Basic Knowledge Test Project

**TEDDY Kim**

# **Purpose**

Visual C++, 리눅스 컴파일 / 빌드

- CPU, Memory, HDD 크기 등

- 동작중인 프로세스 리스트 보기

- 특정 폴더 파일 리스트 보기

- 네트워크 주소 확인

# **Table of Contents**

## Window Visual C++ version

* sisInfo.cpp
* processinfo.cpp
* filelistinfo.cpp
* networkinfo.cpp

## Linux (Unix style) version

* sisInfo.c
* processinfo.c
* filelistinfo.c
* networkinfo.c

## Python execute

* docker
* scp
* ssh

Window Visual C++ version

# Sisinfo.cpp

#include <windows.h> // windows info api

#include <stdio.h>

void hardware\_info();

// Getting the Hardware info(CPU, Memory, HDD)

void hardware\_info()

{

SYSTEM\_INFO siSysInfo;

GetSystemInfo(&siSysInfo);

printf("\tHardware information \n");

printf("====================================\n");

printf("OEM ID: %u\n", siSysInfo.dwOemId);

printf("Number of processors: %u\n", siSysInfo.dwNumberOfProcessors);

printf("Page size: %u\n", siSysInfo.dwPageSize);

printf("Processor type: %u\n", siSysInfo.dwProcessorType);

printf("Minimum application address: %lx\n", siSysInfo.lpMinimumApplicationAddress);

printf("Maximum application address: %lx\n", siSysInfo.lpMaximumApplicationAddress);

MEMORYSTATUSEX memInfo;

GlobalMemoryStatusEx(&memInfo);

memInfo.dwLength = sizeof(MEMORYSTATUSEX);

DWORDLONG totalPhysMem = memInfo.ullTotalPhys; // 전체 메모리

DWORDLONG availPhysMem = memInfo.ullAvailPhys; // 남은 메모리

printf("Total Mem: %I64u KB, Available Mem: %I64u KB\n", totalPhysMem / 1024, availPhysMem / 1024); // KB 로 출력

BOOL fResult;

unsigned \_\_int64 i64FreeBytesToCaller, i64TotalBytes, i64FreeBytes;

fResult = GetDiskFreeSpaceEx("C:",

(PULARGE\_INTEGER)&i64FreeBytesToCaller,

(PULARGE\_INTEGER)&i64TotalBytes,

(PULARGE\_INTEGER)&i64FreeBytes);

if (fResult)

{

printf("\nC: Total space = %I64u MB\n",

i64TotalBytes / (1024 \* 1024));

printf("Free space on drive = %I64u MB\n",

i64FreeBytes / (1024 \* 1024));

}

}

int main()

{

hardware\_info();

printf("====================================\n");

}

// \*Source

// <https://docs.microsoft.com/ko-kr/windows/win32/sysinfo/getting-hardware-information>

// <https://stackoverflow.com/questions/11917946/how-do-i-get-available-disk-space-from-windows-using-c>

// <https://ospace.tistory.com/514>

# Output

Hardware information

====================================

OEM ID: 9

Number of processors: 16

Page size: 4096

Processor type: 8664

Minimum application address: 10000

Maximum application address: fffeffff

Total Mem: 262208 KB, Available Mem: 1921 KB

C: Total space = 453215 MB

Free space on drive = 268304 MB

====================================

# ProcessInfo.cpp

Compile -> gcc -o processInfo.exe processInfo.cpp -lPSAPI

#include <windows.h>

#include <stdio.h>

#include <tchar.h>

#include <psapi.h> // process memory structure api by MSDN

// To ensure correct resolution of symbols, add Psapi.lib to TARGETLIBS

// and compile with -DPSAPI\_VERSION=1

void PrintProcessNameAndID( DWORD processID )

{

TCHAR szProcessName[MAX\_PATH] = TEXT("<unknown>");

// Get a handle to the process.

HANDLE hProcess = OpenProcess( PROCESS\_QUERY\_INFORMATION |

PROCESS\_VM\_READ,

FALSE, processID );

// Get the process name.

if (NULL != hProcess )

{

HMODULE hMod;

DWORD cbNeeded;

if ( EnumProcessModules( hProcess, &hMod, sizeof(hMod),

&cbNeeded) )

{

GetModuleBaseName( hProcess, hMod, szProcessName,

sizeof(szProcessName)/sizeof(TCHAR) );

}

}

// Print the process name and identifier.

\_tprintf( TEXT("%s (PID: %u)\n"), szProcessName, processID );

// Release the handle to the process.

CloseHandle( hProcess );

}

int main( void )

{

// Get the list of process identifiers.

DWORD aProcesses[1024], cbNeeded, cProcesses;

unsigned int i;

if ( !EnumProcesses( aProcesses, sizeof(aProcesses), &cbNeeded ) )

{

return 1;

}

// Calculate how many process identifiers were returned.

cProcesses = cbNeeded / sizeof(DWORD);

// Print the name and process identifier for each process.

printf("\tCurrent Running Processes \n");

printf("==========================================\n");

for ( i = 0; i < cProcesses; i++ )

{

if( aProcesses[i] != 0 )

{

PrintProcessNameAndID( aProcesses[i] );

}

}

printf("==========================================\n");

return 0;

}

//Source

//https://stackoverflow.com/questions/11564148/how-to-get-the-starting-base-address-of-a-process-in-c

# Output

Current Running Processes

==========================================

<unknown> (PID: 4)

<unknown> (PID: 172)

<unknown> (PID: 600)

<unknown> (PID: 900)

<unknown> (PID: 752)

<unknown> (PID: 1012)

<unknown> (PID: 1000)

<unknown> (PID: 1144)

<unknown> (PID: 1172)

<unknown> (PID: 1300)

<unknown> (PID: 1388)

<unknown> (PID: 1504)

<unknown> (PID: 1516)

<unknown> (PID: 1580)

<unknown> (PID: 1708)

<unknown> (PID: 1788)

<unknown> (PID: 1808)

<unknown> (PID: 1844)

<unknown> (PID: 1876)

<unknown> (PID: 1968)

<unknown> (PID: 2028)

<unknown> (PID: 2036)

<unknown> (PID: 2044)

<unknown> (PID: 760)

<unknown> (PID: 2052)

<unknown> (PID: 2112)

<unknown> (PID: 2164)

<unknown> (PID: 2176)

<unknown> (PID: 2184)

<unknown> (PID: 2204)

<unknown> (PID: 2548)

<unknown> (PID: 2580)

<unknown> (PID: 2636)

<unknown> (PID: 2740)

<unknown> (PID: 2816)

<unknown> (PID: 2820)

<unknown> (PID: 2868)

<unknown> (PID: 2888)

<unknown> (PID: 3000)

<unknown> (PID: 3036)

<unknown> (PID: 2496)

<unknown> (PID: 3232)

<unknown> (PID: 3240)

<unknown> (PID: 3248)

<unknown> (PID: 3256)

<unknown> (PID: 3264)

<unknown> (PID: 3276)

<unknown> (PID: 3284)

<unknown> (PID: 3292)

<unknown> (PID: 3300)

<unknown> (PID: 3456)

<unknown> (PID: 3464)

<unknown> (PID: 3492)

<unknown> (PID: 3512)

<unknown> (PID: 3520)

<unknown> (PID: 3528)

<unknown> (PID: 3552)

<unknown> (PID: 3660)

<unknown> (PID: 3784)

<unknown> (PID: 3792)

<unknown> (PID: 3852)

<unknown> (PID: 3872)

<unknown> (PID: 4072)

<unknown> (PID: 4904)

<unknown> (PID: 3136)

<unknown> (PID: 6136)

<unknown> (PID: 5404)

<unknown> (PID: 5612)

<unknown> (PID: 2896)

<unknown> (PID: 7124)

<unknown> (PID: 6348)

<unknown> (PID: 6340)

<unknown> (PID: 9096)

<unknown> (PID: 8060)

<unknown> (PID: 10668)

<unknown> (PID: 3676)

<unknown> (PID: 11956)

<unknown> (PID: 12576)

<unknown> (PID: 13240)

<unknown> (PID: 12932)

<unknown> (PID: 12472)

<unknown> (PID: 3048)

<unknown> (PID: 12624)

<unknown> (PID: 6952)

<unknown> (PID: 9836)

<unknown> (PID: 19208)

<unknown> (PID: 18404)

<unknown> (PID: 15712)

<unknown> (PID: 20948)

<unknown> (PID: 12480)

<unknown> (PID: 13484)

<unknown> (PID: 14524)

<unknown> (PID: 17476)

<unknown> (PID: 3616)

<unknown> (PID: 12068)

<unknown> (PID: 17356)

<unknown> (PID: 20020)

<unknown> (PID: 3172)

<unknown> (PID: 7144)

<unknown> (PID: 19204)

<unknown> (PID: 7052)

<unknown> (PID: 5320)

DPM.exe (PID: 2120)

sihost.exe (PID: 10792)

svchost.exe (PID: 20288)

svchost.exe (PID: 7500)

taskhostw.exe (PID: 20864)

Explorer.EXE (PID: 17620)

svchost.exe (PID: 2276)

<unknown> (PID: 20128)

AnySign4PC.exe (PID: 9500)

ApplicationFrameHost.exe (PID: 8164)

Microsoft.Notes.exe (PID: 13852)

RuntimeBroker.exe (PID: 18552)

StartMenuExperienceHost.exe (PID: 13656)

RuntimeBroker.exe (PID: 12808)

SearchApp.exe (PID: 20148)

RuntimeBroker.exe (PID: 5692)

YourPhone.exe (PID: 16744)

LockApp.exe (PID: 5496)

RuntimeBroker.exe (PID: 16392)

RuntimeBroker.exe (PID: 11816)

RuntimeBroker.exe (PID: 11976)

TextInputHost.exe (PID: 17836)

SecurityHealthSystray.exe (PID: 9552)

RtkAudUService64.exe (PID: 14932)

OneDrive.exe (PID: 9072)

KakaoTalk.exe (PID: 17188)

Cortana.exe (PID: 15132)

RuntimeBroker.exe (PID: 7924)

svchost.exe (PID: 20976)

Win32Bridge.Server.exe (PID: 14100)

svchost.exe (PID: 10620)

SettingSyncHost.exe (PID: 3404)

RuntimeBroker.exe (PID: 6668)

Microsoft.Photos.exe (PID: 14928)

RuntimeBroker.exe (PID: 10548)

WinStore.App.exe (PID: 9396)

DllHost.exe (PID: 7152)

chrome.exe (PID: 7528)

chrome.exe (PID: 16896)

chrome.exe (PID: 15316)

chrome.exe (PID: 19088)

chrome.exe (PID: 1428)

chrome.exe (PID: 19372)

chrome.exe (PID: 2100)

chrome.exe (PID: 6284)

chrome.exe (PID: 1720)

chrome.exe (PID: 18740)

chrome.exe (PID: 21144)

chrome.exe (PID: 12592)

chrome.exe (PID: 9904)

chrome.exe (PID: 6480)

chrome.exe (PID: 11876)

Code.exe (PID: 14120)

Code.exe (PID: 5212)

Code.exe (PID: 16968)

Code.exe (PID: 14904)

Code.exe (PID: 17908)

Code.exe (PID: 18580)

Code.exe (PID: 9324)

Code.exe (PID: 4416)

Code.exe (PID: 18244)

conhost.exe (PID: 7088)

powershell.exe (PID: 14184)

cpptools.exe (PID: 17684)

conhost.exe (PID: 16716)

git.exe (PID: 10236)

Code.exe (PID: 1432)

conhost.exe (PID: 20372)

vsls-agent.exe (PID: 18148)

Code.exe (PID: 5636)

<unknown> (PID: 12488)

<unknown> (PID: 13668)

ShellExperienceHost.exe (PID: 20820)

RuntimeBroker.exe (PID: 18844)

<unknown> (PID: 18392)

<unknown> (PID: 19252)

<unknown> (PID: 11372)

<unknown> (PID: 10112)

VirtualBox.exe (PID: 11648)

VBoxSVC.exe (PID: 4248)

<unknown> (PID: 4940)

VirtualBoxVM.exe (PID: 19676)

VirtualBoxVM.exe (PID: 7036)

VirtualBoxVM.exe (PID: 6248)

SystemSettings.exe (PID: 3960)

UserOOBEBroker.exe (PID: 13804)

chrome.exe (PID: 20816)

<unknown> (PID: 11964)

chrome.exe (PID: 19872)

chrome.exe (PID: 1060)

chrome.exe (PID: 20928)

chrome.exe (PID: 13384)

Video.UI.exe (PID: 16872)

RuntimeBroker.exe (PID: 5236)

chrome.exe (PID: 14996)

chrome.exe (PID: 11448)

chrome.exe (PID: 18020)

<unknown> (PID: 1672)

WINWORD.EXE (PID: 18292)

chrome.exe (PID: 11912)

splwow64.exe (PID: 17900)

svchost.exe (PID: 15836)

chrome.exe (PID: 9012)

SearchProtocolHost.exe (PID: 12884)

<unknown> (PID: 5960)

cpptools-srv.exe (PID: 18664)

processInfo.exe (PID: 9412)

==========================================

# filelistinfo.cpp

#include <iostream>

#include <io.h>

#include <stdio.h>

#include <string>

using namespace std;

void filelist (string path);

int main()

{

string path = "C:\\Users\\teddy\\TechBase\\\*.\*";

filelist(path);

}

void filelist(string path)

{

struct \_finddata\_t fd;

intptr\_t handle;

if ((handle = \_findfirst(path.c\_str(), &fd)) == -1L)

std::cout << "No file in directory!" << std::endl;

std::cout << "in the " << path << std::endl;

printf("\n\tFile list information \n");

printf("====================================\n");

do

{

cout << "File: " << fd.name << endl;

} while (\_findnext(handle, &fd) == 0);

\_findclose(handle);

printf("====================================\n");

}

# Output

in the C:\Users\teddy\TechBase\\*.\*

File list information

====================================

File: .

File: ..

File: .git

File: .gitattributes

File: .gitignore

File: .ipynb\_checkpoints

File: .vs

File: .vscode

File: Basic knowledge Test Project.docx

File: execute.py

File: filelist.cpp

File: filelist.exe

File: ip.txt

File: linuxs.c

File: network.cpp

File: network.exe

File: network.txt

File: process.cpp

File: process2.cpp

File: process2.exe

File: process3.c

File: Python\_exe\_ssh.py

File: p\_list.txt

File: README.md

File: sisinfo.cpp

File: sisinfo.exe

File: Untitled.ipynb

File: Untitled1.ipynb

File: visualC.cpp

File: Wincpp.cpp

File: ~$sic knowledge Test Project.docx

====================================

# Network.cpp

Compile -> gcc -o network.exe network.cpp -lws2\_32

//#define \_CRT\_SECURE\_NO\_WARNINGS

#include <WinSock2.h> // 소켓 lib

#include <Windows.h>

#include <stdio.h>

// #pragma warning(disable:4996)

// #pragma comment(lib, "ws2\_32.lib")

FILE\* fp;

void ViewLocalHostIPv4();

int main()

{

WSADATA wsadata;

WSAStartup(MAKEWORD(2, 2), &wsadata);

fp = \_wfopen(L"network.txt", L"w+");

ViewLocalHostIPv4();

WSACleanup();

fclose(fp);

return 0;

}

void ViewLocalHostIPv4()

{

char localhostname[MAX\_PATH];

IN\_ADDR addr = { 0, };

if (gethostname(localhostname, MAX\_PATH) == SOCKET\_ERROR)//호스트 이름 얻어오기

return;

HOSTENT\* ptr = gethostbyname(localhostname);//호스트 엔트리 얻어오기

printf("Network list information \n");

printf("====================================\n");

while (ptr && ptr->h\_name)

{

if (ptr->h\_addrtype == PF\_INET)//IPv4 주소 타입일 때

{

for (int index = 0; ptr->h\_addr\_list[index]; index++)

{

printf("IP: ");

memcpy(&addr, ptr->h\_addr\_list[index], ptr->h\_length);//메모리 복사

printf("%s\n", inet\_ntoa(addr));

fprintf(fp, "%s\n", inet\_ntoa(addr));

//inet\_aton IP4바이너리, inet\_ntoa IP4주소, inet\_addr, inet\_network

}

}

ptr++;

}

printf("====================================\n");

}

//Source

//https://ehpub.co.kr/2-4-%EB%A1%9C%EC%BB%AC-%ED%98%B8%EC%8A%A4%ED%8A%B8-ip-%EC%A3%BC%EC%86%8C-%EC%96%BB%EC%96%B4%EC%98%A4%EA%B8%B0-tcpip-%EC%86%8C%EC%BC%93-%ED%86%B5%EC%8B%A0-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98/

# Output

Network list information

====================================

IP: 192.168.56.1

IP: 172.31.96.1

IP: 10.50.31.220

====================================

1 ip – VM

2 ip - HyperV

3 ip – localhost

Linux Unix version

#include <stdio.h>

#include <stdlib.h>

#include <sys/sysinfo.h>

#include <sys/utsname.h>

#include <sys/types.h>

#include <dirent.h>

#include <sys/ioctl.h>

#include <net/if.h>

#include <arpa/inet.h>

#include <string.h>

#include <iostream>

#include <cstring>

#include <cstdlib>

#include <proc/readproc.h>

void hardware\_info()

{

struct utsname un;

struct sysinfo sys;

if(sysinfo(&sys) < 0) {

fprintf(stderr, "sysinfo() function error.\n");

exit(1);

}

fprintf(stdout, "\tHardware information \n");

fprintf(stdout, "===================================== \n");

fprintf(stdout,

"업타임: %ld days\n"

"메모리 총계: %ld KB\n"

"사용가능 메모리: %ldKB\n"

"현재 프로세스 갯수: %d\n"

"CPU 갯수: %d\n\n",

((sys.uptime/60)/60)/24, sys.totalram/1024, sys.freeram/1024,

sys.procs, get\_nprocs\_conf());

if(uname(&un) < 0) {

fprintf(stderr, "uname() function error.\n");

exit(1);

}

fprintf(stdout,

"시스템 이름: %s\n"

"노드명: %s\n"

"커널버전: %s\n"

"릴리즈버전: %s\n"

"프로세서타입: %s\n",

un.sysname, un.nodename, un.release, un.version, un.machine);

}

void process\_info()

{

using std::cout;

PROCTAB\* proc = openproc(PROC\_FILLMEM | PROC\_FILLSTAT | PROC\_FILLSTATUS);

proc\_t proc\_info;

memset(&proc\_info, 0, sizeof(proc\_info));

cout << "\tRunning Process information" << std::endl;

cout << "=====================================" << std::endl;

cout << "Program\tPID\tPPID\tMEM\tutime\tstime\n";

while (readproc(proc, &proc\_info) != NULL)

{

cout << proc\_info.cmd << "\t" << proc\_info.tid;

cout << proc\_info.ppid << "\t" << proc\_info.resident;

cout << proc\_info.utime << "\t" << proc\_info.stime << "\n";

}

closeproc(proc);

}

void file\_info()

{

fprintf(stdout, "\tfile list information \n");

fprintf(stdout, "===================================== \n");

DIR \*dir;

struct dirent \*ent;

dir = opendir("./");

if(dir)

{

while (ent = readdir (dir))

printf("%s\n", ent->d\_name);

}

closedir(dir);

}

void network\_info()

{

fprintf(stdout, "\tnetwork information \n");

fprintf(stdout, "===================================== \n");

struct ifreq ifr;

char ipstr[40];

int s;

s = socket(AF\_INET, SOCK\_DGRAM, 0);

strncpy(ifr.ifr\_name, "enp0s3", IFNAMSIZ);

if(ioctl(s, SIOCGIFADDR, &ifr) < 0)

printf("no network");

inet\_ntop(AF\_INET, ifr.ifr\_addr.sa\_data+2, ipstr, sizeof(struct sockaddr));

printf("IP Address : %s\n", ipstr);

}

int main(void)

{

hardware\_info();

fprintf(stdout, "\n===================================== \n");

process\_info();

fprintf(stdout, "\n===================================== \n");

file\_info();

fprintf(stdout, "\n===================================== \n");

network\_info();

}

# Output

=====================================

Hardware information

=====================================

업타임: 0 days

메모리 총계: 4023440 KB

사용가능 메모리: 911756KB

현재 프로세스 갯수: 491

CPU 갯수: 1

시스템 이름: Linux

노드명: teddy-VirtualBox

커널버전: 5.11.0-40-generic

릴리즈버전: #44~20.04.2-Ubuntu SMP Tue Oct 26 18:07:44 UTC 2021

프로세서타입: x86\_64

=====================================

Process information

=====================================

Program PID PPID MEM utime stime

systemd 10 323190 96

kthreadd 20 00 0

rcu\_gp 32 00 0

rcu\_par\_gp 42 00 0

kworker/0:0H-events\_highpri 62 00 0

mm\_percpu\_wq 92 00 0

rcu\_tasks\_rude\_ 102 00 0

rcu\_tasks\_trace 112 00 0

ksoftirqd/0 122 00 64

rcu\_sched 132 081 22

migration/0 142 029 0

idle\_inject/0 152 00 0

cpuhp/0 162 00 0

kdevtmpfs 172 00 0

netns 182 00 0

inet\_frag\_wq 192 00 0

kauditd 202 00 0

khungtaskd 212 02 0

oom\_reaper 222 00 0

writeback 232 00 0

kcompactd0 242 061 103

ksmd 252 00 0

khugepaged 262 07 0

kintegrityd 722 00 0

kblockd 732 00 0

blkcg\_punt\_bio 742 00 0

tpm\_dev\_wq 752 00 0

ata\_sff 762 00 0

md 772 00 0

edac-poller 782 00 0

devfreq\_wq 792 00 0

watchdogd 802 00 0

kworker/0:1H-kblockd 822 00 157

kswapd0 842 00 16

ecryptfs-kthrea 852 00 0

kthrotld 872 00 0

acpi\_thermal\_pm 882 00 0

scsi\_eh\_0 892 00 1

scsi\_tmf\_0 902 00 0

scsi\_eh\_1 912 00 0

scsi\_tmf\_1 922 00 0

vfio-irqfd-clea 942 00 0

ipv6\_addrconf 962 00 0

kstrp 1052 00 0

zswap-shrink 1082 00 0

kworker/u3:0 1092 00 0

charger\_manager 1142 00 0

scsi\_eh\_2 1592 01 0

scsi\_tmf\_2 1602 00 0

jbd2/sda5-8 1812 00 53

ext4-rsv-conver 1822 00 0

systemd-journal 2211 9026179 190

bpfilter\_umh 2422 1280 0

systemd-udevd 2591 18386 6

loop0 3142 00 0

loop1 3212 00 0

loop2 3222 00 0

loop3 3232 00 1

loop4 3242 00 0

loop5 3252 00 0

loop6 3262 00 0

loop7 3272 00 0

loop8 3282 00 0

cryptd 3662 00 0

irq/18-vmwgfx 3682 00 70

ttm\_swap 3932 00 0

card0-crtc0 3942 00 0

card0-crtc1 3972 00 0

card0-crtc2 3982 00 0

card0-crtc3 4012 00 0

card0-crtc4 4042 00 0

card0-crtc5 4082 00 0

card0-crtc6 4092 00 0

card0-crtc7 4112 00 0

systemd-resolve 5651 336350 45

systemd-timesyn 5661 14934 3

accounts-daemon 6031 241634 17

acpid 6041 17917 60

avahi-daemon 6071 80616 16

cron 6081 7244 0

dbus-daemon 6101 1438110 23

NetworkManager 6111 5548531 335

networkd-dispat 6191 49714 0

polkitd 6201 335532 20

rsyslogd 6221 111938 19

snapd 6231 7878347 221

switcheroo-cont 6241 15320 0

systemd-logind 6251 19925 10

udisksd 6261 35836 1

wpa\_supplicant 6271 124217 0

avahi-daemon 635607 780 0

ModemManager 6871 27273 1

containerd 7021 129241732 294

unattended-upgr 7121 56782 1

gdm3 7181 21931 1

gdm-session-wor 733718 24662 2

systemd 7541 260842 17

(sd-pam) 755754 6710 0

pulseaudio 763754 52085868 4722

tracker-miner-f 765754 60567 0

gnome-keyring-d 7691 20084 0

gdm-x-session 773733 16310 0

Xorg 775773 365315793 1715

rtkit-daemon 7771 73514 14

dbus-daemon 779754 163732 10

gvfsd 790754 19882 0

gvfsd-fuse 796754 20730 0

gvfs-udisks2-vo 806754 290136 17

gvfs-afc-volume 812754 224255 55

gvfs-goa-volume 820754 15050 0

goa-daemon 830754 912812 4

goa-identity-se 876754 28353 6

gvfs-mtp-volume 882754 15860 0

gvfs-gphoto2-vo 886754 16900 0

upowerd 8921 24522 1

gnome-session-b 908773 38941 1

ssh-agent 982908 11415 0

at-spi-bus-laun 1005754 23300 0

dbus-daemon 10111005 10306 1

gnome-session-c 1024754 10840 0

gnome-session-b 1031754 44534 3

gnome-shell 1044754 11258617965 1049

whoopsie 10751 395721 4

kerneloops 10821 1090 74

kerneloops 10851 1130 73

gvfsd-metadata 1206754 16341 2

ibus-daemon 12201044 28701550 470

ibus-dconf 12241220 23070 0

ibus-extension- 12261220 8529206 25

ibus-x11 1231754 759518 3

ibus-portal 1266754 23493 0

at-spi2-registr 1398754 158616 8

xdg-permission- 1579754 11680 0

gnome-shell-cal 1598754 51953 1

evolution-sourc 1618754 65663 0

evolution-calen 1628754 765713 3

dconf-service 1643754 14600 1

evolution-addre 1646754 743410 3

nautilus 1667754 31905541 39

gjs 1669754 65812 0

gvfsd-trash 1675790 27545 1

gsd-a11y-settin 1689754 23540 0

gsd-color 1690754 788530 6

gsd-datetime 1691754 42871 0

gsd-housekeepin 1692754 245167 18

gsd-keyboard 1693754 773920 3

gsd-media-keys 1694754 831627 7

gsd-power 1695754 787225 3

gsd-print-notif 1696754 27410 0

gsd-rfkill 1701754 15645 2

gsd-screensaver 1703754 15150 0

gsd-sharing 1704754 309219 1

gsd-smartcard 1708754 27270 0

gsd-sound 1711754 28670 0

gsd-usb-protect 1714754 24631 0

gsd-wacom 1718754 760620 3

gsd-wwan 1734754 27440 0

gsd-xsettings 1741754 800524 4

gsd-printer 1761754 36791 0

evolution-alarm 17871031 1540527 4

gsd-disk-utilit 17931031 14170 0

colord 18291 41475 0

ibus-engine-han 18611220 2663607 122

update-notifier 19481031 1377273 20

ibus-engine-sim 19811220 23216 0

gvfsd-network 5207790 26820 0

gvfsd-dnssd 5219790 21161 0

GeckoMain 31095754 1112592899 618

Socket Process 3114031095 97421 0

Privileged Cont 3115831095 3833668 13

WebExtensions 3120431095 25242392 81

Web Content 3124131095 57247980 150

Web Content 3126931095 181757 6

sd\_dummy 31313754 143474 60

sd\_generic 31316754 147755 79

sd\_espeak-ng 31319754 204680 90

speech-dispatch 31325754 5470 0

sudo 32621754 10990 0

ssh 3262232621 14414 8

sshd 326231 20011 0

sshd 3270432623 13123 5

bash 3270532704 11301 0

sudo 3274832705 11220 0

ssh 3274932748 14503 3

sshd 327501 20371 0

sshd 3280632750 13222 5

bash 3280732806 11642 0

cupsd 332971 18540 0

cups-browsed 332981 28074 2

scp 3574232807 11460 0

ssh 3574335742 14170 0

scp 3574832807 10960 0

ssh 3574935748 14690 0

su 36093754 9640 0

bash 3609436093 9133 4

loop9 366292 00 0

dockerd 367731 20667261 138

containerd 3681836773 10780651 493

dockerd 369331 21767147 2

vi 3760036094 10190 0

sshd 377361 17980 0

containerd-shim 377681 2119130 99

bash 3779037768 8771 4

sudo 3782236094 11850 0

docker 3782337822 1453397 28

ssh 4195032807 15551 2

sshd 4195137736 22481 0

sshd 4203041951 12781 3

bash 4203142030 12432 0

su 4280137790 7170 0

bash 4280242801 9071 4

ssh 4311042031 15650 0

sudo 4311342031 11551 0

ssh 4311443113 15720 0

sshd 4320037790 10680 0

ssh 4325842031 15510 0

sudo 43287754 11940 0

ssh 4328843287 15771 1

sshd 4328943200 20151 1

bash 4330043289 9721 0

ssh-agent 43344769 9380 0

kworker/0:2-events 436852 00 631

kworker/u2:2-ext4-rsv-conversion 441742 03 0

kworker/0:1-events 442022 00 36

gedit 44212754 41928270 23

kworker/u2:1-events\_unbound 442252 00 4

gnome-terminal- 44242754 1580789 17

bash 4433744242 13463 1

kworker/u2:0-events\_power\_efficient 444112 00 0

geany 44413754 1983037 4

bash 4442244413 12060 0

tracker-store 44465754 64363 0

process.exe 4447244337 10860 0

=====================================

file list information

=====================================

.gitattributes

.vscode

.git

network.cpp

execute.py

linuxs.c

network.txt

process.cpp

.gitignore

filelist.cpp

.

process2.cpp

sisinfo.cpp

Python\_exe\_ssh.py

linuxs.exe

README.md

ip.txt

visualC.cpp

p\_list.txt

..

Wincpp.cpp

=====================================

network information

=====================================

IP Address : 10.0.2.15

Python execute

# Docker & Scp & Ssh

/\*

// docker

docker start ubuntu2004

docker attach ubuntu2004

ssh [teddy@10.0.2.15](mailto:teddy@10.0.2.15)

// scp (Linux -> Windows)

scp /home/teddy/문서/TechBase/linuxs.c [teddy@172.31.96.1:C:/Users/teddy/TechBase](mailto:teddy@172.31.96.1:C:/Users/teddy/TechBase)

\*/

import os

os.system('C:\\Users\\teddy\\source\\repos\\Project4\\x64\\Debug\\Project4.exe')

import paramiko

try:

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy)

ssh.connect("127.0.0.1", port="22", username="teddy", password="gusdn147")

print("connected!!")

stdin, stdout, stderr = ssh.exec\_command("cd 문서 ; gcc -o linuxs.exe linuxs.c ; ./linuxs.exe")

lines = stdout.readlines()

resultData = ''.join(lines)

print(resultData)

ssh.close()

except Exception as err:

print(err)

# Output

connected!!

Hardware information

=====================================

업타임: 0 days

메모리 총계: 4023440 KB

사용가능 메모리: 128212KB

현재 프로세스 갯수: 655

CPU 갯수: 1

시스템 이름: Linux

노드명: teddy-VirtualBox

커널버전: 5.11.0-40-generic

릴리즈버전: #44~20.04.2-Ubuntu SMP Tue Oct 26 18:07:44 UTC 2021

프로세서타입: x86\_64

=====================================

Process information

=====================================

PID TTY TIME CMD

754 ? 00:00:00 systemd

755 ? 00:00:00 (sd-pam)

763 ? 00:01:46 pulseaudio

765 ? 00:00:00 tracker-miner-f

769 ? 00:00:00 gnome-keyring-d

779 ? 00:00:00 dbus-daemon

790 ? 00:00:00 gvfsd

796 ? 00:00:00 gvfsd-fuse

806 ? 00:00:00 gvfs-udisks2-vo

812 ? 00:00:01 gvfs-afc-volume

820 ? 00:00:00 gvfs-goa-volume

830 ? 00:00:00 goa-daemon

876 ? 00:00:00 goa-identity-se

882 ? 00:00:00 gvfs-mtp-volume

886 ? 00:00:00 gvfs-gphoto2-vo

982 ? 00:00:00 ssh-agent

1005 ? 00:00:00 at-spi-bus-laun

1011 ? 00:00:00 dbus-daemon

1024 ? 00:00:00 gnome-session-c

1031 ? 00:00:00 gnome-session-b

1044 ? 00:03:43 gnome-shell

1206 ? 00:00:00 gvfsd-metadata

1220 ? 00:00:21 ibus-daemon

1224 ? 00:00:00 ibus-dconf

1226 ? 00:00:02 ibus-extension-

1231 ? 00:00:00 ibus-x11

1266 ? 00:00:00 ibus-portal

1398 ? 00:00:00 at-spi2-registr

1579 ? 00:00:00 xdg-permission-

1598 ? 00:00:00 gnome-shell-cal

1618 ? 00:00:00 evolution-sourc

1628 ? 00:00:00 evolution-calen

1643 ? 00:00:00 dconf-service

1646 ? 00:00:00 evolution-addre

1667 ? 00:00:08 nautilus

1669 ? 00:00:00 gjs

1675 ? 00:00:00 gvfsd-trash

1689 ? 00:00:00 gsd-a11y-settin

1690 ? 00:00:00 gsd-color

1691 ? 00:00:00 gsd-datetime

1692 ? 00:00:00 gsd-housekeepin

1693 ? 00:00:00 gsd-keyboard

1694 ? 00:00:00 gsd-media-keys

1695 ? 00:00:00 gsd-power

1696 ? 00:00:00 gsd-print-notif

1701 ? 00:00:00 gsd-rfkill

1703 ? 00:00:00 gsd-screensaver

1704 ? 00:00:00 gsd-sharing

1708 ? 00:00:00 gsd-smartcard

1711 ? 00:00:00 gsd-sound

1714 ? 00:00:00 gsd-usb-protect

1718 ? 00:00:00 gsd-wacom

1734 ? 00:00:00 gsd-wwan

1741 ? 00:00:00 gsd-xsettings

1761 ? 00:00:00 gsd-printer

1787 ? 00:00:00 evolution-alarm

1793 ? 00:00:00 gsd-disk-utilit

1861 ? 00:00:07 ibus-engine-han

1948 ? 00:00:01 update-notifier

1981 ? 00:00:00 ibus-engine-sim

5207 ? 00:00:00 gvfsd-network

5219 ? 00:00:00 gvfsd-dnssd

32704 ? 00:00:00 sshd

32806 ? 00:00:00 sshd

42030 ? 00:00:00 sshd

43344 ? 00:00:00 ssh-agent

44242 ? 00:00:02 gnome-terminal-

45113 ? 00:00:09 GeckoMain

45159 ? 00:00:00 Socket Process

45177 ? 00:00:00 Privileged Cont

45227 ? 00:00:00 WebExtensions

45267 ? 00:00:11 Web Content

45299 ? 00:00:00 Web Content

45324 ? 00:00:00 RDD Process

45467 ? 00:00:00 sshd

45468 ? 00:00:00 bash

45471 ? 00:00:00 linuxs.exe

45472 ? 00:00:00 sh

45473 ? 00:00:00 ps

=====================================

file list information

=====================================

file.ex

process.exe

a.out

sysinfo.c

Linux.odt

GNOME

process3.exe

sysinfo\_ex

linuxs.c

dir

prac2.c

filelist.c

network.ex

TechBase

.aaa.swp

.

linuxs.exe

aaa

network.c

process3.c

prac.c

.file.txt.swp

linuxs

.swp

..

file.txt

process.c

무제

=====================================

network information

=====================================

IP Address : 10.0.2.15

**Github Link** ->

<https://github.com/teddykim530/TechBase>