```
#include <string>
#include <istream>
#include <stdio.h>
#include <time.h>
#include <unistd.h>
#include <sys/sysinfo.h>

std::string exec(char *cmd) {
        FILE* pipe = popen(cmd, "r");
        if (!pipe) return "ERROR";
        char buffer[128];
        std::string result = "";
        while(!feof(pipe)) {
                if(fgets(buffer, 128, pipe) != NULL)
                        result += buffer;
        }
        pclose(pipe);
        return result;
}

std::string mem_usage(){
        char input[] = "free -m | awk 'NR==2 {print $3/$2*100}'";
        return exec(input);
}

std::string cpu_usage(){
        char input[] =
        "NUMCPUS=`grep ^proc /proc/cpuinfo | wc -l`; FIRST=`cat /proc/stat | awk '/^cpu
 / {print $5}'`; sleep 1; SECOND=`cat /proc/stat | awk '/^cpu / {print $5}'`; USED=`ech
o 2 k 100 $SECOND $FIRST - $NUMCPUS / - p | dc`; echo ${USED}";
        std::string temp = exec(input);
        temp[temp.length()-2] = 0;
        return temp;
}

std::string get_procs(){
        char input[] = "ps -e | wc -l";
        std::string temp = exec(input);
        temp[temp.length()-1] = 0;
        return temp;
}

std::string get_time(){
        time_t timer = time(0);
        struct tm stm;

        char buff[64];

        stm = *localtime(&timer);

        strftime(buff, sizeof(buff), "%Y-%m-%d.%X", &stm);

        return (std::string)buff;

}

std::string get_user(){
        char buff[64];
        getlogin_r(buff, sizeof(buff));
        return buff;
}

std::string get_hostname(){
```

```
        char buff[64];
        gethostname(buff, sizeof(buff));
        return (std::string) buff;
}
```