

Re: Getting same physical address for 2 different user addresses.

Re: Getting same physical address for 2 different user addresses.

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2008-06/msg00454.html>

- *From:* saravana <saravana@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 20 Jun 2008 04:02:00 -0700
-

Thanks Mark, Daniel and Tim for your response.

We are allocating the buffers readbuf and writebuf in the application using malloc.

```
readbuf = malloc(nof pages * 4K);
```

Then we are sending this buffer as inputbuffer to Driver using IOCTL defined as METHOD_IN_DIRECT.

```
writebuf = malloc(nof pages * 4K);
```

Then we are sending this buffer as inputbuffer to Driver using IOCTL defined as METHOD_IN_DIRECT.

Here is the driver code handles the Physical Address conversion,

```
NTSTATUS LockUserModeMemory(  
PDEVICE_EXTENSION pDevExt,  
WDF_REQUEST_PARAMETERS UserModeMemLockParams,  
IN WDFREQUEST UserModeMemLockRequest,  
ULONG *pDataLockedMemSize){
```

```
NTSTATUS ntStatus = STATUS_SUCCESS;  
PVOID pUserModeMemLockOutputBuffer = NULL;  
PVOID pUserModeMemLockInputBuffer = NULL;  
size_t UserModeMemLockOutputBufSize;  
size_t UserModeMemLockInputBufSize;
```

```
PLOCK_USER_MODE_MEMORY_INPARAMS pLockUserModeMemInParams= NULL;  
PLOCK_USER_MODE_MEMORY_OUTPARAMS pLockUserModeMemOutParams= NULL;
```

```
PVOID pIoMapedBaseVa = NULL;  
PVOID pTempLockedMem = NULL;  
PVOID pTempLockedPage = NULL;  
PHYSICAL_ADDRESS UserModeMemLockPhyAddr;  
PHYSICAL_ADDRESS lowAddress;
```

Re: Getting same physical address for 2 different user addresses.

Re: Getting same physical address for 2 different user addresses.

```
PHYSICAL_ADDRESS highAddress;
ULONG ulIndex=0;
PMDL pDmaMdl;
ULONG ulSize=0;
PVOID inBuf, outBuf;
PVOID reqContext = NULL;
size_t inBufLen,outBufLen;

WDF_OBJECT_ATTRIBUTES attributes;

UCHAR *pTempUserModeBufStart;
PVOID pUserModeBuff = NULL;
ULONG ulNoOfPages=0;

KIRQL kOldIrq;
KIRQL kCurrentIrq;

DbgPrint("%s---->\n",__FUNCTION__);

if(UserModeMemLockParams.Parameters.DeviceIoControl.InputBufferLength<=0){
DbgPrint("Buff size is small\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INVALID_PARAMETER;
goto EXIT;
}

pDevExt->ulPageNeeded=0;
ntStatus = WdfRequestRetrieveInputBuffer(
UserModeMemLockRequest,
UserModeMemLockParams.Parameters.DeviceIoControl.InputBufferLength,
&pUserModeBuff,
&UserModeMemLockInputBufSize);

if( !NT_SUCCESS(ntStatus) ){
DbgPrint("WdfRequestRetrieveInputBuffer is failed\n");
*pDataLockedMemSize = 0;
goto EXIT;
}

if(pUserModeBuff == NULL){
DbgPrint("WdfRequestRetrieveInputBuffer is failed\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INSUFFICIENT_RESOURCES;
goto EXIT;
}

ulNoOfPages=UserModeMemLockParams.Parameters.DeviceIoControl.InputBufferLength/PAGE_SIZE;

if((UserModeMemLockParams.Parameters.DeviceIoControl.InputBufferLength%PAGE_SIZE)!= 0){
ulNoOfPages++;
}

```

Re: Getting same physical address for 2 different user addresses.

Re: Getting same physical address for 2 different user addresses.

```
ntStatus = WdfRequestRetrieveOutputBuffer(
UserModeMemLockRequest,
(ulNoOfPages*sizeof(LOCK_USER_MODE_MEMORY_OUTPARAMS)),
&pUserModeMemLockOutputBuffer,
&UserModeMemLockOutputBufSize);

if( !NT_SUCCESS(ntStatus) ){
DbgPrint("WdfRequestRetrieveOutputBuffer is failed\n");
*pDataLockedMemSize = 0;
goto EXIT;
}

if(pUserModeMemLockOutputBuffer == NULL){
DbgPrint("WdfRequestRetrieveOutputBuffer is failed\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INSUFFICIENT_RESOURCES;
goto EXIT;
}

if(UserModeMemLockParams.Parameters.DeviceIoControl.OutputBufferLength<(ulNoOfPages*sizeof(LOCK_USER
DbgPrint("Buff size is small\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INVALID_PARAMETER;
goto EXIT;
}

pLockUserModeMemOutParams=
(PLOCK_USER_MODE_MEMORY_OUTPARAMS)pUserModeMemLockOutputBuffer;

kCurrentIrql = KeGetCurrentIrql();
KeLowerIrql(APC_LEVEL);

pTempUserModeBufStart = (UCHAR *)pUserModeBuff;

for(ulIndex=0,ulSize=UserModeMemLockParams.Parameters.DeviceIoControl.InputBufferLength;ulIndex<ulNoOfPa
DbgPrint("ulIndex:%u\n",ulIndex);
if(ulSize>=4096){
pDmaMdl = IoAllocateMdl((PVOID)
pTempUserModeBufStart,
4096,
FALSE,
FALSE,
NULL //Irp OPTIONAL);

if(pDmaMdl == NULL){
DbgPrint("IoAllocateMdl failed\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INSUFFICIENT_RESOURCES;
goto RAISE_IRQL;
}
}
```

Re: Getting same physical address for 2 different user addresses.

Re: Getting same physical address for 2 different user addresses.

```
MmProbeAndLockPages(pDmaMdl,KernelMode ,IoModifyAccess);
//UserMode
pTempLockedPage =
MmMapLockedPagesSpecifyCache(pDmaMdl,KernelMode,MmCached,NULL,FALSE,NormalPagePriority);

if(pTempLockedPage == NULL){
MmUnlockPages(pDmaMdl);
IoFreeMdl(pDmaMdl);
DbgPrint("IoAllocateMdl failed\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INSUFFICIENT_RESOURCES;
goto RAISE_IRQL;
}

UserModeMemLockPhyAddr=MmGetPhysicalAddress(pTempLockedPage);

//for each page fill the following structure

pLockUserModeMemOutParams[ulIndex].ulNoOfBytes = 4096;
pLockUserModeMemOutParams[ulIndex].pCurrentPageMdl=(ULONG *)pDmaMdl;
pLockUserModeMemOutParams[ulIndex].pMappedLocked=(PVOID)pTempLockedPage;
pLockUserModeMemOutParams[ulIndex].ulUserModePhysicalAdd =
UserModeMemLockPhyAddr.LowPart;

pTempUserModeBufStart+=4096;
ulSize-=4096;
}
else if(ulSize>0 && ulSize<4096) {
pDmaMdl = IoAllocateMdl(
(PVOID)
pTempUserModeBufStart,
ulSize,
FALSE,//SecondaryBuffer,
FALSE,//ChargeQuota,
NULL //Irp OPTIONAL );

if(pDmaMdl == NULL) {
DbgPrint("IoAllocateMdl failed\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INSUFFICIENT_RESOURCES;
goto RAISE_IRQL;
}

MmProbeAndLockPages(pDmaMdl,KernelMode ,IoModifyAccess);

pTempLockedPage =
MmMapLockedPagesSpecifyCache(pDmaMdl,UserMode,MmCached,NULL,FALSE,NormalPagePriority);

if(pTempLockedPage == NULL){
MmUnlockPages(pDmaMdl);
```

Re: Getting same physical address for 2 different user addresses.

Re: Getting same physical address for 2 different user addresses.

```
IoFreeMdl(pDmaMdl);
DbgPrint("IoAllocateMdl failed\n");
*pDataLockedMemSize = 0;
ntStatus = STATUS_INSUFFICIENT_RESOURCES;
goto RAISE_IRQL;
}

UserModeMemLockPhyAddr=MmGetPhysicalAddress(pTempLockedPage);

pLockUserModeMemOutParams[ulIndex].ulNoOfBytes = ulSize;
pLockUserModeMemOutParams[ulIndex].pMappedLocked = (PVOID)pTempLockedPage;
pLockUserModeMemOutParams[ulIndex].pCurrentPageMdl = (ULONG *)pDmaMdl;
pLockUserModeMemOutParams[ulIndex].ulUserModePhysicalAdd =
UserModeMemLockPhyAddr.LowPart;

pTempUserModeBufStart+=ulSize;
ulSize-=ulSize;

}
}

pDevExt->ulPageNeeded =ulNoOfPages;
*pDataLockedMemSize
=UserModeMemLockParams.Parameters.DeviceIoControl.OutputBufferLength;

RAISE_IRQL:
KeRaiseIrql(kCurrentIrql,&kOldIrql);
EXIT:
DbgPrint("<---%s\n",__FUNCTION__);
return ntStatus;
}
```

Our issue is, UserModeMemLockPhyAddr.LowPart is the same for both readbuf and writebuf.

Thanks,
Saravana

.