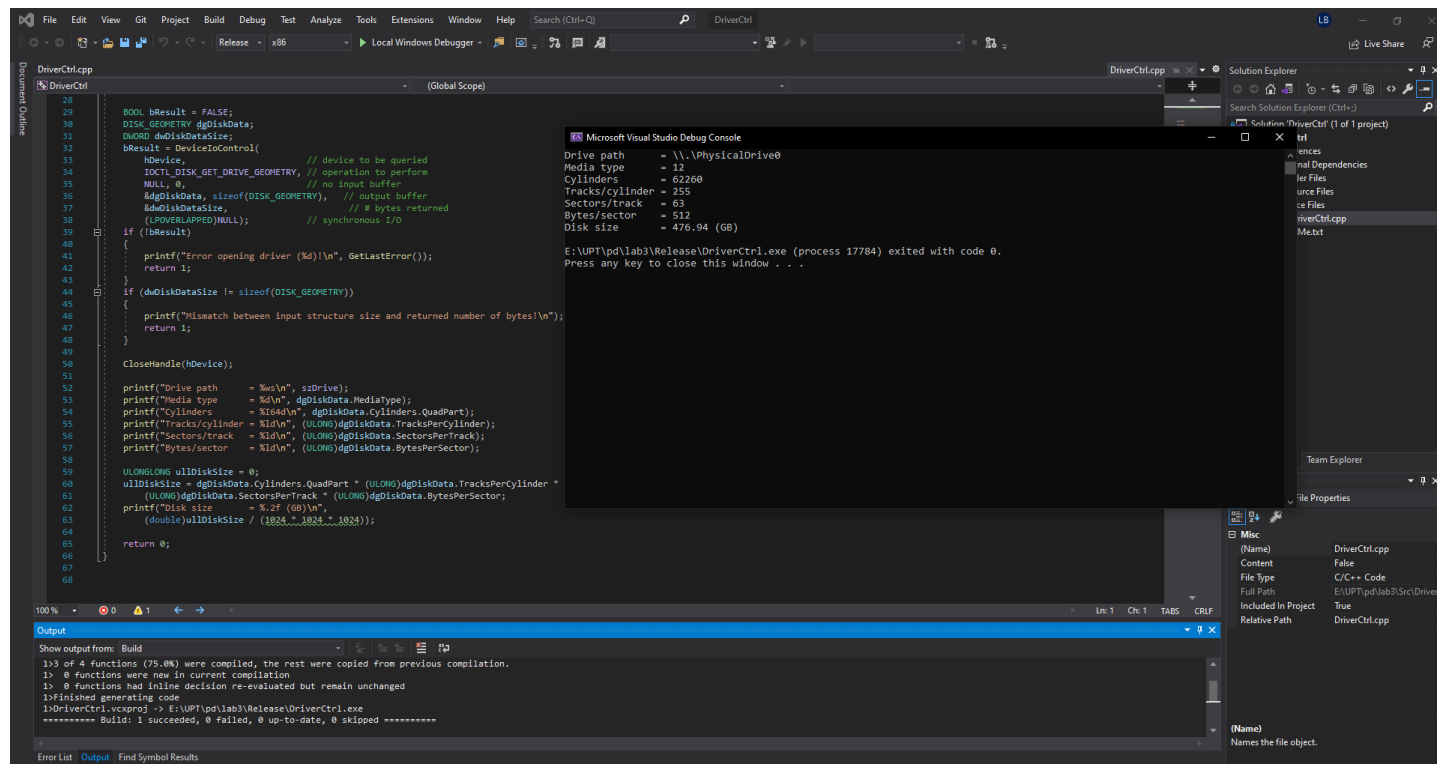


Tema - laborator 3

Detalii sistem

- *procesor: Intel I7-10750H*
- *memorie: 16GB*
- *stocare: SSD de capacitate 512 GB*

Rezultate rulare cod initial



```
DriverCtrl.cpp
29  bool bResult = FALSE;
30  DISK_GEOMETRY dgDiskData;
31  DWORD dwDiskDataSize;
32  bResult = DeviceIoControl(
33      hDevice,           // device to be queried
34      IOCTL_DISK_GET_DRIVE_GEOMETRY, // operation to perform
35      NULL, 0,           // no input buffer
36      &dgDiskData, sizeof(DISK_GEOMETRY), // output buffer
37      &dwDiskDataSize, // # bytes returned
38      (LPOVERLAPPED)NULL); // synchronous I/O
39
40  if (!bResult)
41  {
42      printf("Error opening driver (%d)\n", GetLastError());
43      return 1;
44  }
45  if (dwDiskDataSize != sizeof(DISK_GEOMETRY))
46  {
47      printf("Mismatch between input structure size and returned number of bytes\n");
48      return 1;
49  }
50
51  CloseHandle(hDevice);
52
53  printf("Drive path      = %s\n", szDrive);
54  printf("Media type       = %d\n", dgDiskData.MediaType);
55  printf("Cylinders         = %i64d\n", dgDiskData.Cylinders.QuadPart);
56  printf("Tracks/cylinder   = %i64d", (ULONG)dgDiskData.TracksPerCylinder);
57  printf("Sectors/track     = %i64d", (ULONG)dgDiskData.SectorsPerTrack);
58  printf("Bytes/sector      = %i64d", (ULONG)dgDiskData.BytesPerSector);
59
60  ULONG ulDiskSize = 0;
61  ulDiskSize = dgDiskData.Cylinders.QuadPart * (ULONG)dgDiskData.TracksPerCylinder *
62              (ULONG)dgDiskData.SectorsPerTrack * (ULONG)dgDiskData.BytesPerSector;
63  printf("Disk size        = %.2f (GB)\n",
64         (double)ulDiskSize / (1024 * 1024 * 1024));
65
66  return 0;
67
68

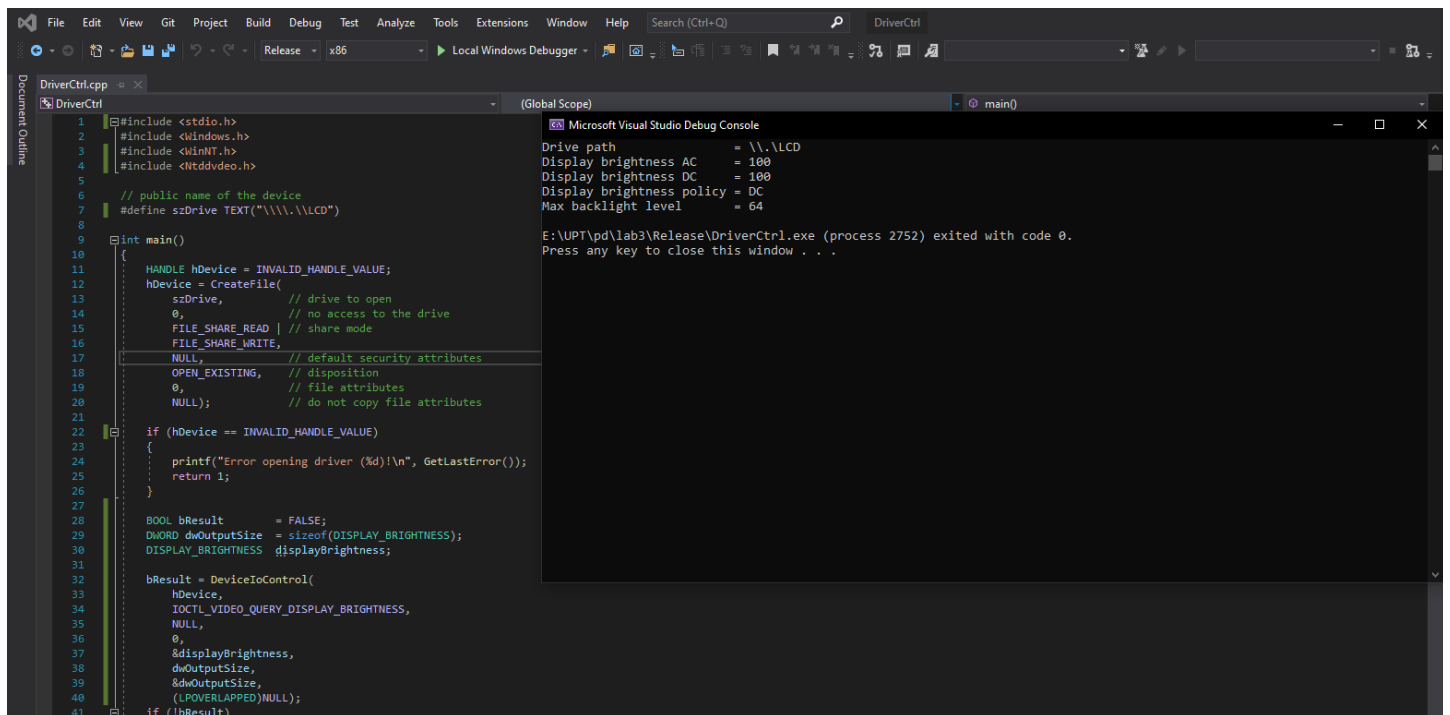
Microsoft Visual Studio Debug Console
Drive path      = \\.\PhysicalDrive0
Media type       = 12
Cylinders        = 62260
Tracks/cylinder  = 255
Sectors/track    = 63
Bytes/sector     = 512
Disk size        = 476.94 (GB)

E:\UPT\pd\lab3\Release\DriverCtrl.exe (process 17784) exited with code 0.
Press any key to close this window . . .

Output
Show output from: Build
13 of 4 functions (75.0%) were compiled, the rest were copied from previous compilation.
1> 0 functions were new in current compilation
1> 0 functions had inline decision re-evaluated but remain unchanged
1> finished generating code
1> DriverCtrl.vcxproj -> E:\UPT\pd\lab3\Release\DriverCtrl.exe
***** Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped *****

Error List  Output  Find Symbol Results
```

Rezultate aplicatie modificata



IOCTL_VIDEO_QUERY_DISPLAY_BRIGHTNESS

Pentru a putea obtine informatii privind stralucirea display-ului si modul de rulare al acestuia, functia DeviceIoControl trebuie apelata in urmatoarul mod:

```
BOOL result = DeviceIoControl(  
    (HANDLE) hDevice,  
    IOCTL_VIDEO_QUERY_DISPLAY_BRIGHTNESS,  
    NULL,  
    0,  
    (LPVOID) lpOutBuffer,  
    (DWORD) nOutBufferSize,  
    (LPDWORD) lpBytesReturned,  
    (LPOVERLAPPED) lpOverlapped
```

unde

- `hDevice` este handle-ul returnat de functia `CreateFile` pentru drive-ul `\\.\\LCD`
- `IOCTL_VIDEO_QUERY_DISPLAY_BRIGHTNESS` este function code-ul
- cel de-al treilea parametru(`NULL`) este bufferul de input
- cel de-al patrulea parametru(`0`) este size-ul in bytes al bufferului de input
- `lpOutBuffer` un pointer catre o structura de tip `DISPLAY_BRIGHTNESS`
- `nOutBufferSize` numarul de bytes al bufferului de output(`DISPLAY_BRIGHTNESS`)
- `lpBytesReturned` numarul de bytes returnat de API

DISPLAY_BRIGHTNESS

```
typedef struct _DISPLAY_BRIGHTNESS {  
    UCHAR ucDisplayPolicy;  
    UCHAR ucACBrightness;  
    UCHAR ucDCBrightness;  
} DISPLAY_BRIGHTNESS, *PDISPLAY_BRIGHTNESS;
```

`ucACBrightness` = valoarea factorului de stralucire al ecranului in curent alternativ

`ucDCBrightness` = valoarea factorului de stralucire al ecranului in curent continuu

`ucDisplayPolicy` = modul de rulare

IOCTL_VIDEO_QUERY_SUPPORTED_BRIGHTNESS

Pentru a obtine informatii despre nivelurile backlightului, functia DeviceIoControl trebuie apelata in felul urmator:

```
BOOL result = DeviceIoControl(
    (HANDLE) hDevice,
    IOCTL_VIDEO_QUERY_SUPPORTED_BRIGHTNESS,
    NULL,
    0,
    (LPVOID) lpOutBuffer,
    (DWORD) nOutBufferSize,
    (LPDWORD) lpBytesReturned,
    (LPOVERLAPPED) lpOverlapped
);
```

unde

- `hDevice` este handle-ul returnat de functia `CreateFile` pentru drive-ul **\\LCD**
- `IOCTL_VIDEO_QUERY_SUPPORTED_BRIGHTNESS` este function code-ul
- cel de-al treilea parametru(**NULL**) este bufferul de input
- cel de-al patrulea parametru(0) este size-ul in bytes al bufferului de input
- `lpOutBuffer` un pointer catre un buffer de 256 de bytes care va contine nivelurile de putere
- `nOutBufferSize` numarul de bytes al bufferului de output(256)
- `lpBytesReturned` numarul de bytes returnat de API -> `lpOutBuffer[lpBytesReturned - 1] ==` nivelul maxim