREQ:0
<bb 2>:
cpus_43 = 0;
*cache_65CD) = 0;
_1 = *cache_65CD);
*model_67CD) = _1;
_2 = *model_67CD);
cpu_MHz1D) = _2;
_3 = cpu_MHz1D);
model name1D = _3;
_4 = model_name1D);
vendor_id1D = _4;
info_73 = fopen ("/proc/cpuinfo", "r");
if (info_73 != 0B)
goto <bb 41>; [0.00%]
goto <bb 42>; [0.00%]
loop 1
FREQ:0
<bb 41>:
# cpus_49 = PHI <cpus_43(2), cpus_114(40), cpus_49(13), cpus_49(22), cpus_49(31), cpus_49(32), cpus_49(40)>
_27 = fgets (&buffer, 1024, info_73);
if (_27 != 0B)
goto <bb 43>; [0.00%]
else
goto <bb 42>; [0.00%]
FREQ:0
<bb 43>:
if (cpus_49 == 1)
goto <bb 44>; [0.00%]
else
goto <bb 45>; [0.00%]
FREQ:0
<bb 44>:
strcpy (model_67CD), "Dual");
goto <bb 46>; [0.00%]
FREQ:0
<bb 45>:
sprintf (model, 67CD), "%d CPU", cpus_49);
FREQ:0
<bb 46>:
_28 = strlen (model_67CD));
cp_77 = model_67CD) + _28;
_29 = vendor_id1D);
if (_29 != 0)
goto <bb 47>; [0.00%]
goto <bb 50>; [0.00%]
FREQ:0
<bb 47>:
if (cp_77 != model_67CD))
goto <bb 48>; [0.00%]
else
goto <bb 49>; [0.00%]
FREQ:0
<bb 48>:
cp_8 = cp_77;
cp_78 = cp_8 + 1;
*cp_8 = 32;
FREQ:0
<bb 49>:
# cp_44 = PHI <cp_77(47), cp_78(48)>
strcpy (cp_44, &vendor_id);
_31 = strlen (&vendor_id);
cp_81 = cp_44 + _31;
FREQ:0
<bb 50>:
# cp_45 = PHI <cp_77(46), cp_81(49)>
_32 = model_name1D;
if (_32 != 0)
goto <bb 51>; [0.00%]
goto <bb 54>; [0.00%]
FREQ:0
<bb 51>:
if (cp_45 != model_67CD))
goto <bb 52>; [0.00%]
else
goto <bb 53>; [0.00%]
FREQ:0
<bb 52>:
cp_1_33 = cp_45;
cp_82 = cp_1_33 + 1;
*cp_1_33 = 32;
FREQ:0
<bb 53>:
# cp_46 = PHI <cp_45(51), cp_82(52)>
strcpy (cp_46, &model_name);
_34 = strlen (&model_name);
cp_85 = cp_46 + _34;
FREQ:0
<bb 54>:
# cp_47 = PHI <cp_45(50), cp_85(53)>
_35 = cpu_MHz1D);
if (_35 != 0)
goto <bb 55>; [0.00%]
goto <bb 58>; [0.00%]
FREQ:0
<bb 55>:
if (cp_47 != model_67CD))
goto <bb 56>; [0.00%]
else
goto <bb 57>; [0.00%]
FREQ:0
<bb 56>:
cp_2_36 = cp_47;
cp_86 = cp_2_36 + 1;
*cp_2_36 = 32;
FREQ:0
<bb 57>:
# cp_48 = PHI <cp_47(55), cp_86(56)>
_37 = atof (&cpu_MHz);
f_88 = (float) _37;
_38 = f_88 * 1.0e+3;
_39 = (int) _38;
sprintf (&cpu_MHz, "%dMHz", i_89);
strcpy (cp_48, &cpu_MHz);
_39 = strlen (&cpu_MHz);
cp_92 = cp_48 + _39;
FREQ:0
<bb 58>:
fclose (info_73);
FREQ:0
<bb 59>:
buffer = {v} {CLOBBER};
model = {v} {CLOBBER};
model_name = {v} {CLOBBER};
cpu_MHz = {v} {CLOBBER};
return;
EXIT
loop 5
FREQ:0
<bb 8>:
# cp_48 = PHI <cp_111(5), cp_113(7)>
if (cp_48 < buffer_end_94)
goto <bb 9>; [0.00%]
else
goto <bb 11>; [0.00%]
FREQ:0
<bb 9>:
_112 = *cp_48;
_6 = _112 == 13;
_9 = _112 == 58;
_10 = _6 | _9;
if (_10 != 0)
goto <bb 7>; [0.00%]
goto <bb 10>; [0.00%]
FREQ:0
<bb 10>:
_11 = cp_48;
_12 = _11 == 10;
if (_12 != 0)
goto <bb 11>; [0.00%]
goto <bb 7>; [0.00%]
FREQ:0
<bb 7>:
cp_113 = cp_48 + 1;
loop 2
FREQ:0
<bb 33>:
# cp_43 = PHI <cp_96(33), cp_98(34)>
if (cp_43 < buffer_end_94)
goto <bb 34>; [0.00%]
else
goto <bb 38>; [0.00%]
FREQ:0
<bb 34>:
_96 = *cp_43;
_97 == _96;
_24 == _97 == 58;
_25 == _23 | _24;
if (_25 != 0)
goto <bb 36>; [0.00%]
goto <bb 37>; [0.00%]
FREQ:0
<bb 36>:
_97 = *cp_43;
_23 == _97 == 32;
_24 == _97 == 58;
_25 == _23 | _24;
if (_25 != 0)
goto <bb 34>; [0.00%]
goto <bb 37>; [0.00%]
FREQ:0
<bb 37>:
_26 = *cp_43;
if (_26 == 9)
goto <bb 34>; [0.00%]
else
goto <bb 38>; [0.00%]
FREQ:0
<bb 38>:
if (cp_43 < buffer_end_94)
goto <bb 39>; [0.00%]
goto <bb 40>; [0.00%]
FREQ:0
<bb 39>:
strcpy (cache_65D9, cp_43);
FREQ:0
<bb 40>:
removeNewLine (cache_65D3);
FREQ:0
<bb 11>:
if (cp_48 < buffer_end_94)
goto <bb 12>; [0.00%]
goto <bb 13>; [0.00%]
FREQ:0
<bb 12>:
strcpy (&vendor_id, cp_48);
FREQ:0
<bb 13>:
removeNewLine (&vendor_id);
goto <bb 41>; [0.00%]
FREQ:0
<bb 42>:
fclose (info_73);
loop 3
FREQ:0
<bb 26>:
# cp_42 = PHI <cp_101(24), cp_103(25)>
if (cp_42 < buffer_end_94)
goto <bb 27>; [0.00%]
else
goto <bb 29>; [0.00%]
FREQ:0
<bb 27>:
_102 = *cp_42;
_18 == _102 == 32;
_19 == _102 == 58;
_20 == _18 | _19;
if (_20 != 0)
goto <bb 25>; [0.00%]
goto <bb 28>; [0.00%]
FREQ:0
<bb 28>:
_21 = *cp_42;
if (_21 == 9)
goto <bb 25>; [0.00%]
goto <bb 29>; [0.00%]
FREQ:0
<bb 25>:
cp_103 = cp_42 + 1;
FREQ:0
<bb 29>:
if (cp_42 < buffer_end_94)
goto <bb 30>; [0.00%]
goto <bb 31>; [0.00%]
FREQ:0
<bb 30>:
strcpy (&cpu_MHz, cp_42);
FREQ:0
<bb 31>:
removeNewLine (&cpu_MHz);
goto <bb 41>; [0.00%]
FREQ:0
<bb 6>:
cpus_114 = cpus_49 + 1;
goto <bb 41>; [0.00%]
FREQ:0
<bb 5>:
_7 = strncmp (&buffer, "vendor_id", 3);
if (_7 == 0)
goto <bb 6>; [0.00%]
goto <bb 14>; [0.00%]
FREQ:0
<bb 14>:
_12 = strncmp (&buffer, "model name", 10);
if (_12 == 0)
goto <bb 15>; [0.00%]
goto <bb 23>; [0.00%]
FREQ:0
<bb 15>:
cp_111 = cp_93 + 9;
goto <bb 8>; [0.00%]
FREQ:0
<bb 23>:
_17 = strncmp (&buffer, "cpu MHz", 7);
if (_17 == 0)
goto <bb 24>; [0.00%]
goto <bb 32>; [0.00%]
FREQ:0
<bb 24>:
cp_101 = cp_93 + 7;
goto <bb 26>; [0.00%]
FREQ:0
<bb 32>:
_22 = strncmp (&buffer, "cache size", 10);
if (_22 == 0)
goto <bb 41>; [0.00%]
else
goto <bb 41>; [0.00%]
loop 4
FREQ:0
<bb 17>:
# cp_41 = PHI <cp_106(15), cp_108(16)>
if (cp_41 < buffer_end_94)
goto <bb 18>; [0.00%]
goto <bb 20>; [0.00%]
FREQ:0
<bb 18>:
_107 = *cp_41;
_13 == _107 == 32;
_14 == _107 == 58;
_15 == _13 | _14;
if (_15 != 0)
goto <bb 16>; [0.00%]
goto <bb 19>; [0.00%]
FREQ:0
<bb 19>:
_16 = *cp_41;
if (_16 == 9)
goto <bb 16>; [0.00%]
goto <bb 20>; [0.00%]
FREQ:0
<bb 16>:
cp_108 = cp_41 + 1;
FREQ:0
<bb 20>:
if (cp_41 < buffer_end_94)
goto <bb 21>; [0.00%]
goto <bb 22>; [0.00%]
FREQ:0
<bb 21>:
strcpy (&model_name, cp_41);
FREQ:0
<bb 22>:
removeNewLine (&model_name);
goto <bb 41>; [0.00%]


hardware ()
ENTRY
19%
FREQ:0
<bb 2>:
os_command = "uname -s -r";
runCommand (&os_command, &os);
_1 = strstr (&os, "Linux");
if (_1 != 0B)
goto <bb 3>; [0.00%]
goto <bb 4>; [0.00%]
FREQ:0
<bb 3>:
readProcCpuInfo (&cpuInfo, &cache);
goto <bb 5>; [0.00%]
FREQ:0
<bb 4>:
model[0] = 0;
cache[0] = 0;
FREQ:0
<bb 5>:
sprintf (&buffer, "CPU          : %s\n\n", &model);
output_string (&buffer, write_to_file, 1)D), global_stfile_1)D));
sprintf (&buffer, "L2 Cache     : %s\n\n", &cache);
output_string (&buffer, write_to_file, 1)D), global_stfile_1)D));
sprintf (&buffer, "OS           : %s\n", &os);
output_string (&buffer, write_to_file, 1)D), global_stfile_1)D));
buffer = {v} {CLOBBER};
os = {v} {CLOBBER};
model = {v} {CLOBBER};
cache = {v} {CLOBBER};
os_command = {v} {CLOBBER};
return;
EXIT