

Lab1 文档

515030910038 张子扬

1. 实现 “%o” 八进制输出
只需要模仿十六进制输出即可，设定 `base = 8`
2. 实现 “%+” 输出
在 `reswitch` 语句中添加 “+” 分支，设置一个输出 “+” 符号的 `flag`，然后在原有的 “%d” 分支中，添加对 `flag` 的判断，在 `goto number` 语句前，将 “+” 添加到输出
3. 实现 “%-” 输出
在 `reswitch` 语句中添加 “-” 分支，设置 `padc = '-'`，在打印数字的函数 `printnum` 中，判断 `padc` 是否为 ‘-’，是则需要先输出数字，再输出对齐的空格，因此改变原有的递归写法，采用迭代的等价写法：先生成原数的逆序数，再使用循环语句打印数，最后循环语句打印空格。
4. 实现 `backtrace` 功能
添加新的命令 `backtrace`，需要完成 `mon_backtrace` 函数。根据给定的格式，打印相应的字符串，注意到打印十六进制数的格式，模仿给出的类似语句，使用 “%08x” 输出。根据指导：为了获得 `stack` 的信息，需要得到 `ebp` 的值，当前的 `ebp` 的值是当前函数的刚开始时 `esp` 的值，因此 `eip` 的值（根据提示它是函数的返回地址的指针的值）处在 `ebp` 的前面一个位置，它的值是 `*(ebp+1)`，另外，还需要输出函数的参数，他们的位置在 `eip` 的值的上面（参数从前到后，它们的位置从低到高），因此参数的值是 `*(ebp+i+2)`，`i` 从 0-4 循环。接下来需要调用 `debuginfo_eip`，只需要补充找到行号的部分，参考已经写好的部分，需要调用 `stab_binsearch`，查看头文件得知类型用 `N_SLINE`，搜索完毕后，判断是否 `lline < rline`（否则为没有找到），最后设置行号为 `lline` 即可。最后，按照 `info` 的信息打印 `backtrace`，需要注意最后要求打印 `eip` 和函数开始地址的相对距离（偏移量）。
5. 实现 `time` 功能
添加新的命令 `time`，需要新建 `mon_time` 函数。根据指示，使用 `x86.h` 提供的 `rdtsc` 的封装函数 `read_tsc` 得到 CPU 的运行周期数。对 `mon_time` 的参数进行解析，得到需要测量时间的函数名，模仿 `runcmd` 函数的解析命令并执行的方式，执行该函数，并记录调用前后的周期差，最后输出执行的周期数即可。