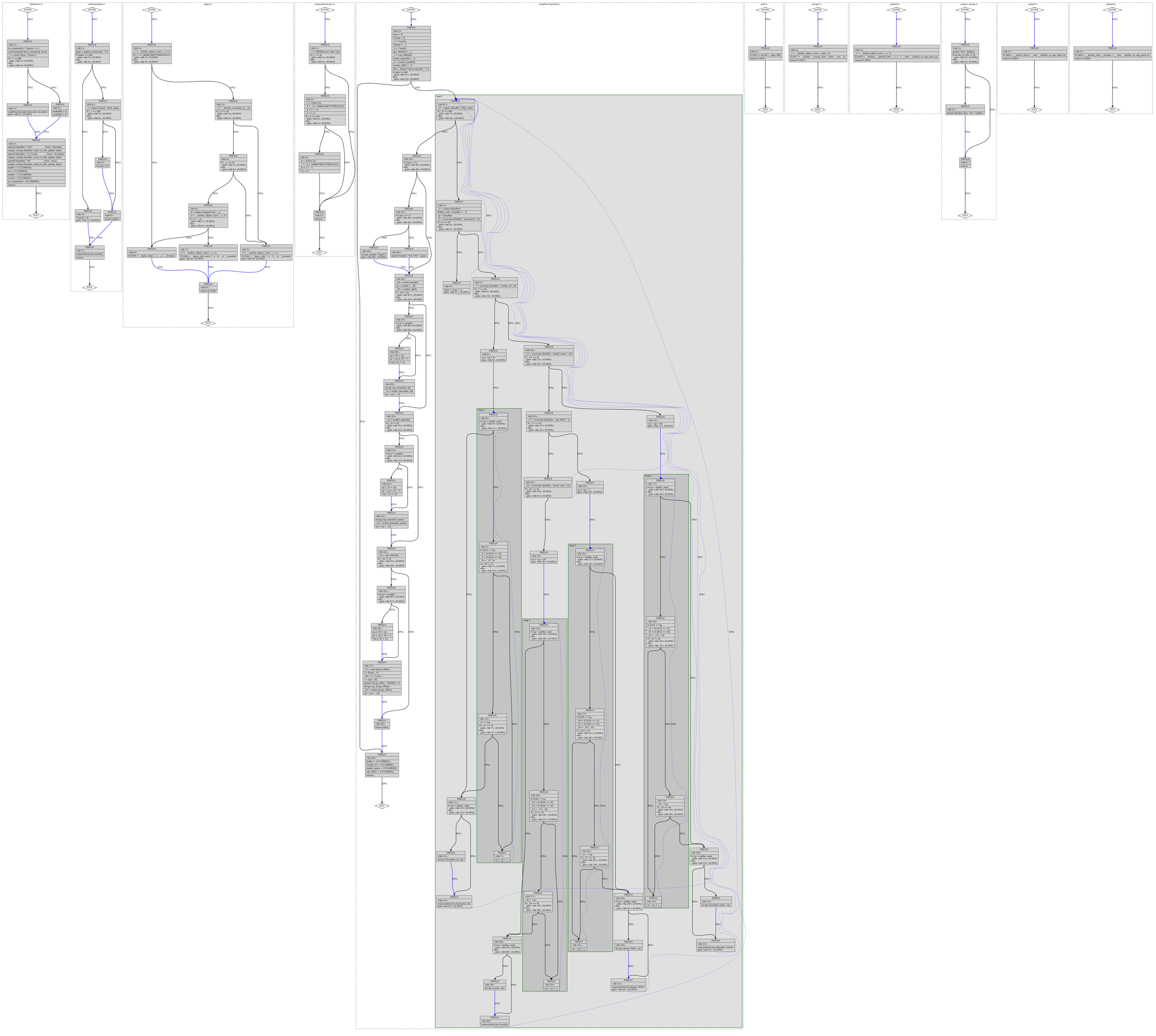
hardware ()
ENTRY
FREQ:0
<bb 2>:
os_command = "uname -s -r";
runCommand (&os_command, &cmd);
_1 = strstr (&cmd, "Linux");
if (_1 != 0B)
goto <bb 3>; [0.00%]
else
goto <bb 4>; [0.00%]
<bb 3>:
readProcCpuInfo (&model, &cache);
goto <bb 5>; [0.00%]
<bb 4>:
model[0] = 0;
cache[0] = 0;
<bb 5>:
sprintf (&buffer, "CPU: %s\n", &model);
output_string (&buffer, write_to_file, global_ofile);
sprintf (&buffer, "L2 Cache : %s\n", &cache);
output_string (&buffer, write_to_file, global_ofile);
sprintf (&buffer, "OS : %s\n", &os);
output_string (&buffer, write_to_file, global_ofile);
buffer = {CLOBBER};
os = {CLOBBER};
model = {CLOBBER};
cache = {CLOBBER};
os_command = {CLOBBER};
return;
EXIT
runCommand ()
ENTRY
<bb 2>:
pipe = popen (command, "r");
if (pipe == 0B)
goto <bb 3>; [0.00%]
else
goto <bb 4>; [0.00%]
<bb 3>:
*result = 0;
goto <bb 7>; [0.00%]
<bb 4>:
_1 = fgets (result, 1024, pipe);
if (_1 == 0B)
goto <bb 5>; [0.00%]
else
goto <bb 6>; [0.00%]
<bb 5>:
*result = 0;
<bb 6>:
pclose (pipe);
<bb 7>:
removeNewLine (result);
return;
EXIT
fgets ()
ENTRY
<bb 2>:
_1 = __builtin_object_size (__s, 1);
if (_1 != 18446744073709551615)
goto <bb 3>; [0.00%]
else
goto <bb 8>; [0.00%]
<bb 3>:
_2 = __builtin_constant_p (__n);
if (_2 == 0)
goto <bb 5>; [0.00%]
else
goto <bb 4>; [0.00%]
<bb 4>:
if (__n <= 0)
goto <bb 5>; [0.00%]
else
goto <bb 6>; [0.00%]
<bb 5>:
_3 = __builtin_object_size (__s, 1);
D.3394 = __fgets_chk (__s, _3, __n, __stream);
goto <bb 9>; [0.00%]
<bb 6>:
_4 = (long unsigned int) __n;
_5 = __builtin_object_size (__s, 1);
if (_4 > _5)
goto <bb 7>; [0.00%]
else
goto <bb 8>; [0.00%]
<bb 7>:
_6 = __builtin_object_size (__s, 1);
D.3394 = __fgets_chk_warn (__s, _6, __n, __stream);
goto <bb 9>; [0.00%]
<bb 8>:
D.3394 = __fgets_alias (__s, __n, __stream);
<bb 9>:
return D.3394;
EXIT
removeNewLine ()
ENTRY
<bb 2>:
_1 = MEM[(const char *)s];
if (_1 != 0)
goto <bb 3>; [0.00%]
else
goto <bb 5>; [0.00%]
<bb 3>:
_2 = strlen (s);
_3 = _2 + 18446744073709551613;
_4 = s + _3;
_5 = *_4;
if (_5 == 10)
goto <bb 4>; [0.00%]
else
goto <bb 5>; [0.00%]
<bb 4>:
_6 = strlen (s);
_7 = _6 + 18446744073709551615;
_8 = s + _7;
*_8 = 0;
<bb 5>:
return;
EXIT
readProcCpuInfo ()
ENTRY
FREQ:0
<bb 2>:
cpus = 0;
*cache = 0;
_1 = *cache;
*model = _1;
_2 = *model;
cpu_MHz[0] = _2;
_3 = cpu_MHz[0];
model_name[0] = _3;
_4 = model_name[0];
vendor_id[0] = _4;
info = fopen ("/proc/cpuinfo", "r");
if (info != 0B)
goto <bb 41>; [0.00%]
else
goto <bb 59>; [0.00%]
loop 1
FREQ:0
<bb 41>:
_27 = fgets (&buffer, 1924, info);
if (_27 != 0B)
goto <bb 2>; [0.00%]
else
goto <bb 42>; [0.00%]
<bb 2>:
_5 = strlen (&buffer);
buffer_end = &buffer + _5;
cp = &buffer;
_6 = strncmp (&buffer, "processor", 9);
if (_6 == 0)
goto <bb 4>; [0.00%]
else
goto <bb 5>; [0.00%]
<bb 4>:
cpus += cpus + 1;
goto <bb 41>; [0.00%]
<bb 5>:
_7 = strncmp (&buffer, "vendor_id", 9);
if (_7 == 0)
goto <bb 6>; [0.00%]
else
goto <bb 14>; [0.00%]
<bb 6>:
cp = cp + 9;
goto <bb 8>; [0.00%]
<bb 14>:
_12 = strncmp (&buffer, "model name", 10);
if (_12 == 0)
goto <bb 15>; [0.00%]
else
goto <bb 23>; [0.00%]
<bb 15>:
cp = cp + 10;
goto <bb 17>; [0.00%]
<bb 23>:
_17 = strncmp (&buffer, "cpu MHz", 7);
if (_17 == 0)
goto <bb 24>; [0.00%]
else
goto <bb 32>; [0.00%]
<bb 24>:
cp = cp + 7;
goto <bb 26>; [0.00%]
<bb 32>:
_22 = strncmp (&buffer, "cache size", 10);
if (_22 == 0)
goto <bb 33>; [0.00%]
else
goto <bb 41>; [0.00%]
<bb 33>:
cp = cp + 10;
goto <bb 35>; [0.00%]
loop 2
FREQ:0
<bb 35>:
if (cp < buffer_end)
goto <bb 36>; [0.00%]
else
goto <bb 38>; [0.00%]
<bb 36>:
D.3434 = *cp;
_23 = D.3434 == 32;
_24 = D.3434 == 58;
_25 = _23 | _24;
if (_25 != 0)
goto <bb 34>; [0.00%]
else
goto <bb 37>; [0.00%]
<bb 34>:
cp = cp + 1;
<bb 37>:
_26 = *cp;
if (_26 == 9)
goto <bb 34>; [0.00%]
else
goto <bb 38>; [0.00%]
<bb 38>:
if (cp < buffer_end)
goto <bb 39>; [0.00%]
else
goto <bb 40>; [0.00%]
<bb 39>:
strcpy (cache, cp);
<bb 40>:
removeNewLine (cache);
loop 3
FREQ:0
<bb 26>:
if (cp < buffer_end)
goto <bb 27>; [0.00%]
else
goto <bb 29>; [0.00%]
<bb 27>:
D.3426 = *cp;
_18 = D.3426 == 32;
_19 = D.3426 == 58;
_20 = _18 | _19;
if (_20 != 0)
goto <bb 25>; [0.00%]
else
goto <bb 28>; [0.00%]
<bb 25>:
cp = cp + 1;
<bb 28>:
_21 = *cp;
if (_21 == 9)
goto <bb 25>; [0.00%]
else
goto <bb 29>; [0.00%]
<bb 29>:
if (cp < buffer_end)
goto <bb 30>; [0.00%]
else
goto <bb 31>; [0.00%]
<bb 30>:
strcpy (&cpu_MHz, cp);
<bb 31>:
removeNewLine (&cpu_MHz);
goto <bb 41>; [0.00%]
loop 4
FREQ:0
<bb 17>:
if (cp < buffer_end)
goto <bb 18>; [0.00%]
else
goto <bb 20>; [0.00%]
<bb 18>:
D.3418 = *cp;
_13 = D.3418 == 32;
_14 = D.3418 == 58;
_15 = _13 | _14;
if (_15 != 0)
goto <bb 16>; [0.00%]
else
goto <bb 19>; [0.00%]
<bb 16>:
cp = cp + 1;
<bb 19>:
_16 = *cp;
if (_16 == 9)
goto <bb 16>; [0.00%]
else
goto <bb 20>; [0.00%]
<bb 20>:
if (cp < buffer_end)
goto <bb 21>; [0.00%]
else
goto <bb 22>; [0.00%]
<bb 21>:
strcpy (&model_name, cp);
<bb 22>:
removeNewLine (&model_name);
goto <bb 41>; [0.00%]
<bb 42>:
if (cpus != 1)
goto <bb 43>; [0.00%]
else
goto <bb 46>; [0.00%]
<bb 43>:
if (cpus == 2)
goto <bb 44>; [0.00%]
else
goto <bb 45>; [0.00%]
<bb 44>:
sprintf (model, "%d CPU", cpus);
<bb 45>:
strcpy (model, "Dual");
goto <bb 46>; [0.00%]
<bb 46>:
_28 = strlen (model);
cp = model + _28;
_29 = vendor_id[0];
if (_29 != 0)
goto <bb 47>; [0.00%]
else
goto <bb 50>; [0.00%]
<bb 47>:
if (cp != model)
goto <bb 48>; [0.00%]
else
goto <bb 49>; [0.00%]
<bb 48>:
cp = cp + -1;
<bb 49>:
strcpy (cp, &vendor_id);
_31 = strlen (&vendor_id);
cp = cp + _31;
<bb 50>:
_32 = model_name[0];
if (_32 != 0)
goto <bb 51>; [0.00%]
else
goto <bb 54>; [0.00%]
<bb 51>:
if (cp != model)
goto <bb 52>; [0.00%]
else
goto <bb 53>; [0.00%]
<bb 52>:
cp = cp + -1;
<bb 53>:
strcpy (cp, &model_name);
_34 = strlen (&model_name);
cp = cp + _34;
<bb 54>:
_35 = cpu_MHz[0];
if (_35 != 0)
goto <bb 55>; [0.00%]
else
goto <bb 58>; [0.00%]
<bb 55>:
if (cp != model)
goto <bb 56>; [0.00%]
else
goto <bb 57>; [0.00%]
<bb 56>:
cp = cp + -1;
<bb 57>:
_37 = atof (&cpu_MHz);
f = (float) _37;
_38 = f + 5.0e-1;
i = (int) _38;
sprintf (&cpu_MHz, "%dMHz", i);
strcpy (cp, &cpu_MHz);
_39 = strlen (&cpu_MHz);
cp = cp + _39;
<bb 58>:
fclose (info);
<bb 59>:
buffer = {CLOBBER};
vendor_id = {CLOBBER};
model_name = {CLOBBER};
cpu_MHz = {CLOBBER};
return;
EXIT
atof ()
ENTRY
FREQ:0
<bb 2>:
D.3455 = strtod (__nptr, 0B);
return D.3455;
EXIT
strcpy ()
ENTRY
FREQ:0
<bb 2>:
_1 = __builtin_object_size (__dest, 1);
D.3457 = __builtin___strcpy_chk (__dest, __src, _1);
return D.3457;
EXIT
sprintf ()
ENTRY
FREQ:0
<bb 2>:
_1 = __builtin_object_size (__s, 1);
D.3459 = __builtin___sprintf_chk (__s, 1, _1, __fmt, __builtin_va_arg_pack ());
return D.3459;
EXIT
output_string ()
ENTRY
FREQ:0
<bb 2>:
printf ("%s", buffer);
if (write_to_file != 0)
goto <bb 3>; [0.00%]
else
goto <bb 4>; [0.00%]
<bb 3>:
fprintf (global_ofile, "%s", buffer);
<bb 4>:
return;
EXIT
printf ()
ENTRY
FREQ:0
<bb 2>:
D.3464 = __printf_chk (1, __fmt, __builtin_va_arg_pack ());
return D.3464;
EXIT
fprintf ()
ENTRY
FREQ:0
<bb 2>:
D.3466 = __fprintf_chk (__stream, 1, __fmt, __builtin_va_arg_pack ());
return D.3466;
EXIT