

## မာတိကာ

	Page
အခန်း (၁) ADB Commands အသုံးပြုရန်ပြင်ဆင်ခြင်း	7 to 10
အခန်း (၂) ADB Commands များကိုလေ့လာခြင်း	11 to 17
အခန်း (၃) ADB Commands များကိုအသုံးချခြင်း	18 to 27
အခန်း (၄) အဆင့်မြင့် ADB Commands များ	28 to 46
အခန်း (၅) Android Terminal အသုံးပြုခြင်း	47 to 64
အခန်း (၆) Fastboot Commands အသုံးပြုပုံ	70 to 82
အခန်း (၇) Adb Commands ဖြင့်လက်တွေပြသုနာ ဖြေရှင်းနည်း	83 to 103
အခန်း (၈) MTK Chips Device များအတွက် Scatter ဖိုင်ထုပ်သူခြင်း	140 to 106

## Preface

Andorid နဲ့ပတ်သက်တဲ့စာအုပ်တွေအပြင်မှာ အများကြီးရှိနေပေမဲ့ အခုလို ADB Command နဲ့ ပတ်သက်ပြီး သိသနစာအုပ်တွေမရှိသေးဘူးထင်တာကြောင့် ဒီစာအုပ်ကိုရေးဖြစ်သွားတာပါ။ ကျွန်တော် ဒီစာအုပ်ကို ရေးထားတာ တစ်နှစ်နှီးပါး ကြောမြင့်ခဲ့ပါပြီ။ စာအုပ်အနေနဲ့ မထွက်တော့ဘူးလို့တောင် ဆုံးဖြတ်ခဲ့ပြီးပါပြီ။ ဒါပေမဲ့ မိတ်ဆွေတွေက တိုက်တွန်းတာကြောင့် ထုတ်ဝေမှု ဖြစ်လာခဲ့ပါတယ်။ ဒီစာအုပ်မှာ Andorid စနစ်သုံး ADB Command တွေအကြောင်းချဉ်း သက်သက် ရေးထားတာပါ။ Andorid ကို စိတ်ဝင်စားသော လေ့လာကာစ၊ လေ့လာနေခဲ့သူတွေ အတွက်တော့ အထောက်အကူဗြာပြု လက်ခွဲမှတ်စုစာအုပ် (Hand Book) အနေနဲ့ ရည်ရွယ်တာပါ။

ဒီစာအုပ်မှာ တစ်ချို့ မပြည့်စုံတဲ့တာတွေ၊ အသေးစိတ်မကျတာတွေ ရှိပါတယ်။ ကျွန်တော့အနေနဲ့လည်း Hand Book ပုံစံဖြစ်တာကြောင့် အသေးစိတ်မရေးဖြစ်ခဲ့တာပါ။ တစ်ချို့ Command တွေလည်း ကျွန်နေခဲ့မှာပါ။ အတတ်နှင့်ဆုံး မသိမဖြစ် အခြေခံအသုံးများတဲ့ Command တွေကိုတော့ မကျွန်ခဲ့ရလေအောင် ထည့်ရေးပေးထားပါတယ်။ ဒီစာအုပ်ရေးထားတာကြောနေပြီ ဖြစ်တာကြောင့် တစ်ချို့ ခေါင်းစဉ်တွေ ဟာ လက်တွေနဲ့ မကိုက်ညီဘူးလို့ ထင်ကောင်းထင်နှင့်ပါတယ်။ ဥပမာ - Root Access ယူတဲ့ ခေါင်းစဉ်ဆုံး အခုနောက်ပိုင်းထွက်ဖုန်းတွေနဲ့ မကိုက်ညီပါဘူး။ စာရေးတဲ့ အချိန်တုန်းတော့ အဆင်ပြေပါတယ်။ ကိုက်ညီမှုလည်း ရှိပါတယ်။ သို့ပေမဲ့ သဘောတရားလောက်ပဲသိဆိုပြီး မပြင်ဘဲ၊ မပယ်ဘဲ စာအုပ်ထဲ ထည့်ပေးလိုက်ပါတယ်။ ကျွန်တဲ့ အခြားခေါင်းစဉ်တွေကတော့ Android ဆိုတဲ့ စနစ်ကြီးလည်ပတ်နေသမှု အသုံးဝင်မှာပါ။

ADB (Android Debug Bridge) ကိုပဲ သီးသန်ရေးရတဲ့ ရည်ရွယ်ချက်ကတော့ - ပထမအချက်အနေနဲ့ ကျွန်တော်ကိုယ်စိတ်ဝင်စားလို့ ဒုတိယအချက် ကျွန်တော်သိတာကို တစ်ခြားသူတွေကိုလည်း သိစေချင်လို့ ဖြစ်ပါတယ်။ ခြင်းချက်အနေနဲ့ စာအုပ်ရဲ့နောက်ဆုံးမှာ China Tablet တွေကို Firmware ရေးနည်းဆိုပြီး ခေါင်းစဉ်နဲ့ Firmware ရေးနည်းကိုတွေ့ရင် ADB Command နဲ့ ဘာမှမဆိုင်ဘဲ ထည့်ရေးထားတယ်မထင်ပါနဲ့။ ကျွန်တော်က စာဖတ်သူတွေကို တတ်နိုင်သလောက်အကုန်ပေးချင်လိုပါ။ Tablet Firmware ရေးဖို့ နည်းလမ်းတွေကိုလည်း သေသေချာချာ

သီဟစိုး

ရှင်းပြန့် ကျွန်းနေပါသေးတယ်။ ဆက်ရေးဖို့လည်း ကြိုးစားနေပါတယ်။

ဒီစာအုပ်ဟာ Android စနစ်မှာသုံးတဲ့ ADB Command တွေအတွက်  
ပြီးပြည့်စုတဲ့ စာအုပ်တစ်အုပ်မဟုတ်တာကြောင့် စာဖတ်သူတွေအနေနဲ့ ဒီထက် ပို  
သိအောင် ကြိုးစားပါ၊ လေ့လာပါလို့ တိုက်တွန်းပါရစေ။ ဒီစာအုပ်က ADB အခြေခံ  
စာအုပ်သာ ဖြစ်တာကြောင့် ဒါကိုအခြေခံပြီး စိတ်ပါဝင်စားရင် စိတ်ပါဝင်စားသလောက်  
ဆက်ပြီး လေ့လာကြပါ။ လေ့လာရင်း ပိုသိချင်တာရှိခဲ့လျှင်ဖြစ်ဖြစ်၊ မရှင်းလင်းဘူး  
ထင်တာဖြစ်ဖြစ် ကျွန်းတော်ဆီကို နည်းလမ်းအမျိုးမျိုးနဲ့ ဆက်သွယ်မေးမြန်း၊  
အကြံပြနိုင်ပါတယ်။ ကျွန်းတော်သည်လည်း Expert တစ်ယောက်မဟုတ်သည့်အတွက်  
အရေးအသားမရှင်းလင်းခြင်း၊ တင်ပြန္တေးနေးပုံမပြပြစ်ခြင်းများ ရှိနိုင်ပါတယ်။  
နားလည်ပေးကြပါ၊ သည်းခံ ခွင့်လွှာတ်ပေးကြပါလို့လည်း ကျွန်းတော်ရေးတဲ့စာတွေ  
မဖတ်ခင် ကြိုတင်ခွင့်တောင်းပါရစေ ခင်ဗျာ။ အားလုံးပဲ ကြိုစာအုပ်ကြောင့် အသိ  
ပညာတစ်စုံတစ်ရာ ဒါမှုမဟုတ် အတွေ့အကြုံ တစ်ခုခုရတယ်ဆိုရင်ပဲ ကျွန်းတော်  
ဒီစာအုပ်ကို ရေးရကျိုး၊ ထုတ်ရကျိုးနှင်းတယ်လို့ ခံယူပါတယ်။ အားလုံး နည်းပညာ  
ရေစီးကြောင်းနဲ့အတူ လိုက်ပါစီးမော်နိုင်ပါစေလို့ ဆုတောင်းပေးလိုက်ပါတယ်။

သီဟစိုး

## အခန်း (၁)

# ADB Commands အသုံးပြုရန်ဖြင့်ဆင်ခြင်း

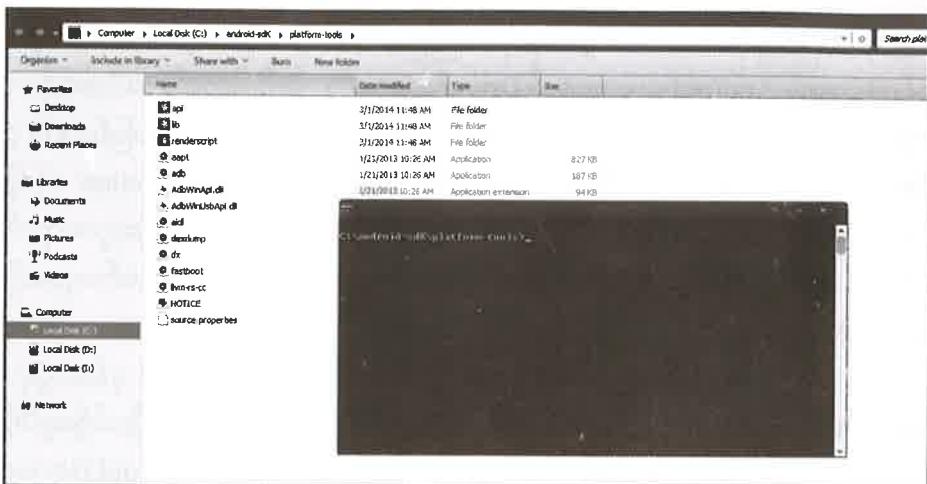
ADB ဆိုတာ ကတေသာ ? Android Debug Bridge ရဲ့ အတိုကောက်ပဲဖြစ်ပါတယ်။ ကွန်ပျူးတာနဲ့ Android Device တွေ ချိတ်ဆက်တဲ့အခါ ကွန်ပျူးတာကနေ ADB Driver ကိုသိနေဖို့လိုပါတယ်။ သိနေမှုသာလျှင် ADB Command တွေကို အသုံးချု နိုင်မှုဖြစ်ပါတယ်။ မိဘိုင်း ဖုန်းအမိုးအစား အများအပြားရှိနေတာနဲ့အမျှ သူတို့နဲ့ ပတ်သက်တဲ့ သီးသန့် ADB Driver တွေကိုလည်း တင်ပေးဖို့ လိုပါတယ်။ ကျွန်တော် ဒီစာအုပ်ရဲ့ နောက်မှာ DVD အခွဲနဲ့ လိုအပ်မယ်ထင်သမှု အဗုံး Drivers တွေကို ထည့်ပေးထားပါတယ်။

Android ဆိုတာ Google ရဲ့မူပိုင်ပြစ်တာကြောင့် Driver တင်တဲ့အခါ သီးခြား သူတို့ရဲ့ကိုယ်ပိုင် Driver တွေတင်ရင်လည်းရသလို Google Driver ကိုပဲ တင်ရင်လည်း ရပါတယ်။ ဘာလို့ အခုလို အကြံ့ောက်ပေးရလဲဆိုတော့ အခုနောက်ပိုင်း ဝင်တဲ့ တရုတ် Android တွေအတွက် Driver ကရှာရတာ ခက်ပါတယ်။ အဲဒီအချိန်မှာ ကျွန်တော်တို့က Google Driver ကိုပဲ တင်ပေးလိုက်ရှုနဲ့ တရုတ်ဖုန်းတွေမှာဖြစ်တတ်တဲ့ Driver ပြဿနာကိုဖြေဖြင့်ပြီးသားဖြစ်သွားလိုပါပဲ။ PdaNet | Easy Tether တို့လို့ Android နဲ့ ကွန်ပျူးတာ ချိတ်ဆက်ဆော့ပဲတွေဟာလည်း Android ဖုန်းတွေအတွက် လက်ခွဲ တစ်ခွဲဖြစ်ပြီး သူတို့ကိုတင်လိုက်တာနဲ့ ဘယ် Driver မှ ထပ်တင်စရာမလို တော့လောက်အောင် ပြည့်စုံပြီးသားဖြစ်နေလိုပါ။

မိမိ ကွန်ပျူးတာမှာ Adb Driver တင်ထယ်ဆိုတာ Android ဖုန်းတွေမှာ ပါတဲ့ Adb interface ကို ကွန်ပျူးတာကနေ Adb ဆော့ပဲကိုအသုံးပြီးထိန်းချုပ်ဖို့ အတွက် တင်ပေးရတာပါ။ Adb Driver တင်ပြီးပြီခိုရင်တော့ မိမိ Android Device ကို Command ကနေအသုံးပြုဖို့အတွက် Adb ကိုထပ်ပြီး Install လုပ်ဖို့လိုပါသေးတယ်။ Adb ဆိုတာကတော့ Google ကနေ Develop လုပ်ထားတဲ့ Android-SDK မှာ တစ်ခါတည်းပါလာပြီးသား Toolkit တစ်ခွဲပါပဲ။ Adb ဟာ ပုံမှန်အားဖြင့်တော့ Command Prompt ထဲကနေပဲ အသုံးပြုနိုင်ပါတယ်။ အခုနောက်ပိုင်းတော့ Adb ကို GUI ကနေအသုံးချိန်အောင်လုပ်ထားတဲ့ Tools တွေ ထွက်ပေါ်နေပါပြီ။ Adb ကို Install လုပ်ဖို့အတွက်တော့ Google က သူရဲ့ Android SDK ကို ဒေါင်းလုတ်ဆွဲပြီး

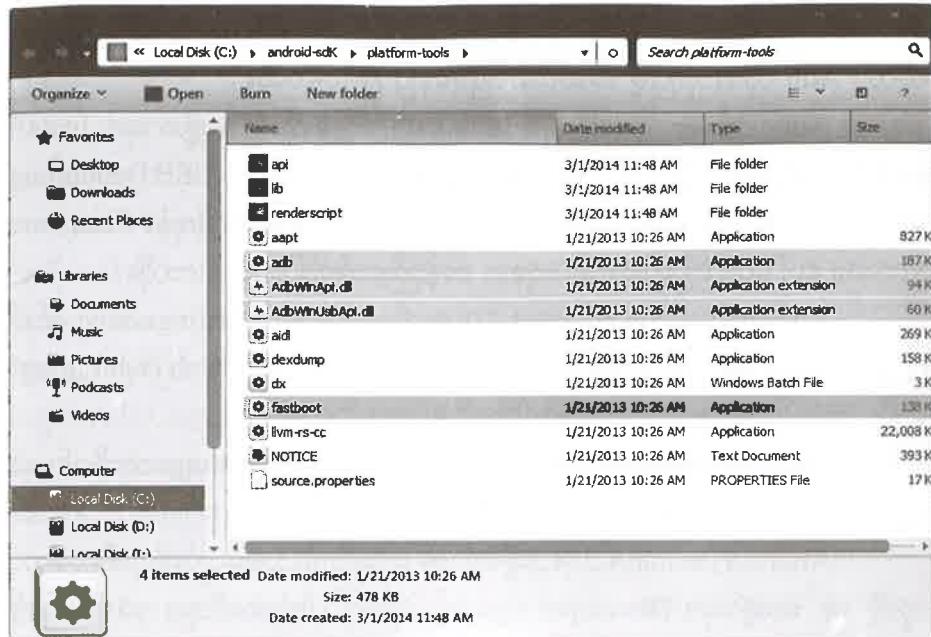
အသုံးပြန်စီမံခိုင်းပါတယ်။ ဒီမှာတော့ ADB Install လုပ်ပုဂ္ဂို သမားရှိုးကျနည်းရော၊ အလွယ်နည်းလမ်းပါ ပြောပြသွားပါမယ်။ ပထမအနေနဲ့ Google ကပေးထားတဲ့ Android SDK ကနေ ADB ကို Install လုပ်တာ ပြောပြပါမယ်။

ပထမဆုံးအနေနဲ့ ဒေါင်းလုပ်ခွဲထားတဲ့ Android SKD ကို C:\ အောက် မှာဖြည့်ချပါမယ်။ Android-sdk ဆိုတဲ့ ဖိုလ်ဒါတွေပါလိမ့်မယ်၊ ဖွင့်လိုက်ပါ။ နောက်ထပ် Platform-Tools ဆိုတာ ထပ်တွေပါမယ် ဆက်ပြီးဖွင့်ကြည့်လိုက်ပါ။ Adb ဆိုတဲ့ .exe ဖိုင်လေးတွေပါလိမ့်မယ်၊ Google ကတော့ Adb ကိုအသုံးပြုမယ်ဆို platform-tools ဖိုလ်ဒါ ထဲမှာ Shift ကိုကိုဖို့ right click လုပ် "Open command window here" ဆိုတာကို နှိပ်ပြီး Adb ကိုစတင်အသုံးပြန်စိုင်ပါပြီးလို့လမ်းညွှန်ထားပါတယ်။ တကယ်လို့ မိမိကွန်ပျူးတာက Windows XP ဆိုရင်တော့ အဲဒီ Function မပါဘူးပေါ့။ အဲဒီ Function ကိုသုံးချင်ရင် registry editor မှာသွားပြင်ပေးရပါမယ်။



Windows XP အတွက် registry မှာသွားပြင်ပေးလို့ ရတယ်ထားဦး၊ ကိုယ်က Adb ကိုသုံးမယ်ဆိုတိုင်း အမြဲတမ်း c:\android-sdk\platform-tools ထဲဝင်ဝင်ပြီး သုံးနေရတော့ မလိုအပ်ဘဲကြာနေသလိုဖြစ်နေမှာပေါ့။ အဲဒီတော့ ကျွန်တော်တို့က Adb ကို အလွယ်တကူအသုံးပြန်စေအင် Windows ရဲ့ path ထဲ သွားထည့်ပေးထားလိုက်ရင် ပိုကောင်းတာပေါ့။ အဲဒီဆိုရင်တော့ Adb သုံးမယ်ဆို ကွန်ပျူးတာကနေ

Command Prompt ကိုတန်းခေါ်ပြီး တစ်ခါတည်းတန်းသုံးနှင့်ပါတယ်။ အဲဒါကတော့ ဒုတိယနည်းပေါ့။



လိုအပ်တဲ့ဖိုင်လေးဖိုင်ကို c:\android-sdk\platform-tools ထဲကနေ ကော်ပီကူးပြီး C:\Windows ဖိုလ်ဒါအောက်ထဲမှာ သွားထည့်ပေးထားလိုက်ရုံပါပဲ။ ဒါကလွယ်တာပေါ့၊ ဘာလို ဒီနည်းပဲမပြောလဲလို မေးစရာ ရှိလာပါလိမ့်မယ်။ အလှယ်နည်းလမ်းနဲ့ သမားရိုးကျေနည်းလမ်းနဲ့မှာ အားနည်းချက်အားသာချက်ပော့ ရှိစွမ်ပါ။ ပုံမှန်အတိုင်း android-sdk\platform-tools အောက်ကနေသုံးနေမယ်ဆို Google ကအမြဲတမ်း SDK ကို update ဖြစ်အောင် လုပ်ပေးထားလို ကိုယ် adb ကိုအသုံးပြုတဲ့အခါ "adb is out of date" "can not start android daemon" ဆိုတဲ့ error ဖျိုးတွေရမှာမဟုတ်ပါဘူး။ တကယ်လို ကိုယ်က ဒုတိယနည်းကိုသုံးမယ်ဆိုရင်တော့ အပေါ်ကလို error တွေရင် ကိုယ့်ရဲ့ Adb.exe ဖိုင်လေးတစ်ခုတည်းကို ပားရှင်းအမြှင့်လေးဒေါင်းပြီး အစားထိုးပေးဖို့လိုအပ်ပါတယ်။ Adb Fixer ကိုသုံးပြီး လည်း ဖြေရှင်းနိုင်ပါပြီ။

Adb ကောင်းကောင်းမွန်မှန်အလုပ်လုပ်နိုင်ဖို့ adb.exe ၊ AdbWinApi.dll ၊ AdbWinUsbApi.dll ဆိုပြီး စုစုပေါင်း(၃)ဖိုင်ပလို အပ်တာပါ။ fastboot.exe ကတော့ Adb နဲ့ဆင်တူ အသုံးများတဲ့ Fastboot Command တွေ အကြောင်းကိုပါ နောက်ပိုင်းမှာပြောပြသားချင်လို တစ်ခါတည်း ကူးခိုင်းခဲ့တာပါ။ အခုဆိုရင် ကျွန်တော် တို့ဟာ Adb ကိုအသုံးပြုဖို့ အဆင်သင့်ဖြစ်နေပြီ ဖြစ်ပါတယ်။

Adb Driver တင်ပြီးပြီး adb.exe ကိုလည်းကွန်ပျုံတာမှာ Install လုပ်ပြီးသွားပြီဖြစ်တာမို့ ဖုန်းကို ကွန်ပျုံတာနဲ့ ချိတ်ဆက်ဖို့အတွက် USB Debugging Mode ကို on ပေးရပါမယ်။ မိမိဖုန်းမှာ အသုံးပြုနေတဲ့ Android Firmware Version ပေါ်မှုတည်ပြီးပြုလုပ်ပေးရတာ အနည်းငယ်ကွဲပြားနိုင်ပါတယ်။

Android version 2.0 ကနေ 2.3 အထိကတော့ (၁) Settings ထဲသို့ ဝင်ပါ၊ (၂) Applications ကိုဆက်နှုပ်ပါ၊ (၃) Development (၄) "Usb Debugging" ဆိုပြီး အဆင့်ဆင့်သွားပေးရမှာဖြစ်ပါတယ်။

Android version 3.0 ကနေ 4.1 အထိကတော့ (၁) Settings ထဲကိုဝင်မယ်၊ (၂) Developer Options၊ (၃) USB Debugging ပါ။

Android version 4.2၊ 4.3 နဲ့ နောက် ဟားရှင်း အသစ်တွေ အတွက်ကတော့ သူတို့ က စစ်ဆေးမှာ Developer options ဆိုတာ ပါမလာပါဘူး။ အဲဒါကြောင့် အရင်ဆုံး Developer options ကို သွားဖွင့်ပေးဖို့ လိုပါတယ်။ အရင်ဆုံး နံပါတ် (၁) အနေနဲ့ Settings ထဲကိုဝင်မယ်၊ (၂) About Device၊ (၃) "Build Number" ဆိုတာကို မြန်မြန်လေး (၅) ချက်လောက် ထောက်လိုက်ပါ အဲဒါအချိန်မှာ Developer options ကို ဖွင့်ပြီးပါပြီ ဆိုတဲ့ မတ်ဆောင်းပြုလာပါလိမ့်မယ်။ (၄) Back ကိုကို ပြန်နိုပြီး ထွက်လိုက်ပါ၊ (၅) အခုမှာပေါ်လာတဲ့ Developer options ကို နှုပ်ပါ၊ (၆) "Developer Options" ဆိုတာလေးကိုဖွင့်ပေးပြီး "USB Debugging" ဆိုတာလေးကို အမှန်ခြစ်လိုက်ပါ။

အခုဆိုရင် ကျွန်တော်တို့လေ့လာချင်တဲ့ Adb Command တွေအကြောင်း ကိုလေ့လာလို့ရသွားပါပြီ။ Adb မှာ Command တွေဘယ်လောက်များပြီး ဘယ်လောက် အသုံးဝင်တယ်ဆိုတာကို ဆက်ပြီးလေ့လာ ကြည့်ပါလို့တိုက်တွန်း အကြံပေးပါရစေ ခင်ဗျာ။

## အခန်း (၂) ADB Command များကိုလေ့လာခြင်း

အခု ပထမဆုံးအနေနဲ့ Adb Command တွေကို အသုံးမပြုခင် သူတို့ကို ဘယ်နေရာမှာသုံးတယ် ဘယ်လိုအသုံးပြုရတယ်၊ ဘယ်လို အသုံးဝင်တယ်ဆိုတာတွေကို အသေးစိတ် ပြောပြချင်ပါတယ်။ ဒါမှာလည်းကိုယ်တကယ်အသုံးချတဲ့အခါ ဘာကြောင့် သုံးရတယ်ဆိုတာ ကြိုတင်သိစေချင်လိုပါ။ နောက်ပြီးတော့ Adb မှာအသုံးပြုမယ့် Commands တွေကိုလည်း တစ်နေရာတည်းမှာစုတော်းပေးချင်တာလည်း ပါပါတယ်။ မေ့နေရင်ဖြစ်ဖြစ် အလွယ်တက္ကပြန်ကြည့်လို ရအောင်ပါ။ ကျွန်တော်အခု ရေးပြတာ တွေက Adb Command တွေကို စတင်လေ့လာသူ Beginner Level နဲ့ Intermediate Level တွေအတွက် ရည်ရွယ်ပါတယ်။ ကျမ်းကျင်ပြီးသားသူတွေအတွက်တော့ မှတ်စု အနေနဲ့ သိမ်းထားချင် သိမ်းထားနိုင်ပါတယ်။

အရင်ဆုံး Adb Command မှာ ဘာတွေသုံးလိုရတယ်ဆိုတာကို Command Prompt မှာ adb help လို ရိုက်ပြီးကြည့်ပါမယ်။ တစ်ဖက်စာမျက်နှာကပုံမှာ ကြည့်လိုက်ပါ အရမ်းကို ရှည်လျားတာတွေ့ပါလိမ့်မယ်။

အဲဒီတော့ စာဖတ်သူက ဒီလောက်ရှုပ်တာ ဘယ်လို လိုက်မှတ်မလဲလို မေးချင် လာပါလိမ့်မယ်။ ဟူတ်ပါတယ်၊ အခုလိုပုံစံနဲ့ကြည့်ရင်တော့အရမ်းရှုပ်တယ် ထင်သွား ပါလိမ့်မယ်။ ကျွန်တော်က သူတို့ကို တူရာ တူရာ အုပ်စုလေးတွေခဲ့ပြီး ယေားလေးနဲ့ ချုပြုပါမယ်။ အားလုံးကိုလည်း အလွယ်တက္က နားလည်လွယ်အောင်ပါ။ ယေားနဲ့ ချုပြေတဲ့အခါ တချို့ သုံးဖို့မလိုအပ်တာ အရမ်းသုံးခဲ့တဲ့ Command တွေကိုတော့ အနည်းငယ်ပ ရှင်းပြပေးသွားပါမယ်။ နှိမ့်ဆို စာအုပ်က အရမ်းကြီးပြီး လူပြောစရာ ဖြစ်သွားမှာစိုးလိုပါ။ ကျွန်တော်ကိုယ်တိုင်ကိုက မှတ်စုစာအုပ်လေးတစ်အုပ်စာလောက်ပ ရည်ရွယ်တာပါ။ ထည့်မရေးပေးမဲ့တဲ့ Command တွေရှိခဲ့လျှင် ကျွန်နေခဲ့လျှင်လည်း ခွင့်လွယ်ပေးပါလို ကြိုတင်တောင်းပန်ထားပါရစေခင်ဗျာ။

```

Administrator: C:\Windows\system32\cmd.exe

G:\Users\Phyo_Ng\adb help
Android Debug Bridge version 1.0.31

-a
connection
-d
e
presents,
-r
running,
-s <specific device>
the given
EMUL
-p <product name or path>

        - directs adb to listen on all interfaces for a
        - directs command to the only connected USB device
        returns an error if more than one USB device is
        - directs command to the only running emulator,
        returns an error if more than one emulator is
        - directs command to the device or emulator with
        serial number or qualifier. Overrides ANDROID_S
environment variable.
simple product name like 'somesm', or
a relative/absolute path to a product
but directory like 'out/target/product/somesm'.
If -p is not specified, the ANDROID_PRODUCT_OUT
environment variable is used, which must
be an absolute path.
Name of adb server host (default: localhost)
Port of adb server (default: 5037)
List all connected devices
C-1 will also list device qualifiers
connect to a device via TCP/IP
Port 5555 is used by default if no port number
is specified.
disconnect [:<port>]
is specified.
is specified.
        , ,
        ,
device commands:
adb push <local> <remote>
adb pull <remote> [<local>]
adb sync [-<directory>] [-<filter>]
adb shell
adb shell <command>
adb emu <command>
adb logcat [-<filter spec>]
adb forward [-<list>]

g format:
adb forward <local> <remote>
        copy file/dir to device
        copy file/dir from device
        copy host->device only if changed
        (-l means list but don't copy)
        (see 'adb help all')
        run remote shell interactively
        run remote-shell command
        run emulator console command
        View device log
        list all forwarded socket connections.
        the format is a list of lines with the following
        <serial> " " <local> " " <remote> "\n"
forward socket connections
forward specs are one of:

```

Commands	အသုံးဝင်ပုံ
Basic Adb Commands ( အခြေခံ Adb Command များ)	
adb device	ကွန်ပူးတာနှင့်ချိတ်ဆက်ထားသော Devices များကြည့်ရန်
adb reboot	ချိတ်ဆက်ထားသော Device ကို Reboot ချက်
adb reboot recovery	ချိတ်ဆက်ထားသော Device ကို Recovery Mode သို့ဝင်ခြင်း
adb reboot download	ချိတ်ဆက်ထားသော Device ကို Download Mode သို့ဝင်ခြင်း၊ ယခု Command ကို Samsung ဖုန်းများတွင်သာ အသုံးပြုနိုင်ပါသည်
adb reboot bootloader	ချိတ်ဆက်ထားသော Device ကို Bootloader Mode သို့မဟုတ် Fastboot Mode သို့ဝင်ခြင်း
Installing, Uninstalling, Updating with Apps with ADB (Adb မှုsoftware တင်ခြင်းနှင့်ဖျက်ခြင်း)	
adb install <applicationname>	Android phone ထဲသို့ Applications တွေတစ်ခါတည်း Install လုပ်တာ ဒီ Command ကိုအသုံးပြုဖို့အတွက် adb install application path ကိုရေးပြီး Enter ခေါက်ရှုပါပဲ။ ဥပမာ "adb install D:\apps\facebook.apk" လိုက်ရှုပါ။ Install လုပ်တဲ့ လုပ်ငန်းအောင်မြင်ရင် မတ်ဆောင်ရွက် Success ဆိုပြီးပြဲလာပါလိမ့်မယ်
adb install -r "application name.apk"	အရင်ရှိပြီးသား ဆော့ဝဲကိုမဖြတ်ဘဲ Upgrade လုပ်တာ
adb install -s "application name.apk"	ဆော့ဝဲ တင်တဲ့အခါ Internal Storage မှာမတင်ဘဲ External Storage မှာလှမ်းတင်တာ
adb uninstall "Package name"	ဖုန်းမှာတင်ထားတဲ့ ဆော့ဝဲကို ပြန်ဖြတ်ခြင်း။ e.g "adb uninstall com.facebook.katana" (ဖိမိရဲ့ Android Device မှာ package name ဘယ်လိုကြည့်ရတယ်ဆိုတာကို နောက်အခန်းမှာရေးပြသွားပါမယ်)
adb uninstall -k "Package name"	ဆော့ဝဲကိုဖြတ်ပစ်ချင်ပေမယ့် Data ဖိုင်တွေကို မပျက်စေချင်တဲ့အခါ မဖျက်ချင်တဲ့အခါ အသုံးပြုပါတယ်

Push and Pull Files ( ဖုန်း (Device) နဲ့ ကွန်ပျူတာ အပြန်အလှန်ဖိုင်ကူးယူခြင်း)	
adb push <local> <remote>	မိမိဖုန်း၊ Device ထဲသို့ကွန်ပျူတာမှ ဖိုင်/ဖိုလ်ဒါကူးထဲခြင်း အခု Command မှာ Local ဆိုတာက မိမိကွန်ပျူတာဖြစ်ပြီး Remote ကတော့ ကိုယ့်ရဲ Android Device ပါ။ e.g. adb push D:\songs\1.mp3 /sdcard/1.mp3 မိမိကွန်ပျူတာ D: ၏အောက်ကသီချင်းတစ်ပုံကို ဖုန်း SDcard ထဲကိုလှမ်းပို့တာပါ။ သတိထားရမှာက Windows ကွန်ပျူတာတွေမှာ D:\songs ဆိုပြီး ()ကိုသုံးပါတယ်။ Android မှာတော့ ()ကိုသုံးပါတယ်။ ကွန်ပျူတာဘက်အခြေမှာမိမိနာမည်ရေးတဲ့အခါစာလုံးအကြောင်းအသေး ပြဿနာမရှိပါဘူး။ Android မှာတော့ အားလုံးတူမှုရပါမယ်။
adb pull <remote> [<local>]	မိမိဖုန်း၊ Device ထဲမှ ဖိုင် ဖိုလ်ဒါတွေကို ကွန်ပျူတာထဲသို့ သို့မဟုတ်အခြားနေရာတစ်ခုခုသို့ကူးယူတဲ့အခါမှာ သုံးပါတယ်
Backing up System and Installed Apps ( Install လုပ်ထားသော Application များနှင့် System ကို Backup ဆွဲခြင်း)	
မှတ်ချက်။	မိမိရဲ့ ဖုန်း၊ Device နဲ့ပတ်သက်တဲ့ System + Applications တွေကို Backup မဆွဲခင် ပထမဗုံးဆုံး အနေနဲ့ Backupဆိုတဲ့ဖိုလ်ဒါ တစ်ခုနဲ့သူအောက်မှာ systemapps နဲ့ installapps ဆိုပြီး ဖိုလ်ဒါ နှစ်ခုအောက်ထားပါ။ ဒါ ဖိုလ်ဒါနှစ်ခုထဲကို ဖုန်းကနေ Backup ဆွဲမယ့် Apps တွေကူးထဲပေးရမှာပါ။
adb pull /system/app backup/systemapps	မိမိဖုန်းထဲက System apps တွေကို systemapps ဆိုတဲ့ဖိုလ်ဒါထဲကို Adb ကနေ ကူးယူဖို့အတွက်သုံးပါတယ်။
adb pull /system/app backup/installledapps	မိမိဖုန်းထဲက Installed apps တွေကို installedapps ဆိုတဲ့ ဖိုလ်ဒါထဲကို Adb ကနေကူးယူဖို့အတွက်သုံးပါတယ်။

Background Terminal (Android Device ထဲသို့ Terminal မှုပ်သုံးခြင်း)	
adb shell	Background Terminal စတင်အသုံးပြုခြင်း
exit	Background Terminal အသုံးပြုခြင်းမှုတွက်ခြင်း
adb shell <command> e.g. adb shell su	မိမိ Android ဖုန်း၊ Device ထဲသို့ Root အနေဖြင့် ဝင်ရောက်ခြင်း၊ ယခု Command အသုံးပြုဖို့အတွက် မိမိ Device၏ Root access ရှိရပါမယ်
မှတ်ချက်။	Terminal Command တွေဟု Adb Command တွေနဲ့ ကွဲလွှာပြီးမိမိရဲ့ Android Device ကိုစိုးစိတ်ပြုတဲ့ အသုံးချိန်ပါတယ်။ Adb အကြခံပြီး Terminal Command တွေအသုံးပြုပဲကို သီးသန့်တစ်ခန်းအနေနဲ့ ဖော်ပြေပေးသွားမှာပါ။
Logcat ( Log များကိုဖြန်လည်စစ်ဆေးခြင်း)	
adb logcat	မိမိဖုန်း၊ Device ရဲ့ Real Time Log တွေကို ပြန်ကြည့်ခြင်း
adb logcat > a.txt	Logs တွေကို mdc ဆိုတဲ့ ဖိုင်တစ်ခုထဲသို့ကူးလဲခြင်း

အပေါ်မှာပြခဲ့တဲ့ ဓယားကတော့လူအသုံးများဆုံး အသုံးပို့ဆုံး Adb Command တွေကို သီးသန့်ခဲ့ပြထားတာပါ။ ဒီအပြင် အခြား Adb Command တွေလည်း ရှိပါသေးတယ်။ ဥပမာ .. Adb နဲ့ ပတ်သက်ပြီး ဘယ်လိုလုပ်လိုရလဲ ဆိုပြီး သိချင်ရင် အကူအညီတောင်းတဲ့ ပုံစံ .. "adb help" လို့ ရှိက်လို့ရပါတယ်။ ကျွန်တော်တို့သုံးတဲ့ Adb ရဲ့ ဗားရှင်းနိမ့်နေရင် update လုပ်ခိုင်းပါတယ်။ အဲဒီတော့ ကိုယ့်ရဲ့ Adb Version က ဘယ်လောက်လဲသိချင်ရင် ရှိက်ရမယ့် Command ကတော့ "adb version" ပဲဖြစ်ပါတယ်။ တစ်ခါတလေ မိမိ Device မှာ USB Debugging ကို ဖွင့်ထားပေမယ့်လည်း Adb မှာ Device ကို Offline လို့ပြုရင်လည်း Adb ကို Update လုပ်ခြင်းဖြင့် ဖြော်ရှင်းပေးနိုင်ပါတယ်။

နောက်ထပ်အသုံးများတဲ့ Command (၃)ခုရှိပါသေးတယ်။ ဒါပေမဲ့ တချို့ယူ တွေက (၂)ခုကိုအသုံးပြုခဲ့ပါတယ်။ နောက်ဆုံးတစ်ခုကတော့ အမြဲလိုလို အသုံးပြုနေရတဲ့ Command ပါ။

### (၁) *adb start-server*

မိမိကွန်ပျိုတာမှာ Adb Service စတင်အလုပ်လုပ်စေဖို့အတွက် အသုံးပြုပါတယ်။

### (၂) *adb kill-server*

adb server အလုပ်လုပ်နေတာကို မလုပ်စေချင်တဲ့အခါ အသုံးပြုပါတယ်။ ကျွန်တော်တို့ တစ်ခါတလေ Adb Error တက်ရင် ဒီ Command ( J ) ခုကိုအသုံးပြုပြီး ဖြေရှင်းနိုင်ပါတယ်။

```
C:\Users\Phyo_NgeI>adb kill-server
C:\Users\Phyo_NgeI>adb start-server
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
C:\Users\Phyo_NgeI>
```

### (၃) *adb remount*

ဒို့ Android Device ရဲ့ System partition ကို read-write Permission ယူလိုက်တာပါ။ အရမ်းအသုံးများပါတယ်။

ဒီအခန်းမှာတော့ adb command တွေနဲ့ သူတို့ကိုအသုံးပြုပုံတွေကို စာအနေ နဲ့ပဲရှင်းပြပေးထားတာပါ။ နောက်အခန်းမှာ Adb Command တွေကို လက်တွေ့အသုံးချုံအနေနဲ့ ဆက်ပြီးရှင်းပြပေးပါမယ်။ မမေ့အောင် များများလေ့ကျင့်ပေးဖို့ လိုပါတယ်။

အကယ်၍ Android Device နှစ်ခုကိုတစ်ပြိုင်နက်ချိတ်ထားတယ်ဆိုရင် adb နောက်မှာ Switch တွေခြားအသုံးပြုရမှာပါ။ ***adb [-d] [-e] [-s] [-p] <command>*** ဆိုတဲ့ ပုံစံအတိုင်း အသုံးပြုရမှာပါ။

***[-d]*** ဆိုရင်တော့ Usb Cable နဲ့တို့က်ရှိက်ချိတ်ဆက်ထားတဲ့ Device အပေါ်သက်ရောက်ပါတယ်။ဘာဖြစ်လို့အခုလို ပြောရလဲဆိုတော့ Adb ဟာ Wi-fi

ကနေလည်းအလုပ်လုပ်နိုင်လိုပါ။

[ -s ] ဆိုရင် ကိုယ့်စိတ်ကြိုက် Device ကို ရွှေးချယ်ဖြီး အလုပ်လုပ်ပေးနိုင်ပါတယ်။  
အသုံးပြုပုံကတော့ "adb -s <ID> <command>" ပဲဖြစ်ပါတယ်။

[ -e ] ကတော့ Adb Service အလုပ်လုပ်နေတဲ့ Android Device ပေါ်အကျိုး  
သက်ရောက်ပါတယ်။

[ -p ] ကတော့ Android Device ရဲ့ Product Name နဲ့ အလုပ်လုပ်ပေးတာပါ။  
များသောအားဖြင့် Device တွေကို Product Name သတ်မှတ်ပေးထားခြင်း မရှိလို့  
[ -p ] ကိုအသုံးမပြုကြပါဘူး။

အခန်း (၃)

## ADB Command များကို အသုံးချခြင်း

အရှေ့က အခန်းနှစ်ခုမှာတော့ Adb ကို Install လုပ်ပဲ၊ Adb Command တွေကို အသုံးပြုပဲ၊ အသုံးဝင်ပဲတွေကို ရှင်းပြထားခဲ့ပြီးပါပြီ။ ယခုအခန်းမှာတော့ Adb Command တွေရဲ့ အသုံးပြုပဲ၊ အသုံးချပဲကို လက်တွေ့ပြပေးသွားမှာပါ။ အဲဒါမှုလည်း ပြန်လေ့ကျင့်တဲ့အခါ လွှာယ်လွှာယ်ကူကူဖြစ်သွားမှာပါ။

ယခုလိုရှင်းပြပေးတဲ့အခါ အမှားပါခဲ့ရင်လည်း ခွင့်လွှာတ်ပေးဖို့ ကြိုပြီးမေတ္တာ ရပ်ခံပါရစေခင်ပျော်။

### (၁) adb devices

မိမိကွန်ပျိုတာမှာ ချိတ်ဆက်ထားသော Adb Device များရှိ၊ မရှိစစ်ဆေးဖို့ အတွက် အသုံးပြုပါတယ်။ ချိတ်ဆက်ထားခြင်းမရှိရင်ဘဲဖြစ်ဖြစ်၊ ချိတ်ဆက်ထားပြီး Adb Driver မသိတာဘဲဖြစ်ဖြစ်၊ USB Debugging Mode ကို မဖွင့်ထားရင်ဖြစ်ဖြစ် "List of devices attached" ရဲ့အောက်မှာ ဘာမှုပြုမှာမဟုတ်ပါဘူး။ တကယ်လို့ ကိုယ်က Device ချိတ်ဆက်ထားပြီးသားကို Adb ကနေ Detect မသိရင်တော့ အပေါ်ကပြောတဲ့ တစ်ခုခုကိုပြန်စစ်ပေးပါ။ ချိတ်ဆက်ထားတာရှိတယ်ဆိုရင် "List of devices attached" မှာ Device ကို ပြပေးသွားမှာပါ။ Device ရဲ့အရှေ့က 4d0036 ဆိုတာတွေကတော့ Device ID ပါ ကိုယ်ကွန်ပျိုတာမှာ Android Device (J) ခုတပ်ထားရင် ID (J)ခုပြပေးမှာပါ။

```
C:\Users\Phyo Ngei>adb devices
List of devices attached
4d003662511b602d          device
```

### (၂) adb install

Adb Command ကနေ မိမိဖုန်း၏ Device ထဲကို ဆော့လှမ်းသွင်းတဲ့ အခါမှာ အသုံးပြုပါတယ်။ အခုလို ဆော့လှမ်းသွင်းတဲ့အခါ ကိုယ်အသုံးပြုချင်တဲ့

apk ဖိုင်လမ်းကြောင်းကို မှန်အောင်ရေးပေးရပါတယ်။ အလွယ်ဆုံးကတော့ "adb install" လိုခြုံကြပြီး Space ခြားမယ် ကိုယ့်ထည့်မယ့် ဖိုင်ကိုရှာပြီး Mouse နဲ့ ဆွဲထည့်ပေးလိုက်ပါ။ ပြီးရင် Enter ခေါက်လိုက်ပါ။ ဆော့ဝဲထည့်တာ အောင်မြင်ရင် နောက်ဆုံးမှာ Success ဆိုတဲ့ မက်ဆွဲချုပ်လာပြပါလိမ့်မယ်။ ကွန်ပူးတာမှာ Device နှစ်ခု ချိတ်ထား တယ်ဆို adb နောက်မှာ [-s] ခံပြီး အသုံးပြုပေးပါ။ မိမိဖုန်း၊ Device မှာ ကြည့်လိုက်ပါ ခုနက တင်လိုက်တဲ့ဆော့ဝဲက အသုံးပြုလိုရနေပြီဆိုတာ တွေ့ရပါလိမယ်။

```
C:\Users\Phyo NgeI>adb install "C:\Users\Phyo NgeI\Desktop\Audio Manager.apk"
2642 KB/s (906540 bytes in 0.335s)
pkg: /data/local/tmp/Audio Manager.apk
Success
```

ကျွန်တော်ကတော့ ကွန်ပူးတာ Desktop ပေါ်မှုရှိတဲ့ Audio Manager ဆိုတဲ့ ဆော့ဝဲကိုယ့်လုမ်းတင်ပြတာပါ။ အဲမှာ သတိထားရမှာက ကျွန်တော်က adb install ဆိုပြီးနောက်က ဖိုင်လမ်းကြောင်းကိုတန်းပေးပြီး ဆော့ဝဲတင်ထားတာမလို Internal Storage မှာတင်မယ်၊ sd card မှာတင်မယ်ဆိုတာ မရွေးပေးထားပါဘူး System က သူဘာသာ သူရွှေးချယ်ပြီးတင်ပေးသွားမှာပါ။

```
adb install -r D:\Facebook_8.0.0.0.4.eh_2.apk
```

အရင်ရှုပြီး Facebook ဗားရှင်းအဟောင်းကိုမဖြေတဲ့ အသစ်ထွက်ဗားရှင်းကို Upgrade လုပ်လိုက်တာပါ။ အရင်က Data တွေမဖျက်ဘဲ အသစ်ထပ်တင်ချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
adb install -s D:\Zapya.apk
```

Adb Install ရဲနောက်မှာ (-s) Switch ကိုသုံးခြင်းဖြင့် အခုတင်မယ့် Zapya ဆော့ဝဲကို sd card ထဲကိုတန်းပြီး Install လုပ်ပေးပါတယ်။ တစ်ခုသတိထားရမှာက ဆော့ဝဲသွင်းတဲ့အခါ မြန်မာလိုပေးထားတဲ့ Folder name အောက်ကဖိုင်တွေ၊ မြန်မာလိုနာမည်ပေးထားတဲ့ဖိုင်တွေကို တင်လိုမရနိုင်ပါဘူး။

## (၃) adb uninstall

Adb Uninstall ကိုတော့ လက်ရှိအသုံးပြုနေတဲ့ Application တွေ ဆော့ဝဲတွေကို အသုံးမပြုချင်တော့တဲ့အခါမှာ ပြန်ပြီးဖြူတ်ပစ်ဖို့အတွက်သုံးပါတယ်။ သူကို အသုံးပြုပုံကတော့ အရှေ့ကအခန်း (j) မှာရေးပြုခဲ့သလို "adb uninstall <PACKAGE>" နဲ့ "adb uninstall [-k] <PACKAGE>" နှစ်ခုပြုပါတယ်။ ကိုယ်ဖြူတ်ပစ်မယ့် ဆော့ဝဲရဲ့ Data တွေကိုမဖျက်ပစ်ဘဲ ချွန်ထားချင်တဲ့အခါ [-k] ကိုသုံးပြီးဖျက်ပစ်နိုင်ပါတယ်။ မလိုအပ်ဘူးလို့ ထင်ကောင်းထင်နိုင်ပါတယ်။ ဥပမာ အခုချက်ချင်းပဲ မသုံးချင်လို့ Viber ဆော့ဝဲကိုဖျက်ပစ်ချင်နေတယ်၊ ပုံမှန်အတိုင်း -k ကို မသုံးဘဲဖျက်လိုက်ရင် နောက်တစ်ခါ ပြန်သုံးချင်လို့ Viber ပြန်တင်ရင် Activate ပြန်လုပ်ရမယ်။ အရင်က ပြောထားတဲ့ Message တွေမရှိတော့လို့ စိတ်ညွစ်ရမယ်။ ကိုယ် Download ဆွဲထားတဲ့ စတစ်ကာတွေ ပျက်ကုန်လို့ ပြန်ဖောင်းရမယ်၊ ဒါလို့ မဖြစ်အောင် ခဏနေပြန်သုံးမယ်ဆိုရင်တော့ Data တွေချွန်ထားသင့်ပါတယ်။

```
C:\Users\Phyo Ngei>adb uninstall com.facebook.katana
Success
```

အပေါ်က Command ကတော့ ဖုန်းထဲကနေပြီးတော့ Facebook ဆော့ဝဲကို ပြန်ဖြူတ်ချလိုက်တာပါ။ အောင်မြင်သွားလို့ Success ဆိုတဲ့ စာသားပေါ်လာပါတယ်။

ဒီနေရာမှာ တစ်ခုပြောစရာရှိတာက ကိုယ်ဖုန်းမှာ ဘာမော့ဝဲတွေတင်ထားလဲ၊ ပြန်ဖြူတ်မယ်ဆို PACKAGE တွေကို ဘယ်လိုရှာမလဲ၊ ဒါကအရေးကြီးဆုံးပါပဲ။ PACKAGE တွေမသိဘဲ ဘယ်လိုမှ ဘယ်ဆော့ဝဲကိုမှ ပြန်ဖြူတ်ပစ်လို့ရမှာ မဟုတ်ပါဘူး။ Google Play Store မှာတော့ Package တွေကြည့်တဲ့ ဆော့ဝဲရှိပါတယ်။ ကျွန်တော် ကတော့ "adb shell pm" ဆိုတဲ့ Command နဲ့ပဲ Package တွေကို ကြည့်ပါတယ်။ ယခုတော့ အကြမ်းဖျင်းပဲပြောပြပါမယ်၊ နောက်အခန်းရောက်မှု အသေးစိတ်ရှင်းပြပါမယ်။ "pm" command က အရမ်းကိုကျယ်ပြန်လိုပါ။

```
C:\Users\Phyo_Nge1>adb shell pm list package
package:com.mobogenie.daemon
package:com.scribd.app.reader0
package:com.sec.knox.knoxappinstaller
package:com.bit.mingalar
package:com.samsung.android.snote
package:com.facebook.katana
package:com.BinarySheep.eTool
package:it.tolelab.fud
```

အပေါ်ကပဲ ကိုကြည့်လိုက်ရင် ကျွန်ုတ်က "adb shell pm list package" ဆိုတဲ့ Command ကိုသုံးပြီး မိမိဖုန်းမှာတင်ထားတဲ့ဆောင်တွေရဲ့ PACKAGE တွေကိုကြည့်ထားတာပါ။ Facebook ဆိုရင်လည်း com.facebook.katana ဆိုတဲ့ နာမည်နဲ့ ရှိနေပါတယ်။ ကျွန်ုတ်တော်တို့ ပြန်ဖြေတယ်ချင်ရင် ဒီနာမည်တွေကိုပဲ အသုံး ပြုရမှာပါ။ တစ်ခုသိထားရမှာက Package တွေအားလုံး com. နဲ့ချည်းပဲ မစတူး ဆိုတာပါပဲ .. သူတို့ဟာလည်း Web Site တွေလိုပါပဲ .. အနည်းငြားအများ ဆိုသလို Networking နဲ့ဆိုင်တဲ့ ဆောင်တွေဆို net. နဲ့စတက်ပြီး အစိုးရနဲ့ပတ်သက်ရင် gov. နဲ့စပါတယ်။

### (၅) adb push

Adb ကနေ ဖိုင် ဖိုလ်ဒါတွေကို မိမိရဲ့ Android Phone Device ထဲကို ကူးယူဖို့အတွက် အသုံးပြုပါတယ်။ သုံးရတာ လွယ်ကူသလို အသုံးလည်းဝင်တဲ့ Command တစ်ခုပါ။ push ကို အသုံးပြုမယ်ဆိုရင်တော့ "adb push <local> <remote>" ဆိုပြီး local ဖြစ်တဲ့ ကွန်ပျိုးတာတဲ့ File (or) Folder တွေရဲ့ လမ်းကြောင်းနဲ့ remote ဖြစ်တဲ့ Device မှာသွားသိမ်းချင်တဲ့နေရာ(လမ်းကြောင်း)ကို မှန်အောင်ရေးပေါ်မယ်။

```
C:\Users\Phyo_Nge1>adb push "d:\CWM Recovery 2.5.1.8.zip" "/sdcard/CWM Recovery.zip"
3655 KB/s (5188916 bytes in 1.386s)
C:\Users\Phyo_Nge1>
```

အပေါ်မှာပြထားတာကတော့ D:\ အောက်က CWM Recovery ဆိုတဲ့ ဖိုင်ကို ဖုန်းရဲ့ sd card အောက်ထဲ ကူးထည့်လိုက်တာပါ။ Adb နဲ့ ပတ်သက်တဲ့

Command တွကို "adb help" နဲ့ကြည့်တဲ့အခါ [ - ] နဲ့ပြထားတာတွေ < - > နဲ့  
ပြထားတာတွေ တွေ့ရမှာပါ။ အဲဒီအခါကျရင် < - > ထဲမှာပါတဲ့ Command တွက  
မဖြစ်မနေထည့်ပေးရမယ့် Command တွေပါ [ - ] ထဲက Switch တွကတော့  
မထည့်လည့်ရတယ် ထည့်ပေးရင်လည့်ရပါတယ် (Optional)ပေါ့။ ကိုယ်သုံးမယ့်  
Command ကို ပိုမြေး တိကျစေချင်လို့ ပိုမြေးပြည့်စုံစေချင်တဲ့အခါမျိုးမှာ ထည့်ပေး  
ရပါတယ်။

### (၅) *adb pull*

အပေါ်က Push Comand က ကွန်ပူးတာထဲကနေ ကွန်ပူးတာနဲ့ချိတ်ဆက်  
ထားတဲ့ Android Device ထဲကို ဖိုင်ကူးထည့်တာဖြစ်လို့ အခါ Pull Command  
ကတော့ ကွန်ပူးတာနဲ့ချိတ်ဆက်ထားတဲ့ Android Device ကနေ မိမိကွန်ပူးတာ  
ထဲကို ပြန်ကူးယူတာပဲဖြစ်ပါတယ်။ အပေါ်က Command နဲ့အတူတူပဲမလို့ အကျယ်တဝါး  
မရှင်းပြတော့ပါဘူး။ သတိထားစရာတစ်ခုကတော့ . . . Android Device ထဲက  
ဖိုင်၊ ဖိုလ်ဒါတွေကို ကူးယူတဲ့အခါ File Name မှန်အောင်ကူးဖိုပါပဲ၊ Android က  
File Name မှာစာလုံးအကြိုး၊ အသေးကအစအရေးပါပါတယ်။ အဲဒီကို *case sensitive*  
ဖြစ်တယ်လို့ခေါ်ပြီး Windows ကွန်ပူးတာတွေမှာတော့ File Name မှန်ဖိုပဲလိုပါတယ်။  
စာလုံးအကြိုးအသေး ပြဿနာမရှိပါဘူး။ အဲဒီတော့ ကွန်ပူးတာပေါ်မှာဆို File Name  
တူရင်ထပ်ပေးလို့မရတော့ပါဘူး၊ Android မှာတော့ File Name တူရင် အဟောင်းက  
စာလုံးအသေးနဲ့ဆို အသစ်ကို စာလုံးအကြိုးထည့်ပေးလိုက်ပါ။ ပေးလို့ရပါတယ်။  
ဗဟိုသုတေသနနဲ့သိရအောင် ရေးပေးလိုက်တာပါ ယခု Command နဲ့တော့မဆိုပါဘူး။

```
C:\Users\Phyo Ngel>adb pull "/sdcard/video/Katy Perry - Dark Horse.mp4" d:\  
3152 KB/s (71382049 bytes in 22.109s)
```

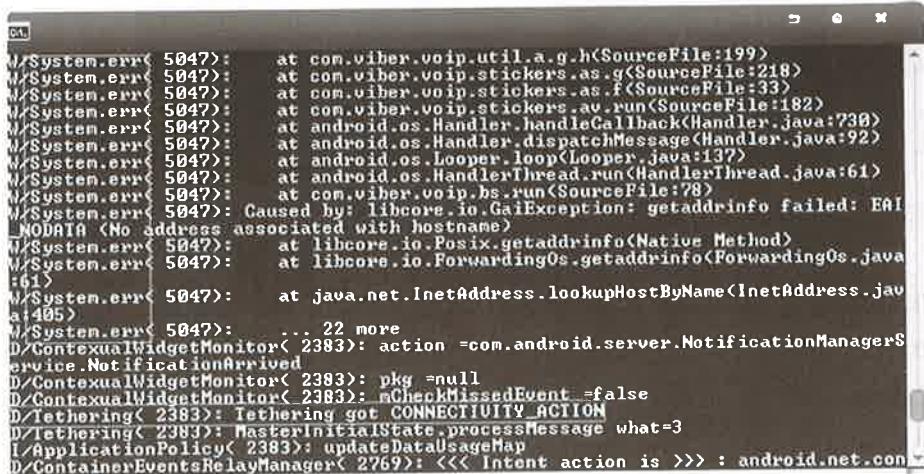
```
C:\Users\Phyo Ngel>
```

အပေါ်က Pull Command မှာတော့ ကွန်တော်က Sdcard/video/ ထဲက  
mp4 ဖိုင်ကို D:\ အောက်ကိုကူးယူလိုက်တာပါ။ File Name မှာ Space တွေပါလို့  
sd card ရဲ့အရွှေမှာ ( " ) နဲ့စပြီး mp4 ရဲ့နောက်မှာ ( " )နဲ့ပိုတ်ပေးထားတာပါ။  
အောက်တစ်ကြောင်းမှာတော့ ဖိုင်ကူးတဲ့ Transfer Rate၊ File Size နဲ့ကူးတဲ့ ကြောခိုန်  
တွေ ပြပေးပါတယ်။ ပုံမှန် Copy၊ Paste လုပ်တာထက် မြန်လည့်မြန်ပါတယ်။

အခုထပ်ပြီးတော့ မိမိရဲ့ Android Device ရဲ့ Log ဖိုင်တွေကိုကြည့်တာ ပြောပြချင်ပါတယ်။ မိမိရဲ့ Device ဟာဘယ်အချိန်မှာ ဘာဖြစ်ခဲ့လဲ ဘယ်အချိန်မှာ ဘယ်လို မက်ဆောင်တွေပြလည်း၊ ကိုယ့်ဖုန်းက Harm နေတယ်ဆိုရင်တောင် အဲဒီ အချိန်မှာ ဘာတွေကြောင့်ဖြစ်တယ်ဆိုတာ ကြည့်နိုင်ပါတယ်။ တစ်ခါတစ်လေ ကိုယ့် အသံးဝင်ပါတယ်။

### (၆) adb logcat

Adb Logcat ကို run လိုက်ရင် System က Mornitoring လုပ်ထားတဲ့ အချက်အလက်တွေများတာကြောင့် အစအဆုံး ဘယ်လိုမှုမကြည့်နိုင်ပါဘူး။ ကြည့်လို ရသလောက်ပဲ ဗဟိုသုတအနေနဲ့ ကြည့်ကြည့်တာပါ။ တကယ်တော့ ဘယ်သူမှုလည်း ဒါကိုကြည့်မှာ မဟုတ်ပါဘူး။



```

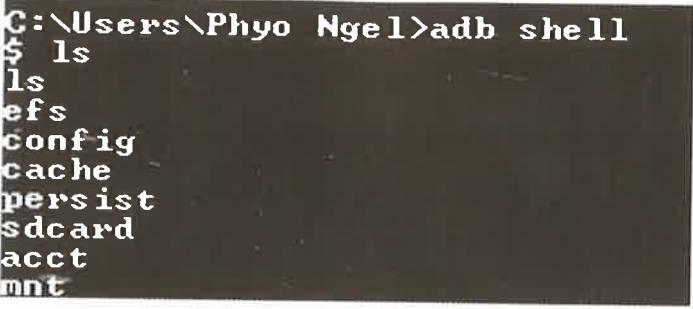
D/System.err< 5047>: at com.viber.voip.util.a.g.h<SourceFile:199>
D/System.err< 5047>: at com.viber.voip.stickers.as.g<SourceFile:218>
D/System.err< 5047>: at com.viber.voip.stickers.av.run<Sourcefile:182>
D/System.err< 5047>: at android.os.Handler.handleCallback(Handler.java:730)
D/System.err< 5047>: at android.os.Handler.dispatchMessage(Handler.java:92)
D/System.err< 5047>: at android.os.Looper.loop<Looper.java:137>
D/System.err< 5047>: at android.os.HandlerThread.run<HandlerThread.java:61>
D/System.err< 5047>: at com.viber.voip.bs.run<SourceFile:78>
D/System.err< 5047>: Caused by: libcore.io.GaiException: getaddrinfo failed: EAI_NODATA <No address associated with hostname>
D/System.err< 5047>: at libcore.io.Posix.getaddrinfo(Native Method)
D/System.err< 5047>: at libcore.io.ForwardingOs.getaddrinfo<ForwardingOs.java:61>
D/System.err< 5047>: at java.net.InetAddress.lookupHostByName<InetAddress.java:495>
D/System.err< 5047>: ... 22 more
D/ContextualWidgetMonitor< 2383>: action =com.android.server.NotificationManager$Service.NotificationArrived
D/ContextualWidgetMonitor< 2383>: pkg =null
D/ContextualWidgetMonitor< 2383>: mCheckMissedEvent =false
D/Tethering< 2383>: Tethering got CONNECTIVITY_ACTION
D/Tethering< 2383>: MasterInitialState.processMessage what=3
I/ApplicationPolicy< 2383>: updateDataUsageMap
D/ContainerEventsRelayManager< 2769>: <<< Intent action is >>> : android.net.con

```

မိမိ Android Device မှာဖြစ်နေတဲ့ ဖြစ်ခဲ့တဲ့ System Error တွေပြပေးပါတယ်။ ဖုန်းမှာ ဘာအသံးပြုခဲ့တယ်ဆိုတာ ပြပေးပါတယ်။

## (၃) *adb shell*

ဒီ Command ကတော့ Command Prompt ထဲကနေပြီး Android ရဲ့ Terminal ထဲကိုဝင်ပြီးအသုံးပြုတာပါ။ အခု Command ကို အသုံးပြုလိုက်တာနဲ့ ကျွန်တော်တို့ဟာ Android ရဲ့ File System ထဲကို တန်းရောက်သွားပါတယ်။ Terminal ထဲမှာအသုံးပြုနိုင်တဲ့ Command ပေါင်းများစွာရှုပါတယ်။ သူတို့ကတော့ Adb ရဲ့ မူလ အခြေခံ command တွေမဟုတ်ပါဘူး။ Android System က Support ပေးတားတဲ့ Command တွေပါ /system/bin အောက်နဲ့ /system/usr/sbin အောက်မှာတွေနိုင်ပါတယ် Background Terminal အသုံးပြုပုံကို သီးသန်အခန်း တစ်ခန်းအနေနဲ့ ရှင်းပြပေးပါမယ်။ နမူနာအနေနဲ့ အောက်ကပုံကိုပဲ လေ့လာကြည့်ပါ။



```
C:\Users\Phyo Ngei>adb shell
$ ls
ls
efs
config
cache
persist
sdcard
acct
mnt
```

အပေါ်က Command ကိုကြည့်ရင် *adb shell* ဆိုတာနဲ့ Command Prompt ကAndroid ရဲ့ Terminal ထဲကိုတန်းရောက်သွားပါတယ် \$ ပုံစံလေးနှုပါ။ သူအောက်မှာ ဘာတွေရှိတယ်၊ ဘယ်လို ဖိုင်တွေရှိတယ်ဆိုတာကို List အနေနဲ့ပြအောင်လို့ *ls* ဆိုတဲ့ Command ကို ထပ်သုံးလိုက်တာပါ။ ဘာနဲ့တူသလည်းဆိုတော့ ကျွန်တော်တို့ Android Device မှာ Root Explorer ကိုတင်ပြီးဖွင့်ကြည့်သလိုပါပဲ။ Root Explorer နဲ့ကြည့်သလောက်လည်း မမြင်ရဘူးလို့ စောဒကတက်ကြီးမယ်ထင်ပါတယ်၊ ဟုတ်ပါတယ်။ အဲဒီလောက်တော့ အကုန်မမြင်ရပါဘူး။ ဘာဖြစ်လိုလဲဆိုတော့ Root Explore ကနေကြည့်ရင် Android System ရဲ့ Hidden File + Folder တွေပါမြင်ပြီး Command Prompt ကနေကြည့်ရင် Hidden File + Folder တွေကို မမြင်ရလိုပါ။ Hidden File + Folder ဆိုတာက File Name တွေရဲ့ အရှေ့မှာ Dot ( . ) ပါတဲ့

ဖိုင်တွေပါ။ ကွန်ပူးတာမှာ ဖိုင်တွေဖျောက်ချင်ရင် ကိုယ့်ဖျောက်ချင်တဲ့ ဖိုင်၊ ဖိုလ်ဒါ အပေါ်မှာ Right Click လုပ်ပြီး Properties ကိုနိုင်ပြီးရင် Hidden ရွှေ့မှာအမှုန်ခြစ်ခြစ်ပေးရပါမယ်၊ Android မှာတော့ ဖိုင်နာမည်ရွှေ့မှာ Dot (.) လေး ထည့်ပေးရှုပါ။ နည်းနည်းပိုရည်သွားပြီ ထင်ပါတယ်။

### (၇) *adb shell <command>*

အခု Command ကတော့ အပေါ်က Command နဲ့ ဆင်တူပါတယ် ကဲလွှာတာကတော့ အပေါ်က Command က Terminal ထဲသွားပြီးမှ တြော်သံဃားသံဃားချင်တဲ့ Command တွေကို terminal ထဲကနေ ဝင်အသုံးပြုရမှာပါ။ အခု Command ကတော့ Terminal ထဲကိုမသွားဘဲနဲ့ Terminal ထဲမှာပဲအသုံးပြုနိုင်တဲ့ Command တွေကို Adb ကနေပဲ လှမ်းပြီးအသုံးပြုတာပါ။

```
C:\Users\Phyo Ngei>adb shell su
su: permission denied
```

ဒါကတော့ Adb ကနေ လှမ်းပြီး Superuser Access လှမ်းယူတာပါ။ Permission Denied ပြုတာကတော့ ကိုယ်စမ်းသပ်နေတဲ့ Android Device မှာ RootAccess မရှိသေးလိုပါ။သူ့၏ Command Formula က "*adb shell <command>*" ဆိုတဲ့အတိုင်း *<command>* နေရာမှာ အစားထိုးအသုံးပြုနိုင်တဲ့ Command တွေ အများအပြားရှိပါတယ်။ ဆက်ပြီးလေ့လာကြည့်ပါဉီးလို့ တိုက်တွန်းပါရစေခင်ဗျာ။

### (၈) *adb remount*

ဒီ Command ကိုတော့ Android Device ရဲ့ System Partition (/system) ရဲ့ read/write Permission ကို Adb သုံးပြီး လှမ်းယူလိုက်တာပါ။ အသုံးများပြီး အရမ်းအသုံးဝင်တဲ့ Command တစ်ခုပါ။ "remount" ကိုသုံးဖို့အတွက် မိမိရဲ့ Android Device ဟာ Root Access ရှိဖို့တော့လိုပါတယ်။

```
C:\Users\Phyo Nge1>adb remount
remount succeeded
```

```
C:\Users\Phyo Nge1>
```

Remount လုပ်တာအောင်မြင်ရင် အောင်မြင်ပါတယ်ဆိုတဲ့ remount succeeded ဆိုတဲ့ မက်ဆွဲချုပေးပြပေးပါတယ်။ တကယ်လို မိမိ Android Device မှာ Root Access မရှိလို Remount လုပ်လို မရတဲ့အခါ "remount failed: Operation not permitted" ဆိုတဲ့ မတ်ဆွဲချုပြလာပါလိမ့်မယ်။ ခြင်းချက်အနေနဲ့ China Tablet တွေဖြစ်တဲ့ A13 Chip သုံး Tablet အခါးမှာတော့ Root Access မရှိပေမယ့် File System ကို "adb remount" သုံးပြီး read/write Permission ရယူနိုင်ပါတယ်။

### (၁၀) adb connect

လက်တွေ့အသုံးချု Adb Command တွေကိုတော့ နောက်ဆုံးတစ်ခု "adb connect" ဆိုတာလေးနဲ့ အဆုံးသတ်ပါရစေ။ Adb Connect ကတော့ ကျွန်တော် တို့ရဲ့ Android Device ကို ကွန်ပျံတာမှာ ကြိုးနဲ့မသုံးဘဲ Wireless နဲ့ချိတ်ပြီး သုံးတာပါ။ ကျွန်တော်တို့ရဲ့ Android Device ကို Wireless ကနေပြီး ဆော့ဝဲတင်တာ၊ ဖြေတ်တာ၊ ဖိုင်ကူးတာ၊ ဖျက်တာ အားလုံးသုံးလို ရပါတယ်။ ဒါပေမဲ့ Adb Connect ကိုသုံးဖို့အတွက်ဆို သူချည်းသက်သက်သုံးဖို့မလွယ်ပါဘူး။ မရဘူးလို မပြောပါဘူး။



နံပါတ် (၁) အနေနဲ့ ကျွန်ုပ်တော်တို့ရဲ့ Android Device မှာ Root Access လိုမယ်။ နံပါတ် (၂) ကတော့ "ADB wireless" ဆိုတဲ့ Android Application တစ်ခုလိုပါတယ်။ Play Store မှာအောင်လုတ်ဆဲလိုလည်းရနိုင်သလို စာအုပ်နဲ့အတူတွဲပါတဲ့ CD ခွေထဲမှာလည်း ထည့်ပေးထားပါတယ်။

"ADB Wireless" ကိုသုံးဖို့အတွက် ကျွန်ုပ်တော်တို့ရဲ့ကွန်ပျူတာနဲ့ Android Device ဟာ Network တူနေရပါမယ်။ အသုံးပြုရမယ့် Command ကတော့ "adb connect <host> <port>" ပဲဖြစ်ပါတယ်။ <port> နေရာမှာတော့ ဘာမှတဲ့ပေးစရာ မလိုပါဘူး Adb က Wireless နဲ့ချိတ်တဲ့ Device တွေရဲ့ Default port ကို "5555" အဖြစ်သတ်မှတ်ပေးပါတယ်။ Host ဆိုတာကတော့ ကိုယ်နဲ့ချိတ်မယ့် Android Device ရဲ့ "IP address" ပဲဖြစ်ပါတယ်။ "ADB Wireless" မှာပြထားတဲ့အတိုင်း မျဉ်းသားပေးထားတဲ့ Command ကို Adb ကနေ ရိုက်ပြီးချိတ်ဆက်အသုံးပြုရမှာ ဖြစ်ပါတယ်။

```
C:\>Users\Phyo Ngeli>adb connect 192.168.43.151
connected to 192.168.43.151:5555

C:\>Users\Phyo Ngeli>adb devices
List of devices attached
192.168.43.151:5555    device

C:\>Users\Phyo Ngeli>adb install "D:\Mobile Service\Android Softwares\motor.apk"
181 KB/s (15071396 bytes in 81.185s)
      pkg: /data/local/tmp/RE 3D Motor.apk
Success

C:\>Users\Phyo Ngeli>
```

အရင်ဆုံး Wireless နဲ့ ချိတ်လိုက်ပါတယ်။ ပြီးတော့ တကယ်ချိတ်ဆက်မိ မမိ စစ်ပါတယ်။ ချိတ်ဆက်မိတယ်ဆိုတာနဲ့ "adb install" ကိုသုံးပြီး ဆော့ဝဲတစ်ခု လှမ်းတင်ကြည့်ပါတယ်။ Wireless ကနေ တင်တာကြောင့် ပုံမှန်ထက် အနည်းငယ် ပိုကြာနေတတ်ပါတယ်။ စောင့်ကြည့်နေလိုက်ပါ။ ခကာအကြောမှာတော့ ဆော့ဝဲတင်လို့ အောင်မြင်တာကြောင့် Success ဆုံးတဲ့ မက်ဆော်ချုပ် ပြလာပေးပါလိမ့်မယ်။

လက်တွေ့အသုံးချုပ် အခန်းကတော့ ဒီလောက်နဲ့ပဲ ရပ်လိုက်ပါတော့မယ်။ အခြား Command တွေလည်း အများအပြားရှိတာမို့ ကျွန်ုပ်တော်ရေးပြသလောက်လေး တွေနဲ့ တတ်ပြီဆုံးပြီး ရပ်မနေဖို့ ဒီထက်လည်း ပိုမိုတတ်ကျွမ်းဖို့ စိတ်အားထက်သန်ရင် အခြား Command တွေကိုလည်း ထပ်ပြီးလေ့လာပါလို့ တိုက်တွန်းပါရစေ။ ဒီထက် ပိုအသုံးဝင်မယ့် နောက်ကအန်းတွေကိုလည်း ဆက်ပြီးလေ့လာကြည့်ပါဉီးခင်ဗျာ။

## အခန်း (၄) အဆင့်မြင့် ADB Command များ

ဒီအခန်းကတော့ အရှေ့ကအခန်းမှာ အကျဉ်းချုပ်ပြဲခဲ့တဲ့ "adb shell <command>" ဆိုတာလေးကို အကျယ်တဝ်ဂျုင်းပြုပေးမှာပါ။ အသုံးဝင်ပုံ အသုံးပြုပုံ တွေမရှင်းပြခင် ကျွန်တော်ဘာလို့ ဒီစာအုပ်ရေးဖြစ်လဲဆိုတာ ပြောပြချင်ပါသေးတယ်။ အမိက ကတော့ အခုလို အသုံးနည်းတဲ့ command တွေကို လေ့လာရင်း သုံးကြည့်ရင်း နဲ့ သူများတွေကိုလည်း ပြန်ပြောပြချင်လာပါတယ်။ ကျွန်တော် စာအုပ်တစ်အုပ်မှ မရေးဖူးပါဘူး။ ဒီစာအုပ်ကိုလည်း အခုရေးမယ့် အခန်းကြောင့် ရေးဖြစ်သွားတာပါ။ အဆင့်မြင့် ADB Command လိုပြောရတာကတော့ အရှေ့ကအခန်းဖြစ်တဲ့ အခြေခံ ADB Command မှာပါပြီးသား Command တွေနဲ့ မတူတာကြောင့်ရယ်၊ ပြီးတော့ အသုံးနည်း Command တွေဖြစ်တာကြောင့်ရယ် နာမည်ပေးလိုက်တာပါ။

ဒီအခန်းရဲ့ ပထမဆုံးတစ်ခုအနေနဲ့ Android ရဲ့ Package Manager (pm) အကြောင်းကို ပြောပြချင်ပါတယ်။ Package Manager လိုပြောလိုက်တာနဲ့ ဆော့ပဲတွေ ကို ဖြုတ်မယ် Disable ပေးမယ် Enable ပေးမယ်၊ Android Application တွေကို ကိုယ့် Device ရဲ့ Internal Memory မှာတင်မလား၊ External Memory မှာတင်မှုလား ဆိုတာတွေကအစ အသေးစိတ် စီမံခန့်ခွဲနိုင်ပါတယ်။ သတိရလိုပြောလိုက်ပါဉိုးမယ်၊ စာဖတ်နေရင်းနဲ့ ခေါင်းထပ်လာမယ်ထင်ပါတယ်။ ကျွန်တော်ဟို "adb install" ဆိုတဲ့ Command သုံးတုန်းက ကိုယ်က ဆော့ပဲကို Sdcard ပေါ်မှာ တင်မယ်ဆို [ -s ] Switch ခံပြီးတင်ခဲ့ပါတယ်။ ဟုတ်ပါတယ်၊ အဲဒါက ဆော့ပဲတစ်ခုချင်းစီအတွက် ပဲ သက်ရောက်တာပါ။ Package Manager ကနေ သတ်မှတ်ရင်တော့ နောက် အသစ်ထပ်တင်မယ့် ဆော့ပဲတွေအတွက်ပါ သက်ရောက်မှုရှိပါတယ်။

### (၁) adb shell pm

"adb shell <command>" ဆိုတဲ့ ခေါင်းစဉ်အောက်မှာကြည့်ရင် <command> ဆိုတဲ့နေရာမှာအသုံးပြုလိုပဲတဲ့ Command တွေအများကြီးကိုမှ ကျွန်တော်က pm အကြောင်းကို အရင်ဆုံးပြောပြတယ်ဆိုတာက ဒါလေးကို သေသေချာချာ ဖတ်စေချင် လိုပါ။ ကျွန်တော်ထို့ Command Prompt မှာ "adb shell pm" လိုခြုံပြီး Enter

ခေါက်လိုက်ရင် သူမှာသုံးလို့ရမယ့် တဗြား Function တွေ အသုံးပြုနိုင်တဲ့ Command တွေကို အသေးစိတ် လာပြောပါလိမ့်မယ်။ အသုံးပြုနိုင်တာတွေ အရမ်းကိုများပါတယ်။ ကျွန်တော် အားလုံးရှင်းပြပါမယ်ဆိုတာထက် အသုံးတကယ်ဝင်ပြီး အသုံးတည့်မယ့် Command တွေကိုဦးစားပေးရှင်းပြပါမယ်။ ကျွန်တော်ပြောပြတာတွေကို အခြေခံပြီး ဆက်လက်လေ့လာကြည့်ဖို့ တိုက်တွန်းပါတယ်။

```
C:\Users\Phyo_Ngai>adb shell pm
* daemon not running. starting it now on port 5637 *
* daemon started successfully *
usage: pm [list packages [-f] [-d] [-e] [-s] [-3] [-i] [-u] [-user USER_ID] [FILTER]
          pm list permission-groups
          pm list permissions [-g] [-f] [-d] [-u] [GROUP]
          pm list instrumentation [-f] [TARGET-PACKAGE]
          pm list features
          pm list libraries
          pm list users
          pm path PACKAGE
          pm install [-l] [-r] [-t] [-i INSTALLER_PACKAGE_NAME] [-s1|-f]
                     [-algo <algorithm name>] --key <key-in-hex> --iv <iv-in-hex>
                     [--originating-uri <URI>] [--referrer <URI>] PATH
          pm uninstall [-k] [-n] USER_ID PACKAGE
          pm clear [-n] USER_ID PACKAGE
          pm enable [-n] USER_ID PACKAGE_OR_COMPONENT
          pm disable [-n] USER_ID PACKAGE_OR_COMPONENT
          pm disable-user [-n] USER_ID PACKAGE_OR_COMPONENT
          pm disable-until-used [-n] USER_ID PACKAGE_OR_COMPONENT
          pm grant PACKAGE PERMISSION
          pm revoke PACKAGE PERMISSION
          pm set-install-location [0/autofill] [1/internal] [2/external]
          pm get-install-location
          pm set-permission-enabled PERMISSION [true/false]
          pm trimcaches DESIRED_FREE_SPACE
          pm set-user USER_NAME
          pm remove-user USER_ID
          pm get-max-users
```

အပေါက ပုံကိုကြည့်ရင် ကျွန်တော်က "adb shell pm" လိုရိုက်ပြီး pm မှာ သုံးလို့ရမယ့် Command တွေကို ကြည့်လိုက်တာပါ။ ပုံမှာ ဘောင်နဲ့ခတ်ပြထားပေးပါတယ် pm မှာ အသုံးပြုနိုင်မယ့် Command တွေအားလုံး စုစုပေါင်း (၂၀)လောက် ရှုပါတယ်။ အသုံးများ၊ အသုံးဝင်မယ့် Command တွေပဲပြောပြပါမယ်။

## (၁.၁) pm list package

pm ရဲ့အရေးမှာပါတဲ့ Adb Shell ဆိုတာကိုတော့ခေါင်းစဉ်မှာ ထည့်မေးပြတော့ပါဘူး။ ဆက်ပြီးတော့ List Package ရဲ့ နောက်မှာ [-f] [-d] [-e] [-s] [-3] [-i] [-u] ဆိုပြီးတဲ့ဖက်အသုံးပြုနိုင်တဲ့ Switch တွေအများကြီးရှုပါတယ်။ Switch တစ်ခုချင်းစီရဲ့လုပ်ဆောင်နိုင်ပုံတွေကို အောက်ကပုံမှာ ကြည့်လိုက်ပါ။

```
pm list packages: prints all packages, optionally only
those whose package name contains the text in FILTER. Options:
-f: see their associated file.
-d: filter to only show disabled packages.
-e: filter to only show enabled packages.
-s: filter to only show system packages.
-3: filter to only show third party packages.
-i: see the installer for the packages.
-u: also include uninstalled packages.
```

List Package ရဲနောက်မှာ [-f] ကိုခံပြီးသုံးမယ်ဆိုရင် တော့ ကိုယ့် Android Device မှာတင်ထားတဲ့ Application နာမည်တွေနဲ့ သူတို့နဲ့ဆိုင်တဲ့ PACKAGE တွေကို ယူဉ်ပြီးပြပေးသွားမှာပါ။ ဒီ Command နဲ့မကြည့်ဘဲ ဘယ် Application က ဘယ် PACKAGE ဆိုတာ မသိနိုင်ပါဘူး။ မသိဘဲနဲ့တော့ ပြန်ဖြတ်ပစ်လို့ရမှာ မဟုတ်ပါဘူး။ မဖြစ်မနေ အသုံးပြုရမှာပါ။ နမူနာတော့ မပြတော့ပါဘူး စာမျက်နှာတွေ များသွားမှာစုံလို့ ကိုယ့်ဘာသာစမ်းသပ်ကြည့်လိုက်ပါခင်ဗျာ။

[-d] ခံပြီးသုံးမယ်ဆို မိမိ Android Device တွေထဲကမှ Disable ဖြစ်နေတဲ့ PACKAGE တွေ တစ်နည်းအားဖြင့် Application တွေကိုပဲပြပေးမှာပါ။ ဥပမာ ကိုယ့်ဖုန်းက ဒေါင်းလုပ်ဆွဲလို့ မရဘူးဆို Download Service များ Disable ပေးထားမိ သလားဆိုတာ စစ်ကြည့်လို့ရပါတယ်။ [-e] ကတော့ [-d] နဲ့ ပြောင်းပြန် Enable ဖြစ်နေတာတွေကိုပဲ ပြပေးမှာပါ။

[-s] ခံပြီးအသုံးပြုမယ်ဆိုရင်တော့ ကိုယ်ကအသစ်ထပ်တင်ထားတဲ့ Application (Third Party) တွေကိုမပြုဘဲ Android Device စတင်ဝယ်ယူစဉ်ကတည်းက စက်ချကထည့်ပေးလိုက်တဲ့ Application တွေရဲ့ PACKAGE တွေပဲပြပေးမှာပါ။ [-3] ကတော့ ကိုယ့်ဘာသာ ထပ်တင်ထားတဲ့ Third Party Application တွေကိုပဲ သီးသန်အနေနဲ့ပြပေးမှာပါ။ ဒါဆိုရင်းပြီပေါ့။ ကိုယ့် Android Device မှာဘယ် Application တွေတင်ထားတယ် ဘယ်လောက်တောင်ရှိတယ် ဆိုတာ အလွယ်တကူ သိနိုင်တာပေါ့။ အခုရေးပြတဲ့ Command ကတော့ ကျွန်ုတ်ကိုယ်တိုင် အသုံးများ ပါတယ်။

```
C:\Users\Phyo Nge1>adb shell pm list package -3
package:air.mis.wof
package:cn.jingling.motu.photo.wonder
package:com.amazing.tattoos.slideshow
package:com.ansangha.drdriiving
package:com.avast.android.mobilesecurity
package:com.bit.androSMART.kbinapp
package:com.comquas.mahabote
package:com.devmobile.kuaiya
package:com.dragon.android.pandaspace
package:com.facebook.katana
package:com.gamevil.smileplants.global
package:com.gau.go.launcherex
package:com.gau.go.launcherex.theme.template
package:com.google.android.apps.maps
package:com.higtalk
package:com.ijinshan.kbatterydoctor_en
package:com.imperialgamestudio.bumblee
package:com.katecca.screenoff.lockdonate
package:com.lovehaydin
package:com.mobilityware.solitaire
package:com.mxtech.videoplayer.pro
package:com.nd.assistance
package:com.netqin.ps
```

အပေါ်မှာ [-3] အသုံးပြုပုံ နမူနာကိုပြထားပါတယ် အခဲဆက်ပြီး ကျွန်ုန်းနေသေးတဲ့ [-i] [-u] ဆိုတဲ့ switch နှစ်ခုအကြောင်းထပ်ပြောပါမယ်။ ပထမ [-i] ဆိုတာကတော့ မိမိ Android Device မှာတင်ထားတဲ့ လက်ရှိအသုံးပြုနေတဲ့ ဆော့ဝဲများရဲ့ Installer ဖိုင်ကိုရှာပေးတာပါ။ ကိုယ်ကသာ sd card ထဲကနေ application ကိုယူတင်ထားတယ်ဆို PACKAGE ရဲ့နောက်က installer ရဲ့ နောက်မှာ ဖိုင်ရဲ့ လမ်းကြောင်းပြပေးမှာပါ။ Installer ဖိုင်ကို ရှာလိုမတွေ့တော့ဘူးဆိုရင်တော့ "null" ဆိုပြီး ပြပေးပါလိမ့်မယ်။

```
C:\Users\Phyo Ngeli>adb shell pm list package -i
package:air.mis.wof installer=null
package:android_id installer=null
package:androididhwext installer=null
package:buke.besttone.caipiao.plugin installer=null
package:cn.goapk.market installer=null
package:cn.jingling.motu.photowonder installer=null
package:com.UGMobile installer=null
package:com.akazam.android.wlandialer installer=null
package:com.amazing.tattoos.slideshow installer=null
package:com.android.backupconfirm installer=null
package:com.android.bluetooth installer=null
package:com.android.browser installer=null
package:com.android.calculator2 installer=null
package:com.android.calendar installer=null
package:com.android.certinstaller installer=null
package:com.android.contacts installer=null
package:com.android.contacts.ext installer=null
package:com.android.defcontainer installer=null
package:com.android.deskclock installer=null
package:com.android.email installer=null
package:com.android.exchange installer=null
package:com.android.gallery3d installer=null
package:com.android.htmlviewer installer=null
```

"pm list package" ဆိုတဲ့ Command ရဲ့နောက်မှာ အသုံးပြုစရာ နောက်ထပ် Switch တစ်ခုရှိပါသေးတယ် [-u] ပဲဖြစ်ပါတယ်။ သူကိုအသုံးပြုမယ်ဆိုရင်တော့ မိမိရဲ့ Android Device မှာဘယ်ဆော့ဝဲတွေ ဖြုတ်ပစ်ထားတယ်(Uninstall) ဆိုတာ တွေ့ရမှာပါ [-u] ရဲ့ လုပ်ဆောင်ချက်ကိုက ပြန်ဖြတ်ထားတဲ့ (Uninstall) ဆော့ဝဲတွေ ကြည့်တဲ့ command ပါ။

```
C:\Users\Phyo Ngel>adb shell pm list package -u
package:air.mis.wor
package:android
package:android.idhwext
package:buke.besttone.caipiao.plugin
package:ch.smalltech.ledflashlight.pro
package:cn.goapk.market
package:cn.Jingling.motu.photowonder
package:com.UCMobile
package:com.akazam.android.wlandialer
package:com.amazing.tattoos.slideshow
package:com.android.backupconfirm
package:com.android.bluetooth
package:com.android.browser
package:com.android.calculator2
package:com.android.calendar
package:com.android.certinstaller
```

တကယ်လို့ တခြားတစ်ယောက်ရဲ့ Android Device မှာ အရင်နောက ဂိမ်း  
တစ်ခုတွေတယ် ဒါနဲ့ မတွေ့တော့လို့ သူအနေနဲ့ ဒီဂိမ်းကို App Hide သုံးပြီး ဖွက်  
ထားတာလား၊ တကယ်ဖြောက်လို့ကိုတာလားဆိုတာကို ကိုယ်တိုင် စစ်ကြည့်နိုင်ပါတယ်။

### (o. j) pm list features

တစ်ချို့ Command တွေကိုကျန်တော် ချွန်ထားခဲ့ပါတယ်။ အခုသုံးတဲ့ list  
features ဆိုတာကတော့ Android Device တစ်ခုမှာပါဝင်လာတဲ့ Features  
တွေကိုကြည့်တဲ့အခါသုံးပါတယ်၊ ကိုယ့် Android Device မှာ wi-fi ပါသလား၊  
bluetooth ပါသလား၊ ဘယ်လို sensor တွေပါသလဲဆိုတာတွေကို စစ်ကြည့်လို့ရ  
ပါတယ်။

```
C:\Users\Phyo Ngel>adb shell pm list features
feature:reqGLsVersion=0x200000
feature:android.hardware.bluetooth
feature:android.hardware.camera
feature:android.hardware.faketouch
feature:android.hardware.location
feature:android.hardware.location.gps
feature:android.hardware.location.network
feature:android.hardware.microphone
feature:android.hardware.screen.landshape
feature:android.hardware.screen.portrait
feature:android.hardware.sensor.accelerometer
feature:android.hardware.telephony
feature:android.hardware.telephony.cdma
feature:android.hardware.touchscreen
feature:android.hardware.touchscreen.multitouch
feature:android.hardware.touchscreen.multitouch.distinct
feature:android.hardware.usb.accessory
feature:android.hardware.wifi
feature:android.software.live_wallpaper
```

Permission အကြောင်းကို ချိန်ပြီး Features အကြောင်းကို ကျော်ပြီးရေးပေးလိုက်ရတာကတော့ ကျွန်တော် ဒီစာအုပ်မှာ Android System ထဲက ဖိုင် တွေကို Permission ပေးတာနဲ့ပတ်သက်ပြီး မရှင်းပြုရသေးလိုပါ။ နောက်အခန်းမှာ ကိုယ့်ရဲ့ Android Device ကို ဆော့မလိုဘဲ Adb ထဲကနေ တန်းပြီး Root ဖောက်ပြတဲ့အခါ Permission အကြောင်းပါ ရှုင်းပြုပေးသွားပါမယ်။ အသုံးပြုစရာ ပလိုတာကြောင့်လည်း ဒီအခန်းမှာတော့ ချုန်လှပ်ထားခဲ့တာပါ။

### (၁။ ၃) *pm install (or) uninstall <package>*

Pm Command ကို အသုံးချုပြီး Application ကို Install လုပ်ခြင်းကတော့ အသုံးပြုခဲတဲ့ Command ပါ။ ဒါပေမဲ့ *pm uninstall* ကတော့ မဖြစ်မနေ အသုံးပြုရမှာပါ။ ဘာလိုလဲဆိုတော့ အရှေ့အခန်းမှာ application တွေကို ပြန်ဖြေတ်ပစ်ချင်ရင် "adb uninstall <package\_name>" ကိုအသုံးပြုပြီး ပြန်ဖြေတဲ့ပေမဲ့ ကိုယ်က data တွေကိုမဖြေတ်ပစ်ချင်လို့ [-k] ကို ခံပြီး "adb uninstall -k <package>" ဆိုရင်တော့ သုံးလို့မရဘဲ သူက pm ကိုအသုံးပြုပါဆိုတဲ့ မက်ဆွဲချုပ်လာလာပြတတ်ပါတယ်။ အဲအချိန်မှာ ကျွန်တော်တို့ဟာ application တွေကို ပြန်ဖြေတ်ပစ်ဖို့အတွက် အခု command ကိုမဖြစ်မနေအသုံးပြုရမှာပါ။

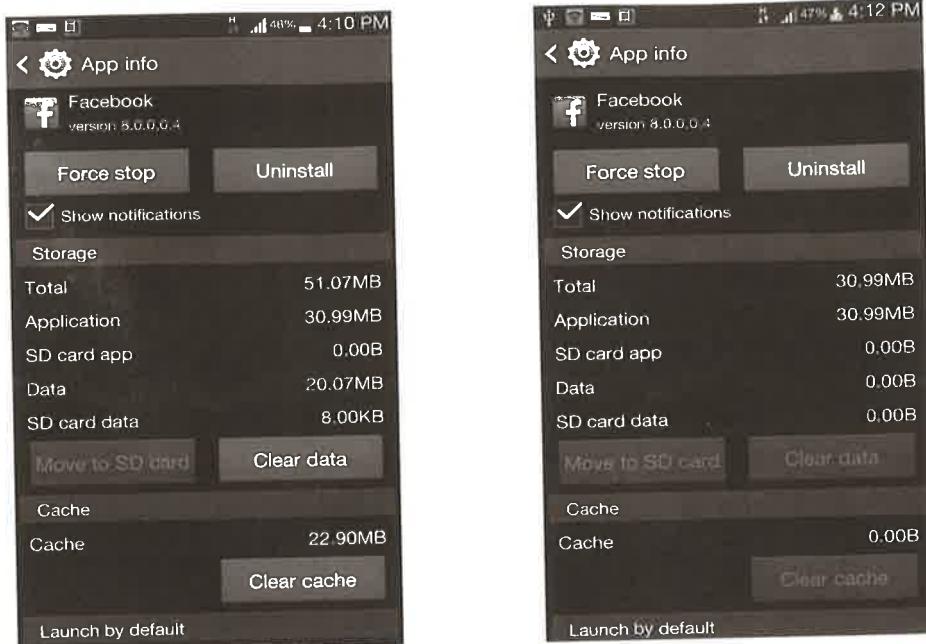
### (၁။ ၄) *pm clear <package>*

ဒီ Command ကတော့ Android Device မှာတင်ထားတဲ့ Application တွေရဲ့ cache၊ data တွေ ကိုဖျက်တဲ့အခါ သုံးပါတယ်။ အရေးမကြိုးဘူးထင်ရပေမယ့် တစ်ခါတစ်လေ ကိုယ်တင်ထားတဲ့ Launcher က Settings မှားပြီး အလုပ်မလုပ်တော့တဲ့အခါ ကိုယ်ကလည်း Android device ရဲ့ Device setting ထဲကို ဝင်လို့ မရဖြစ်နေတဲ့အခါ သုံးပါတယ်။ သူငယ်ချင်းကို စချင်နောက်ချင်တဲ့အခါလည်း သုံးလို့ရပါတယ်။

```
C:\Users\Phyo Ngel>adb shell pm clear com.facebook.katana
Success
```

```
C:\Users\Phyo Ngel>_
```

အပေါ်က Command မှာဆိုရင် Facebook ရဲ့ Data နဲ့ Cache ကို pm clear ဆိုတဲ့ Command ကို အသုံးပြုပြီး ဖျက်ပစ်လိုက်တာပါ။ clear မလုပ်ခင်နဲ့ clear လုပ်ပြီးပဲကို အောက်မှာပြထားပါတယ်။



### (၁၁။) pm enable (or) disable package

ကိုယ့် Android Device မှာတင်ထားတဲ့ PACKAGE တွေကို enable ဖြစ်နေတဲ့ အနေအထားကနေ disable ဖြစ်နေတဲ့ အနေအထား ကိုပြောင်းဖို့အတွက် အသုံးပြုပါတယ်။ နမူနာ မပြတော့ပါဘူး။ ဘယ် application တွေ disable ဖြစ်နေတယ်ဆိုတာကို "adb shell pm list package -d" နဲ့ကြည့်မယ်၊ လိုအပ်ရင် သူတို့ကို "adb shell pm enable <PACKAGE>" ကိုသုံးပြီး enable ပြန်လုပ်ပေးမယ်။ enable ဖြစ်နေနေတာတွေကိုလည်း list package နောက်မှာ [-e] ခံပြီးကြည့်မယ်။ disable လုပ်ချင်ရင် "adb sehll pm disable <PACKAGE>" ဆိုပြီး disable လုပ်လို့ရပါတယ်။

**(၁. ၆) *pm set-install-location***

Android Device မှာ ဆောင်တွေတင်တဲ့အခါ Internal memory မှာတင်မှာလား၊ External memory မှာတင်မှာလား ဒါမှမဟုတ် System က သူဘာသာ အလိုလို ရွေးချယ်တင်ပေးတာကို လိုချင်တာလားဆိုပြီး Android Device ရဲ့ Application တင်မယ့်နေရာကို သတ်မှတ်ပေးနိုင်ပါတယ်။ "pm set-install-location" ရဲ့နောက်ကနေ [0] [1] [2] ဆိုပြီး သတ်မှတ်ပေးရမှာပါ။ [0] ဆို device က အလိုအလောက်ရွေးချယ်ပြီးတင်ပေးမယ့် (auto) ဖြစ်ပြီး၊ [1] ကတော့ Internal [2] ကတော့ External မှာတင်မယ်လို့ သတ်မှတ်တာဖြစ်ပါတယ်။

**(၁. ၇) *pm get-install-location***

လက်ရှိကိုယ့် Android Device ရဲ့ Application တွေ တင်နေတဲ့ နေရာကို စစ်ဆေးကြည့်တာပါ။အပေါ်က Command ထိပါပဲ [0/ Auto]၊ [1/ Internal]၊ [2/ External] ပါ။

**(၁. ၈) *pm create-user <USER\_NAME>***

ကိုယ့်ရဲ့ Android Device မှာ ကိုယ့်စိတ်ကြိုက် User တွေထပ်ဖွဲ့လိုရပါတယ်။ User တွေထပ်ဖွဲ့လိုရပေမယ့် ကျွန်တော်တို့ဟာ Computer မှာသုံးသလို User တွေ အများကြီးဖွင့်ပြီး တစ်ယောက်တစ်ခုဝင်ပြီး အသုံးပြုလိုမာနိုင်ပါဘူး။ ဒါပေမဲ့ နောက်ပိုင်း ဖိုင်တွေကို Permission ပေးတာတို့ File တွေရဲ့ Owner ပြောင်းတာတို့ကို ဖတ်ပြီး စမ်းတဲ့အခါ အသုံးဝင်မှာပါ။

**(၁. ၉) *pm remove-user <USER\_ID>***

User တွေအသစ်ဆောက်ပြီး ပြန်ဖျက်ချင်တဲ့အခါ အသုံးပြုပါတယ်။ တော်ကြာ User တွေဆောက်ပြီး ပြန်မဖျက်တက်ဘူးဖြစ်နေရင် ကိုယ့် Android Device မှာ User တွေများနေပါလိမ့်မယ်။

## (J) *adb shell date*

Adb ကိုအသုံးပြုပြီး Android Device ရဲ့ date and time ကိုလှမ်းပြင်လိုက်ပါတယ်။ *date command* ကိုသုံးမယ်ဆိုရင် တော့ "adb shell date -s YYYYMMDD.HHmmss" ဖော်မြှုပူလာကိုအသုံးပြုရမှုပါ။ အရင်ဆုံး android device ရဲ့ date and time ကို "adb shell date" ဆိုတဲ့ command နဲ့စစ်ကြည့်လိုက်ပါမယ်။ ကြည့်ပြီးရင် "adb shell date -s YYYYMMDD" ဆိုတဲ့ command နဲ့ ပြောင်းကြည့်လိုက်ရင် ကျွန်တော်တိုက Date ပဲရေးပေးလိုက်တာ သူနောက်က Time ကို မရေးဘဲပြောင်းလိုက်တာကြောင့် date ပြောင်းသွားသလို time က 00:00 ကိုပြောင်းသွားပါတယ်။ အဲဒါ ကြောင့် ကိုယ်က Date လေးပဲပြောင်းချင်ရင်တောင် ကျွန်တော်တို့ဟာ *date command* ရဲ့ ဖော်မြှုပူလာအတိုင်း အပြည့်အစုံ ရေးပေးဖို့လိုပါတယ်။

```
C:\Users\Phyo_Nge1>adb devices
List of devices attached
78F5FDFFC4312      device

C:\Users\Phyo_Nge1>adb shell date
Mon Mar 10 13:33:22 MMT 2014

C:\Users\Phyo_Nge1>adb shell date -s 20140312
Wed Mar 12 00:00:00 MMT 2014

C:\Users\Phyo_Nge1>adb shell date -s 20140312.134500
Wed Mar 12 13:45:00 MMT 2014

C:\Users\Phyo_Nge1>
```

## (R) *adb shell ime*

*ime command* ကတော့ Android Device ရဲ့ ကိုဘုတ် တစ်နည်းအားဖြင့် input method နဲ့ပတ်သက်တဲ့ ပြင်ဆင်ချက်တွေ၊ သတ်မှတ်ချက်တွေအတွက် အသုံးပြုပါတယ်။ *ime* (input method engine) မှာ *list*, *set*, *enable*, *disable* ဆိုပြီး Command (ဂု) မျိုးရှုပါတယ်။ အရင်ဆုံး *list command* ကို ကြည့်ကြည့်ပါ။

"adb shell ime list" command မှာလည်း [-a] [-s] ဆိုပြီး *switch (J)* ခု ရှုပါသေးတယ်။ [-a] ကို ခံပြီးရှုက်ရင် ကိုယ့် Android Device မှာတင်ထားသမှု ကိုဘုတ်တွေအားလုံးကို သူတို့၏ ID | PACKAGE | Setting တွေပါ အပြည့်အစုံဖော်ပြပေးပါတယ်။ [-s] ကတော့ Android Device မှာတင်ထားတဲ့ ကိုဘုတ်တွေရဲ့ ID

ကိုပဲ ပြပေးပါတယ်။ ID ဆိုတာတောင် Install လုပ်ထားတဲ့ ကိုးဘုတ်တွေထဲကမှ enable (Settings > Language and input ထဲသွားပြီး Install လုပ်ထားတဲ့ ကိုးဘုတ် ကို အမှန်ခြစ် ခြစ်ပေး) လုပ်ထားတဲ့ keyboard တွေကိုပဲ သီးသန့်ပြပေးတာပါ။

```
C:\Users\Phyo_Nge1>adb shell ime list -a
com.huawei.inputmethod.hwpal/PinyinIME
  id=com.huawei.inputmethod.hwpal/PinyinIME mSettingsActivityName=com.huawei.inputmethod.hwpal.SettingsActivity
  mDefaultResId=0x7f070000
Service:
  priority=0 preferredOrder=0 match=0x100000 specificIndex=-1 isDefault=false
  ServiceInfo:
    name=com.huawei.inputmethod.hwpal.PinyinIME
    packageName=com.huawei.inputmethod.hwpal
    labelRes=0x7f0b0000 nonLocalizedLabel=null icon=0x0
    enabled=true exported=true processName=com.huawei.inputmethod.hwpal
    permission=android.permission.BIND_INPUT_METHOD
C:\Users\Phyo_Nge1>
```

အပေါ်က ပုံမှာ ကြည့်လိုက်ရင် Huawei Device မှာ တခြား ဘာ ကိုးဘုတ်မှ မတင်ရသေးခင်ဖြစ်တာကြောင့် "list -a" နဲ့ကြည့်တဲ့အခါ Huawei ရဲ System keyboard ဖြစ်တဲ့ Huawei keyboard လေးပဲမြင်ရသေးတာပါ။ အဲဒါကြောင့် ကျွန်ုတ်က အရင်ဆုံး android device မှာ frozen keyboard တင်လိုက်ပါမယ်။

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Phyo_Nge1>adb install "D:\Mobile Service\Fonts and Keyboards\FrozenKeyboard_1.1.3.apk"
1812 KB/s (282060 bytes in 0.152s)
      pkg: /data/local/tmp/FrozenKeyboard_1.1.3.apk
Success
C:\Users\Phyo_Nge1>adb shell ime list -a
com.huawei.inputmethod.hwpal/PinyinIME
  id=com.huawei.inputmethod.hwpal/PinyinIME mSettingsActivityName=com.huawei.inputmethod.hwpal.SettingsActivity
  mDefaultResId=0x7f070000
Service:
  priority=0 preferredOrder=0 match=0x100000 specificIndex=-1 isDefault=false
  ServiceInfo:
    name=com.huawei.inputmethod.hwpal.PinyinIME
    packageName=com.huawei.inputmethod.hwpal
    labelRes=0x7f0b0000 nonLocalizedLabel=null icon=0x0
    enabled=true exported=true processName=com.huawei.inputmethod.hwpal
    permission=android.permission.BIND_INPUT_METHOD
com.thihaaayekyau.FrozenKeyboard/com.example.android.softkeyboard.SoftKeyboard
  id=com.thihaaayekyau.FrozenKeyboard/com.example.android.softkeyboard.SoftKeyboard
  mSettingsActivityName=com.example.android.softkeyboard.PKSettings
  mDefaultResId=0x0
Service:
  priority=0 preferredOrder=0 match=0x100000 specificIndex=-1 isDefault=false
  ServiceInfo:
    name=com.example.android.softkeyboard.SoftKeyboard
    packageName=com.thihaaayekyau.FrozenKeyboard
    enabled=true exported=true processName=com.thihaaayekyau.FrozenKeyboard
    permission=android.permission.BIND_INPUT_METHOD
C:\Users\Phyo_Nge1>
```

adb install ကိုသုံးပြီး Frozen keyboard ကို တင်လိုက်ပါတယ်။ ပြီးတော့ "adb.shell ime list -a" ကိုသုံးပြီး ကြည့်လိုက်ရင် input method နှစ်ခုဖြစ်သွားပါပြီ။ အပေါ်မှာ "adb shell ime list -a" အောက်မှာ ဘောင်ခတ်ပြထားတာကတော့ ကျွန်ုတ်ထို Android Device မှာလက်ရှိ Install လုပ်ထားတဲ့ keyboard software

တွေ့ရဲ့ ID တွေပါ။ နောက်ပိုင်း enable၊ disable၊ set command တွေသုံးတဲ့အခါ အခုပြထားတဲ့ ID တွေနဲ့ အလုပ်လုပ်ရမှာပါ။ ဆက်ပြီးတော့ "adb shell ime list -s" နဲ့ကြည့်လိုက်ပါ အေမှ တင်လိုက်တဲ့ frozen keyboard ကို Enable သွားမလုပ်ရ သေးတာကြောင့် original ဖြစ်တဲ့ Huawei Keyboard တစ်ခုပဲ တွေ့ရညီးမှာပါ။

Frozen keyboard ကို "adb shell ime enable ID" ဆိုတဲ့ Command အသုံးပြုပြီး Adb ကနေ Enable လှမ်းလုပ်ပါမယ်။ "now enable" ဆိုတဲ့ မက်ဆွဲချုံလာပြုပါလိမ့်မယ်၊ ဒါဆို Enable လုပ်တာ အောင်မြင်တယ်ပဲ့။ တကယ် Enable ဖြစ် မဖြစ်တဲ့ "adb shell ime list -s" နဲ့ဖြန့်ကြည့်လိုက်ပါ။ Frozen keyboard ရဲ့ ID ကိုလည်းတွေ့ရပါပြီ။

```
C:\Users\Phyo\Nge1>adb shell ime list -s
com.huawei.inputmethod.hwpal.PinyinIME
C:\Users\Phyo\Nge1>adb shell ime enable com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard
com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard: now enabled
C:\Users\Phyo\Nge1>adb shell ime list -s
com.huawei.inputmethod.hwpal.PinyinIME
com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard
C:\Users\Phyo\Nge1>
```

အခြားကိုဘုတ်တွေ့လည်း ဒီအတိုင်းပါပဲ။ လက်ရှိအသုံးများတာက Frozen keyboard ဖြစ်တာကြောင့်သူနဲ့ပဲ လက်တွေ့စမ်းပြတာပါ။ Frozen keyboard ID ကတော့ အနည်းငယ်ရှည်ပါတယ်။ စာလုံးပေါင်းမှားရှုက်လို့ မရပါဘူး။ စာလုံးအကြီးအသေးတောင် မှားလို့မရပါဘူးခင်ဗျာ။ သတိထားစေချင်ပါတယ်။ အခုဆိုရင် Frozen keyboard ကို Enable ပေးပြီးပါပြီ။ ဒါဆို ကျွန်ုတ်တော်တို့က Frozen keyboard ကို input method မှာ Default keyboard အဖြစ်သတ်မှတ်ပေးရပါတီးမယ်။ ဒီတစ်ခါတော့ "adb shell ime set ID" ဆိုတဲ့ Command ကို အသုံးပြုပါမယ်။

```
C:\Users\Phyo\Nge1>adb shell ime list -s
com.huawei.inputmethod.hwpal.PinyinIME
com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard
C:\Users\Phyo\Nge1>adb shell ime set com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard
com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard: unknown id: com.thihauayekau.frozenkeyboard/com.example.android.softkeyboard.SoftKeyboard selected
C:\Users\Phyo\Nge1>
```

Command ကို နှစ်ခါရိုက်ပြထားပါတယ်။ ဘာလို့ရှိက်ပြထားလဲဆိုတော့ ကိုးဘူတ်တွေရဲ့ ID တွေကို ရှိက်တဲ့အခါ စာလုံးအကြီး၊ အသေးတောင်လွှဲလို့မရဘူး ဆိုတာကိုပြချင်လိုပါ။ မူရင်း "SoftKeyboard" ဆိုတဲ့ စာသားမှာ ပထမအကြိမ် "softkeyboard" လို့ရှိက်တဲ့အခါ Error ပြပါတယ်။ နောက်တစ်ကြိမ် SoftKeyboard လို့အမှန်ရှိက်မှု selected ဆိုတာ လာပြပါတယ်။ အခုခိုရင် ကျွန်တော်တို့ရဲ့ Android Device မှာ Frozen Keyboard ကို Default input method အနေနဲ့ ရွေးပြီးသား ဖြစ်သွားပါပြီ။ အခုလိုပဲ အခြားကိုးဘူတ်တစ်ခုခုကိုလည်း ရွေးချယ်နိုင်ပါတယ်။

*adb shell ime* ရဲ့ နောက်ဆုံး command တစ်ခုကတော့ "adb shell ime disable ID" ဖြစ်ပါတယ်။ ကိုယ်က Enable လုပ်ထားပြီးသားကို disable ပြန် လုပ်ချင်တဲ့အခါ (Settings ထဲမှာ အမှန်ခြစ် ခြစ်ထားတာကို အမှန်ခြစ်ပြန်ဖြတ်) အသုံးပြုပါတယ်။ Disable လုပ်လိုက်ပြုဆိုရင်တော့ "adb shell ime list -s" နဲ့ပြန်ကြည့်ရင် frozen keyboard ရဲ့ ID ကိုမတွေ့ရတော့ပါဘူး။

```
C:\Users\Phyo Ngeli>adb shell ime disable com.thihaaagekyaav.frozenkeyboard/com.example.android.softKeyboard.softKeyboard.SoftKeyboard: now disabled
C:\Users\Phyo Ngeli>
```

### (၅) *adb shell netcfg*

ကိုယ်ရဲ့ Android Device ရဲ့ IP ကိုသိချင်တဲ့အခါအသုံးပြုပါတယ်။ ပုံမှန်အတိုင်း GUI ကနေကြည့်ချင်ရင်တော့ Settings > Wireless and networks > WLAN settings ကိုသွားပြီး လက်ရှိ ချိတ်ဆက်ထားတဲ့ နက်ဝါးရဲ့ SSID ကို နှုပ်လိုက်ရင် IP address ဆိုတဲ့နေရာမှာ လက်ရှိ ကိုယ့် Android Device ရဲ့ IP ကို ရှာတွေ့နိုင်ပါတယ်။ ADB ကနေ ကြည့်ချင်တယ်ဆိုရင်တော့ "adb shell netcfg" ဆိုတဲ့ Command ကိုအသုံးပြုရပါမယ်။

```
C:\Users\Phyo Ngeli>adb shell netcfg
lo      UP    127.0.0.1      255.0.0.0      0x000000049
usb0   DOWN  0.0.0.0       0.0.0.0      0x000001002
sit0   DOWN  0.0.0.0       0.0.0.0      0x000000080
wlan0  DOWN  0.0.0.0       0.0.0.0      0x000000080
softap0 UP    192.168.43.1  255.255.255.0  0x00001043
ppap0  UP    10.244.165.68  255.255.255.0  0x000010d1
C:\Users\Phyo Ngeli>
```

မျဉ်းသားပြထားတာကတော့ ကိုယ့် Android Device ရဲ့ IP address ပါ အရှေ့မှာ စိုင်းပြထားတဲ့ softap0 ဆိုတာက လက်ရှိ Android Device ဟာ WLAN hotspot အဖြစ် အသုံးပြုနေတယ်လို့ဖော်ပြတာပါ။ softap ဆိုတာကတော့ "software access point" ကိုပြောတာပါ။ တကယ်လို့ AP(Acess Point) အဖြစ်အသုံးပြုနေတာ မဟုတ်ဘဲ ကိုယ့်နဲ့ နက်ဝါခိုင်တဲ့ (တစ်နေရာတည်းမှာ Wi-fi အတူသုံးခြင်း) သုံးနေတာ ဆိုရင်တော့ softap0 နေရာမှာ eth0 ဆိုပြီးပြပေးမှာပါ။ eth ဆိုတာ Ethernet ပဲဖြစ်ပါတယ်။

			netcfg	
lo	UP	127.0.0.1	255.0.0.0	0x0000000049
dummy0	DOWN	0.0.0.0	0.0.0.0	0x0000000082
rmonet0	DOWN	172.17.0.106	255.255.255.252	0x0000010002
rmonet1	DOWN	0.0.0.0	0.0.0.0	0x0000010002
rmonet2	DOWN	0.0.0.0	0.0.0.0	0x0000010002
rmonet3	DOWN	0.0.0.0	0.0.0.0	0x0000010002
rmonet4	DOWN	0.0.0.0	0.0.0.0	0x0000010002
rmonet5	DOWN	0.0.0.0	0.0.0.0	0x0000010002
rmonet6	DOWN	0.0.0.0	0.0.0.0	0x0000010002
rmonet7	DOWN	0.0.0.0	0.0.0.0	0x0000010002
usb0	DOWN	0.0.0.0	0.0.0.0	0x0000010002
tun10	DOWN	0.0.0.0	0.0.0.0	0x0000000080
sit0	DOWN	0.0.0.0	0.0.0.0	0x0000000080
infiniband	DOWN	0.0.0.0	0.0.0.0	0x0000000080
eth0	UP	192.168.137.11	255.255.255.0	0x000001043

C:\Users\Phyo\_Nge1>

### (၂) adb shell df

df ဆိုတာကတော့ " disk free " ဖြစ်ပြီး ကိုယ့် Device မှာ memory ဘယ်လောက်လွှတ်သေးလဲ ဘယ်အပိုင်းတွေက များနေပြုလဲဆိုတာ ကြည့်လို့ရပါတယ်။ ဥပမာ ကျွန်ုတ်တို့ Device က လေးနေတယ် (တစ်ခုခုဖွင့်လိုက်တိုင်း ကြာနေတယ်) ဆို cache တွေပြည့်နေလို့လား ဘယ်အပိုင်းမှာ ဒေတာတွေများလို့ လေးနေတာလဲ ဆိုတာ စစ်ဆေးကြည့်နိုင်ပါတယ်။

Filesystem	Size	Used	Free	B1ksize
/dev	193M	64K	193M	4096
/mnt/asec	193M	0K	193M	4096
/mnt/obb	193M	0K	193M	4096
/system	504M	493M	10M	4096
/data	634M	501M	132M	4096
/cache	188M	4M	184M	4096
/eust*	79M	46M	32M	4096
/mnt/sdcard	2G	2G	0K	32768

C:\Users\Phyo\_Nge1>

အပေါ်ကပုံမှာဆို ကိုယ့်ရဲ့ sdcard 2G ဟာ လုံးဝပြည့်နေပါပြီ။ cache အပိုင်းကတော့ system က သတ်မှတ်ထားတဲ့ size နဲ့ဆို တော်တော်ကျွန်ုပါသေးတယ်။ အင်တာနာက်အသုံးပြုတဲ့ ဖုန်းတွေ၊ Device တွေဆို cache က အမြဲလိုလိုပြည့်နေ တတ်ပါတယ်။

### (၆) *adb shell am*

*am* ဆိုတာကတော့ Activity Manager ရဲ့အတိုကောက်ဖြစ်ပါတယ်။ သူကတော့ ကိုယ့်ဖုန်းထဲမှာ ရှိတဲ့ Services တွေ Application တွေကို *adb command* အသုံးပြုပြီး ဖွင့်ပေးနိုင်ပါတယ်၊ ပိတ်ပေးနိုင်ပါတယ်။ ဒါကြောင့် အသုံးဝင်တဲ့ Android ရဲ့ Tool တစ်ခုဖြစ်ပါတယ်။

ကျွန်ုတ်တို့ဟာ *am command* ကိုအသုံးပြုပြီး ဖုန်းခေါ်တဲ့ dailer ကိုဖွင့်ပြီး တစ်ခါတည်း ကိုယ်ခေါ်ချင်တဲ့နံပါတ်ကို ရှိက်ထည့်ပေးလို့ရသလို Browser၊ Buletooth၊ wifi ကအာဆ *am* ကိုအသုံးပြုပြီး ဖွင့်လိုပါတယ်။ အရမ်းအသုံးဝင်တဲ့ android ရဲ့ Tool တစ်ခုဖြစ်လို့ ကျွန်ုတ်တို့လို့ လေ့လာ၊ စမ်းသပ်ချင်တဲ့သူတွေအတွက်တော့ သဘောကျစေမှုအမှန်ပါပဲ။ ပထမဆုံးအနေနဲ့ *am* နဲ့ပတ်သက်ပြီး ဘာတွေအသုံးပြုနိုင်တယ်ဆိုတာကို "adb shell am" လို့ ရှိကြပြီးကြည့်ပါမယ်။

```
C:\Users\Phyo.Ngei>adb shell am
usage: am [subcommand] [options]
      start: an start [-D] [-U] <INTENT>
              -D: enable debugging
              -U: wait for launch to complete
      start a Service: am startservice <INTENT>
      send a broadcast Intent: am broadcast <INTENT>
      start an instrumentation: am instrument <FILE>/<COMPONENT>
          -r: print raw results (otherwise decode REPORT_KEY-STREAMRESULT)
          -e <NAME> <VALUE> set argument <NAME> to <VALUE>
          -p <FILE> write profiling data to <FILE>
          -w: wait for instrumentation to finish before returning
      start profiling: am profile <PROCESS> start <FILE>
      stop profiling: am profile <PROCESS> stop
      start monitoring: am monitor [-gdb <port>]
          --gdb: start gdbserver on the given port at crash/ANR
      <INTENT> specifications include these flags:
          -a <ACTION> [-d <DATA_URI>] [-t < MIME_TYPE>]
          -c <CATEGORY> [-c <CATEGORY>] ...
          [-ei--es <EXTRA KEY> <EXTRA STRING_VALUE> ...]
          [-esn <EXTRA KEY> ...]
          [-es <EXTRA KEY> <EXTRA BOOLEAN_VALUE> ...]
          [-e <EXTRA KEY> <EXTRA INT_VALUE> ...]
          [-n <COMPONENT>] [-f <FLAGS>]
          [-grant-read-uri-permission] [-grant-write-uri-permission]
          [-debug-log-resolution]
          [-activity-honour-to-front] [-activity-clear-top]
          [-activity-clear-when-task-restart] [-activity-exclude-from-recents]
          [-activity-launched-from-history] [-activity-multiple-task]
          [-activity-no-animation] [-activity-no-history]
          [-activity-no-user-action] [-activity-previous-is-top]
          [-activity-reorder-to-front] [-activity-reset-task-if-needed]
          [-activity-single-top]
          [-receiver-registered-only] [-receiver-replace-pending]
          [<URI>]
```

*am command* ကိုသုံးပြီး မိမိ ဖုန်းထဲမှာရှိတဲ့ Application တွေကို  
တန်းပြီးခေါ်သုံးလို့ရပါတယ်။ ခေါ်သုံးပုံ ခေါ်သုံးနည်းကတော့

```
adb shell "am start -n package/activity_name_or_full_activity_name_
with_package"
```

ဆိုတဲ့ ပုံစံအတိုင်းပဲဖြစ်ပါတယ်။ အရင်ဆုံး Android ဖုန်းရဲ့ Settings ကို  
adb ကနေလှမ်းပြီး ဖွင့်ကြည့်ပါမယ်။

```
am start -n com.android.settings/com.android.settings.Settings
```

အလားတူပဲ Settings ကိုဖွင့်ချင်တယ်ဆိုရင် တစ်ခြားနည်းလမ်းအနေနဲ့  
အမြန်

```
am start -n com.android.settings/.Settings
```

ဆိုပြီးလည်း ဖွင့်နိုင်ပါတယ်။

နောက်ပြီး *adb command* ကနေပြီး Dailer ကိုတန်းပြီး ခေါ်သုံးမယ်ဆိုရင်လည်း  
ခေါ်သုံးလို့ ရပါတယ်။ တစ်ခါတည်း ဖုန်းနံပါတ်ကို တန်းခေါ်လို့ရသလို၊ တန်းမခေါ်ဘဲ  
dailer ကိုဖွင့် ကိုယ်ခေါ်မယ့်နံပါတ်ကိုရှိက်ထည့်ထားတဲ့ အဆင့်ပဲရောက်အောင်လည်း  
လုပ်လို့ရပါတယ်။ အခုတစ်ခါတည်း **adb** ကနေ ဖုန်းလှမ်းခေါ်ကြည့်ပါမယ်။

```
adb shell am start -a android.intent.action.CALL -d tel:094500 xxx
```

အခုရေးလိုက်တဲ့ Command မှာဆိုရင်လည်း အပိုင်း (j) ပိုင်းရှုပါတယ်။  
ပထမ အပိုင်းက ဘာလုပ်မယ်ဆိုတဲ့ action ဖြစ်ပြီး ကိုယ်က ဖုန်းခေါ်မှာမြှို့လို့ သူရဲ့  
action ဖြစ်တဲ့ *CALL* action ရဲ့ *Action* အပြည့်အစုံဖြစ်တဲ့ *android.intent.action.*  
*CALL* " ကိုရေးပါတယ်။ ပြီးတော့ ကိုယ်ခေါ်ချင်တဲ့ နံပါတ်၊ တစ်နည်းအားဖြင့် *CALL*

ဆိုတဲ့ action အတွက် လိုအပ်တဲ့ data အပိုင်းဖြစ်တဲ့ "tel:phone number" ကိုရေးပေးရတာပါ။ ရှုပ်သွားသလား မသိဘူးခင်ဗျာ။ ကျွန်တော်တိုက တစ်ယောက် ယောက်ကို သွားလိုက်ပါကွာ လိုပြောရင် နောက်ကနေ ဘယ်ကိုသွားရမယ်ဆိုတာ လေးကိုပါ ပြောပြလိုက်တဲ့ သဘောပါ။ တကယ်လို ကျွန်တော်တိုက ဖုန်းခေါ်တဲ့ action ကို မလုပ်ဘူး၊ dailer ကိုဖွံ့ဖြိုး ဖုန်းနံပါတ် ရှိက်ထည့်တဲ့ အဆင့်ထိပ်လုပ်မယ် ဆိုရင်တော့

*adb shell am start tel:phone number*

ဆိုပြီးရိုက်ရုံပါပဲ။ PHONE NUMBER နေရာမှာ ရေးပေးရမှာက ကိုယ်ခေါ်ချင်တဲ့ ဖုန်းနံပါတ်ပါ။ ဒါကတော့ action မပါပါဘူး data ချည်းသက်သက်ရှိက်လိုက်တာပါ။ ဒါတင်ပဲ **am** ကအသုံးကျတာလားဆိုတော့ ကျွန်ပါသေးတယ်၊ contact မှတ်တာကို လည်း **am command** အသုံးပြုပြီး မှတ်လိုပါတယ်။ ကြောတာပေါ့ Contact တစ်ခုကို Command ထဲကနေမှတ်ရင် ဖုန်းကိုဖွံ့ဖြိုး ရှိုးရှိုးတန်းတန်း contact မှတ်လိုက်ရင် ပိုမြန်တယ်လို ပြောကြပါပြီးမယ်။ ဟုတ်ပါတယ် ဒါပေမဲ့ စမ်းသပ်စေချင်လိုရယ် **am command** အလုပ်လုပ်နိုင်ပုံကို လေ့လာစေချင်လိုရယ် ရေးပြတာပါခင်ဗျာ။

*adb shell am start -a android.intent.action.INSERT -t vnd.android.cursor.dir/contact -e name 'red hat' -e phone 09450032521*

Command က အနည်းငယ်ရှည်ပါတယ်။ ဒါပေမဲ့ အပြည့်အစုံရှိက်ပေးရမှာပါ။ အခု လို ရှိက်လိုက်ရင် Android က အလိုလို နားလည်ပြီး Contact မှာ ကိုယ်ပေးလိုက်တဲ့ နာမည် ဖုန်းနံပါတ်တွေနဲ့ contact တစ်ခု ဖန်တီးပေးပါလိမ့်မယ်။ ဒါပေမဲ့ နောက်ဆုံးအဆင့်ဖြစ်တဲ့ save ဆိုတာကိုတော့ ကိုယ်တိုင်ပဲ ဖုန်းမှာရွေးပေးရမှာပါ။ အပေါ်က Command မှာဆို INSERT ဆိုတာက action ဖြစ်ပါတယ်။ သူ့နောက်က [-t] ကတော့ "type of inserting data" ဖြစ်ပြီး ကိုယ်ထည့်မယ့် data type ကိုရွေးချယ်ပေးတာပါ။ [-e] ကတော့ EXTRAKEY ဖြစ်ပြီး [-e] ရဲ့ နောက်ကနေ name ကိုထည့်ပြီး ကိုယ်မှတ်ချင်တဲ့ contact ရဲ့နာမည်ကို ရေးပေးရပါတယ်။ နောက်ထပ် [-e] ကိုဆက်သုံးထားတာကလည်း ကိုယ်ထည့်မယ့် data က

တစ်နေရာတည်းမဟုတ်ဘဲ နေရာတစ်ခုထက်ပို့နေလို ပို့သလောက်ကို [-e] ထပ်ခံပြီး ရေးပေးရတာပါ။ [-e] နောက်က phone ဆိုတာက ကိုယ်မှတ်မယ့် contact ရဲ့ phone number ကိုရေးဖို့အတွက် အသုံးပြုထားတာပါ။ တကယ်လို သူ့ရဲ့ email ပါ ထပ်ထည့်ချင်တယ်ဆိုရင် နောက်ထပ် [-e] email <email account> ဆိုပြီး [-e] ကိုခံပြီး ဆက်ရေးပေးသွားလို ရပါတယ်။

*am* ကိုအသုံးပြုပြီး Browser ကိုဖွင့်မယ် အဲဒီကနေတစ်ဆင့် ကိုယ်ကြည့်ချင်တဲ့ web page တွေကို တစ်ခါတည်းဖွင့်ချင်တယ်ဆိုရင် ရပါတယ်။ အကယ်၍ ကျွန်တော်တိုက Browser ကိုဖွင့်ချင်တယ်ဆိုရင်တော့

```
adb shell am start -n com.android.browser/.BrowserActivity
```

ဒါနဲ့ အပေါ်မှာရှုပြခဲ့တဲ့ data တွေ action တွေအကြောင်းကို ကျွန်တော် ထပ်ရှင်းပြဖို့ လို့ဦးမလားမသိဘူးခင်ပျော်။ ဘာပဲဖြစ်ဖြစ် ကျွန်တော် action နဲ့ပတ်သက်ပြီး Brower ကနေ Youtube က ရပ်ရှင်တွေကို လုမ်းကြည့်တာကို ထပ်ပြေပြပေးပါမယ်။ အဓိကကတော့ action အကြောင်းကို ပို့သိစေချင်လိုပါ။ ဟူတ်ကဲ့ပါ။ ဒီတစ်ခါ Browser ကိုဖွင့်ပြီး Youtube ကနေ Video ကို တိုက်ရှုက်ဖွင့်ကြည့်မယ် ဆိုရင်တော့

```
adb shell am start -a android.intent.action.VIEW -d http://www.youtube.com/watch?v=8ADwPLSFeY8
```

[-a] နောက်က action က VIEW လုပ်မယ်လိုပြောပြီး [-d] နောက်က data ကတော့ ဘာကိုကြည့်ချင်တယ်လို ပြောပါတယ်။ VIEW ဆိုတဲ့ action ကို သုံးပြီးပြီဆိုရင်တော့ နောက်ကနေ Youtube ကို ကြည့်မှုရယ်မဟုတ်ပါဘူး။ ကိုယ်ကြည့်ချင်တဲ့ website လိပ်စာတွေကိုလည်း [-d] နောက်မှာ တစ်ခါတည်း ရှိက်ထည့်ကြည့်နိုင်ပါတယ်။ VIEW action ကိုအသုံးပြုပြီး Google map ကိုလည်းကြည့်နိုင်ပါသေးတယ်။

```
adb shell am start -a android.intent.action.VIEW -d geo:37.8,-122.23?z23
```

အပေါ် Command တစ်ကြောင်းလုံးမှာမှ [-d] နောက်က data ကိုကြည့်လိုက်ပါ geo:37.8,-122.23 ဆိုတာ ကိုယ်ကြည့်ချင်တဲ့ မြေပုံတည်နေရာကိုပြောပြီး၊ နောက်ဆုံးမှာပါတဲ့ ?z=23 မှာဆိုရင် z ဆိုတာက zoom ဖြစ်ပြီး 23 ဆိုတာကတော့ zoom size ပါ။ zoom size ကို 1 to 23 ခဲ့ကြည့်လို့ရပါတယ်။ အခုံရေးပြထားတဲ့ 23 ဆိုတာကတော့ အကျယ်ဆုံးပဲဖြစ်ပါတယ်။

သတိထားစရာရှိပါတယ်။ တကယ်လို့ ကျွန်တော်တို့က am command ကိုအသုံးပြုပြီး Bluetooth ကိုဖွင့်မယ်ဆိုရင်တော့ တကယ်ဖွင့်မယ်၊ မဖွင့်ဘူးဆိုတာ user confirm လာတောင်းပါလိမ့်မယ်။ အဲဒီအခါကျ ကိုယ့်ရဲ့ Device မှာ **ok** ကိုနှပ်ပေးမှာသာ Bluetooth က ပွင့်လာမှာ ဖြစ်ပါတယ်။

```
adb shell am start -a android.bluetooth.adapter.action.REQUEST_ENABLE
```

ဖွင့်တာချည်းပဲရေးပြနေလို့ မပိတ်တတ်ဘဲလည်း ဖြစ်နေမှာစီးရပါတယ် ခင်ဗျာ။ တကယ်လို့ ကျွန်တော်တို့ဟာ လက်ရှိအလုပ်လုပ်နေတဲ့ activities တွေကို ပိတ်ချင်ရင်လည်းပိတ်ပစ်လို့ရပါတယ်။ အားလုံးကို တစ်ပြီးစုနက်တည်း ပိတ်နိုင်သလို တစ်ခု ချင်းစီလည်း ပိတ်နိုင်ပါတယ်။

```
adb shell am kill all
```

တကယ်လို့ အားလုံးကိုတစ်ပြီးစုနက်တည်းမပိတ်ချင်ဘူး။ တစ်ခုချင်းစီပဲ ပိတ်ချင်တယ်ဆိုရင်လည်း သူတို့ရဲ့ PACKAGE အလိုက်ပိတ်ပေးရမှာပါ။

```
adb shell am kill <package>
adb shell am force-stop <package>
```

PACKAGE နေရာမှာ ကိုယ်ပိတ်ချင်တဲ့ PACKAGE ကို ရေးပေးပါ။ ဥပမာ- viber ကိုပိတ်ပစ်ချင်တယ် ဆိုရင်တော့ com.viber.voip လို့ရေးပေးရမှာပါ။ command ကို တစ်ကြောင်းပဲရေးပေးရမှာပါ။ kill နဲ့လည်း ပိတ်ပစ်နိုင်သလို force-stop နဲ့လည်း ပိတ်နိုင်တယ်ဆိုတာကို ရေးပြထားတာပါ။

## (၃) *adb shell monkey*

ဆက်ပြီးတော့ ရေးပြုမယ့် Command ကတော့ monkey ဆိုတဲ့ android ရဲ့ အသုံးဝင်တဲ့ Tool တစ်ခု အကြောင်းပါပဲ။ သူကလည်း am၊ pm၊ busybox တို့လို command line tool တစ်ခုဖြစ်လို သူကိုအသုံးပြုမယ်ဆို အခုလို command prompt ထဲကနေပဲအသုံးပြုနိုင်မှာပါ။ monkey tool ကိုတော့ ကိုယ်ဆီမှာ Install လုပ်ထားတဲ့ Application တွေကို စစ်ဆေးဖို့အတွက်အသုံးပြုနိုင်ပါတယ်။ အသုံးပြုပုံကို အကျယ် မရှင်းပြတော့ပါဘူး။ ကိုယ်တိုင် ဆက်ပြီးလေ့လာကြည့်ပါ။ android developer တွေရဲ့ အဆိုကတော့ (သင်သာ monkey tool ကိုအသုံးပြုနိုင်ပြီးဆို သင့်ဘဝတစ်ဆစ်ချိုး ပြောင်းလဲသွားပြီ) လို ဆိုပါတယ်။ monkey tool ကိုအသုံးပြုပြီး Application တွေကို စစ်ကြည့်မယ်ဆိုရင်တော့

```
adb shell monkey -p <ALLOWED PACKAGE> [event_number]
```

အပေါ်က command လိုင်းကတော့ အသုံးပြုရမယ့် ပုံစံကိုပြပေးထားတာပါ။ တကယ်လို ကျွန်တော်တို့က facebook application ကို moneky tool နဲ့ စစ်ကြည့်မယ်ဆိုရင်တော့

```
adb shell monkey -p com.facebook.katana 500
```

ဆိုပြီး စစ်ဆေးကြည့်နိုင်ပါတယ်။

## အခန်း (၅) Android Terminal အသုံးပြုခြင်း

Android Terminal အသုံးပြုခြင်းဆိုတာကတော့ "adb shell" command ကိုအသုံးပြုပြီး android file system ထဲကိုဝင်မယ် သူရဲ့ Terminal ထဲမှာ တစ်နေရာကနေ တစ်နေရာ ဖိုင်ကူးမယ် ဖျက်မယ် ဖိုင်တွေ အသစ်ဆောက်မယ် ပြင်မယ် ဖိုလ်ဒါတွေ အသစ်ဆောက်မယ် အစရှိသွေ့ပြင့် များစွာသော အလုပ်တွေကို လုပ်နိုင်ပါတယ်။ "adb shell <command>" နဲ့သုံးတာဟာ Android ရဲ့ Terminal ထဲကိုမဝင်တော့ဘဲ ADB ကိုသုံးပြီးပဲ android device တွေကိုထိန်းချုပ်တာပါ။ အခုလို Terminal ထဲကနေအသုံးပြုတာလောက်တော့ အသေးစိတ်မကျပါဘူး။ Android file system ဆိုတာကလည်း open source ဖြစ်ပြီး လက်ရှိကွန်ပျိုးတွေမှာ ခေတ်စားလာတဲ့ Ubuntu OS နဲ့အသုံးပြုရတာချင်းဆင်တူပါတယ်။ Ubuntu (Linux) သုံးဖူးရင်လည်း ဒါကလွယ်ပါတယ်။ ထိုအတူပဲ ဒါကို ကျမ်းကျမ်းကျင်ကျင် နားလည်ပြုဆိုရင်လည်း တစ်ဖက်က Computer System ကိုလည်း အတော်အသင့် နားလည်သွားပါလိမ့်မယ်။

Android မှာဆိုရင် အသုံးပြုလိုရနိုင်မယ့် Command တွေကို /system အောက်က bin\sbin\xbin ဖိုလ်ဒါတွေအောက်မှာတွေ့နိုင်ပါတယ်။ Terminal ထဲမှာ "echo \$PATH command" ဆိုတဲ့ command အသုံးပြုပြီး path တွေကိုကြည့်လို ရပါတယ်။ အဲဖိုလ်ဒါတွေအောက်က Command တွေအားလုံး အသုံးပြုနိုင်ပါတယ်။ ဒီအခန်းမှာဆိုရင်လည်း ကျွန်တော်က Android Device အတွက် အသုံးဝင်၊ အသုံးတည့်မယ့် Comand တွေ Tools တွေကိုပဲ ရှုင်းပြပေးသွားမှာပါ။ Android Device တွေကို ဆားပစ်လုပ်ပေးနိုင်လောက်အောင်၊ လုပ်ပေးနိုင်လောက်တဲ့ အသုံးဝင် Command တွေကို ဦးစားပေး ရေးပြပေးသွားပါမယ်။ အရင်ဆုံး command prompt ကိုဖွင့် adb devices command ကိုအသုံးပြုပြီး ကိုယ့် android device ကို ကွန်ပျိုးတာက သို့ မသိစစ်ကြည့်လိုက်ပါ။ သိပြုဆိုရင် android ရဲ့ file system ထဲကို Terminal ကနေ ဝင်ရောက်အသုံးပြုဖို့အတွက် "adb shell" လို့ရှုက်ပြီး Terminal ထဲ စပြီးဝင်ပါမယ်။ Terminal ထဲမှာ (\$) နဲ့ဆုံးရှိုးရှိုး user access ဖြစ်ပြီး (#) နဲ့ဆိုရင်တော့ root access ဖြစ်ပါတယ်။ ကိုယ့်ရဲ့ android device က root access ရှိပြီးသား သို့ပေမဲ့ (\$) ပဲ ပြတယ်ဆိုရင်တော့ \$u လို ထပ်ရှိက်ပြီး root access (#) ယူလို့ရပါတယ်။ အခုဆိုရင်

android device ခဲ့ Terminal ထဲကိုရောက်နေပြီဖြစ်လို့ အသုံးပြုပဲနဲ့ အသုံးပြုနိုင်မယ့် command တွေကို ဆက်ပြီးလေ့လာကြည့်ပါ၍ီးခင်ဗျာ။

### (၁) ls

ပထမဦးဆုံးအနေနဲ့ *ls command* ကိုအသုံးပြုပါမယ်။ *ls* ခဲ့ အရှည်က *list*ပါ။ သူကတော့ android device ခဲ့ ဘယ် partition အောက်မှာဘာတွေရှိတယ်၊ ဘယ် ဖို့လ်ဒါအောက်မှာ ဘာတွေရှိတယ်ဆိုတာကို အသေးစိတ်ကြည့်လို့ရပါတယ်။ Adb ကနေလည်း ဒီ Command တွေကို လှမ်းသုံးလို့ရပါတယ်။ ဥပမာ sd card အောက်မှာ ဘယ်ဖိုင်တွေရှိသလဲဆိုတာကို "adb shell ls /sdcard" ဆိုတဲ့ Command ကိုအသုံးပြုပြီး ကြည့်နိုင်ပါတယ်။ *ls* နောက်မှာ [-1] ခံပြီးအသုံးပြုမယ်ဆိုရင်တော့ ဖိုင် ဖို့လ်ဒါတွေရဲ့ permission၊ user name၊ size၊ modified date တွေပါပြပေးပါတယ်။

### (၂) cd

*cd* ဆိုတာကတော့ change directory ပါ။ Android File System ကို Terminal ကနေ ဝင်ရောက်အသုံးပြုတဲ့အခါ Directory တစ်ခုကနေ အခြားတစ်ခု၊ Partition တစ်ခုကနေ အခြားတစ်ခုဆိုကိုပြောင်းတဲ့အခါ အသုံးပြုပါတယ်။ Android မှာ root directory ကို (/) နဲ့ဖော်ပြပါတယ်။ လက်ရှိရောက်နေတဲ့ (/) အောက်က နေပြီး တော့ system ထဲကိုသွားချင်ရင် *cd / system* ဆိုပြီး အသုံးပြုနိုင်ပါတယ်။ ကျွန်တော်တို့၏နေကျ ဖို့လ်ဒါဆိုတာက တစ်နည်းအားပြင့် Directory လို့၏ပေါ်ပါတယ်။ ကိုယ်လက်ရှိရောက်နေတဲ့ ဖိုင်လမ်းကြောင်းကနေ သူရဲ့ upper level ကို ပြန်သွား မယ်ဆိုရင်တော့ *cd ..* နဲ့ ပြန်ထွက်ပါမယ်။ တကယ်လို့ ကျွန်တော်တို့ရောက်နေတာ (/) အောက်မဟုတ်ဘဲ (/data) အောက်၊ (/proc) အောက် ဆိုရင်တော့ System အောက် ပြန်သွားချင်ရင် *cd/system* command ကို အသုံးပြုနိုင်ပါတယ်။

### (၃) pwd

*pwd* ဆိုတာက *print working directory* ခဲ့ အတိုကောက်ပါ။ ပြောချင်တာက လက်ရှိ ကျွန်တော်တို့ဟာ ဘယ် directory ဘယ် partition အောက်ကနေ အလုပ်လုပ်နေတာလဲ၊ ကိုယ်က *cd* command တွေသုံးပြီး ဟိုဝင် ဒီဝင်နဲ့ ဘယ်အောက်

ရောက်နေလဲဆိုတာကို ကြည့်တဲ့အခါ အသုံးပြုပါတယ်။

### (၅) *mkdir*

make directory ရဲ့ အတိုကောက်ပါ။ ဖိုလ်ဒါ (Directory) အသစ်ဆောက်တဲ့အခါ ဒီ command ကို အသုံးပြုပါတယ်။ *mkdir <directory name>* ဆိုပြီး ကိုယ်ဆောက်ချင်တဲ့ ဖိုလ်ဒါ (Directory) နာမည်နဲ့တွဲပြီး အသုံးပြုရပါတယ်။

```
C:\Users\      >adb shell
$ ls
ls
persist
cust
config
cache
HwUserData
sdcard
.

$ cd sdcard
cd sdcard
$ pwd
pwd
/mnt/sdcard
$ mkdir thihasoe
mkdir thihasoe
$ ls
proc
init.rc
init.ubuntu.rc
init.ubuntu-touch.rc
init.ubuntu-touch-graphite.rc
init.ubuntu-graphite.rc
init.ubuntu-fishhook.rc
.

$ fault.pmp
thihasoe
```

အပေါ်ပုံမှာဆို ပထမဆုံး *adb shell* နဲ့ terminal ထဲဝင်ပါတယ်။ ပြီးတော့ *ls* ကိုသုံးပြီး ဘယ်ဖိုလ်ဒါတွေ၊ ပိုင်တွေရှိလဲဆိုတာ ကြည့်ပါတယ်။ *cd* ကိုအသုံးပြုပြီး *sd card* ထဲကို ပြောင်း *pwd* နဲ့ အခုဘက်ရှိ ဘယ်အောက်ရောက်နေလဲဆိုတာ ကြည့်ပါတယ်။ ပြီးတော့ *mkdir* နဲ့ ဖိုလ်ဒါတစ်ခုဆောက်တယ်။ တကယ်ရှိ၊ မရှိကို *ls* သုံးပြီး ပြန်စစ်ဆေးကြည့်တာကို ပြပေးထားတာပါ။

**(၅) rm**

`rm` ကိုတော့ ဖိုင်တွေ ဖျက်တဲ့အခါ အသုံးပြုပါတယ်။ `rm <filename>` ဆိုပြီးအသုံးပြုပါတယ်။ Directory တစ်ခုအောက်မှာရှိတဲ့ File အားလုံးကို ဖျက်ချင်တယ်ဆို `rm *.*` ဆိုပြီး အားလုံးကို ဖျက်ပစ်နိုင်ပါတယ်။ ကိုယ်ဖျက်မယ့် ဖိုင်လမ်းကြောင်း ကိုတော့ မှန်အောင်ရေးပေးဖို့ လိုပါတယ်။

**(၆) rmdir**

remove directory ပါ။ ဖိုလ်ဒါတွေကိုဖျက်ဖို့အတွက် အသုံးပြုပါတယ်။

**(၇) cat**

`cat` ကိုတော့ Android မှာ Configuration File တွေကို ဖတ်ဖို့အတွက် အသုံးပြုပါတယ်။ `cat <filename>` ဆိုပြီးအသုံးပြုနိုင်ပါတယ်။ ဖိုင်တွေကို လက်ရှိပေးထားတဲ့နာမည်ကနေ တွေ့နာမည်တစ်ခုကိုပြောင်းပြီး ကော်ပိကူးချင်တဲ့အခါ မှာလည်း `cat` command ကို အသုံးပြုနိုင်ပါတယ်။ ဥပမာ- "cat /system/test.txt /sdcard/test\_copy.txt ". အသုံးပြုခဲ့ပါတယ်။

**(၈) cp**

Terminal ကိုအသုံးပြုနေချိန် Android System အတွင်းမှာ ဖိုင်တွေကို ကော်ပိကူးဖို့အတွက် အသုံးပြုပါတယ်။ အသုံးပြုပုံကတော့ `cp <source><destination>` ပါ။

**(၉) ln**

In ကတော့ Directory ကိုဖြစ်ဖြစ်၊ Partition ကိုဖြစ်ဖြစ် တစ်နေရာရာကို shourtcut ထုတ်တဲ့အခါ အသုံးပြုပါတယ်။ ဥပမာ - `ln /data/data/test ဆိုပြီးရှုက်လိုက်ရင် /data` အောက်က data ဆိုတဲ့ ဖိုလ်ဒါဟာ (/) အောက်မှာ test ဆိုတဲ့ နာမည်နဲ့ shourtcut လာဖြစ်နေမှာပါ။ test ကိုဖွင့်ရင်လည်း /data အောက်က data ဖိုလ်ဒါ (directory) ကို ဖွင့်တာနဲ့ အတူတူပါပဲ။

**(၁၀) mv**

move လိုပဲခေါ်လိုက်ပါရစေ။ mv ကိုတော့ ဖိုင်၊ ဖိုလ်ဒါတွေ တစ်နေရာကနေ တစ်နေရာရွှေ့တဲ့အခါ ဒါမှုမဟုတ် ဖိုင်၊ ဖိုလ်ဒါတွေကို နာမည်ပြောင်းတဲ့အခါ မှာအသုံးပြုပါတယ်။ System ဖိုင်တွေကို ရွှေ့မယ် ဆိုရင်တော့ Root Access လိုပါ တယ်။ move လုပ်တဲ့အခါ ဖိုင်နာမည်၊ ဖိုလ်ဒါနာမည်တွေကို အတိအကျရေးပေးရပါ တယ်။ အသုံးပြုပုံကတော့ "mv <source> <destination>" ပါပဲ။

ဥပမာ - mv /data/local/tmp/su /system/bin/su ဆိုပြီး data အောက်က su ဆိုတဲ့ဖိုင်ကို /system အောက် bin ဖိုလ်ဒါထဲကို လုမ်းပြီးရွှေ့လိုက်တာပါ။ ကိုယ် လက်ရှိရောက်နေတဲ့ ဖိုင်လမ်းကြောင်းအောက်မှာပဲ a ဆိုတဲ့ ဖိုလ်ဒါကို b ဆိုပြီးနာမည်ပြောင်းချင်ရင်လည်း "mv a b" ဆိုပြီး အသုံးပြုနိုင်ပါတယ်။

**(၁၁) date**

Terminal ထဲကနေပဲ ကိုယ့် Device ရဲ့ အချိန်ကို ကြည့်လို့ရပါတယ်။

**(၁၂) uptime**

ကိုယ့်ရဲ့ clndroid A/Device ဟာ ပါဝါ စဖွင့်ပြီးတဲ့အချိန်ကနေ အခု အချိန်ထိတယ်လောက် ကြော်ပြီးဆိုတာကို စစ်ကြည့်နိုင်ပါတယ်။ အချိန်၊ နာရီ အတိအကျ ဖော်ပြုပေးပါတယ်။

**(၁၃) su**

սu ဆိုတာကတော့ Substitute User ရဲ့ အတိုကောက် (Super User မဟုတ်ပါဘူး) ဖြစ်ပါတယ်။ လက်ရှိအသုံးပြုနေတဲ့ ရှုံးရှုံးuser (\$) ကနေ root user (#) အဖြစ် ပြောင်းချင်တဲ့အခါမှာ အသုံးပြုပါတယ်။ Terminal ထဲမှာတော့ တခို့ Command တွေကို အသုံးပြုဖို့အတွက် su နဲ့ အရင်ဆုံး root user အနေနဲ့ ပြောင်းထားသင့်ပါတယ်။ ဘာဖြစ်လို့လဲဆိုတော့ Command တော်တော်များများဟာ ရှုံးရှုံး user နဲ့ run လို့ မရလိုပါ။

### (၁၄) ***busybox***

*busybox* ဆိုတာကတော့ Android System မှာအသုံးပြနိုင်တဲ့ command ပေါင်းများစွာကို တစ်ခုတည်းဖြစ်အောင် စုစည်းထားတဲ့ tool တစ်ခုပဲဖြစ်ပါတယ်။ *busybox* ကိုအသုံးပြုချင်ရင်တော့ မိမိရဲ့ Device ဟာ Root Access ရှိုးသား ဖြစ်နေရပါမယ်။ *busybox* ကို /system/bin အောက် ဒါမှုမဟုတ် System/bin အောက်ကို သီးသန့်ထပ်တင်ပေးရပါတယ်။

### (၁၅) ***ps***

ကိုယ့် Device မှာ အလုပ်လုပ်နေတဲ့ Process တွေကိုပြောပါတယ်။ ဘာနဲ့ တူသလဲဆိုတော့ ကွန်ပျူးတာရဲ့ TASK MANAGER နဲ့ ဆင်တူပါတယ်။ ကြည့်ကြည့် လိုက်ရင် user၊ pid(process ID)၊ ppid၊ name အစရှိသဖြင့် အသေးစိတ်ပြောပါတယ်။ Process တွေကို ကြည့်လို့ရပြီဆိုမှုတော့ ပိတ်လို့လည်းရပြီပေါ့။

### (၁၆) ***top***

*top* command ကတော့ TASK MANAGER ကိုပြောတာပါ။ လက်ရှိပရှိရမ်တွေ၊ ဆော့ဝဲတွေကနေ အသုံးပြုနေတဲ့ Cpu Usage၊ Memory Usage တွေကိုပါ ကြည့်နိုင်ပါတယ်။

### (၁၇) ***kill***

လက်ရှိအလုပ်လုပ်နေတဲ့ process တွေကို ပိတ်ဖြစ်ဖို့အသုံးပြုပါတယ်။ *kill* command ကိုအသုံးပြုဖို့အတွက် root access လိုအပ်ပါတယ်။ အသုံးပြုပုံကတော့ " kill [option (s)] process ID " ပဲဖြစ်ပါတယ်။ [option (s)] နေရာမှာဆိုရင် ဒီ process ကို ဘယ် level နဲ့ kill မလဲဆိုတာ ရေးပေးရပါတယ်။ level 1 to 9 ရှိပါတယ်။ ( 9 ) ကတော့ အမြင့်ဆုံး level ပါ။ ဒါကိုပဲ အသုံးပြုကြပါတယ်။ ဥပမာ - " kill -9 <process ID>"

### (၁၈) ***mount***

*mount* command ကတော့ ကိုယ့်ရဲ့ system မှာ disks တွေ၊ cd rom

တွေကို Directory တွေအဖြစ် ချိတ်ဆက်အသုံးပြုဖို့အတွက် အသုံးပြုရတာပါ။ android မှာတော့ အဲဒီလို ချိတ်ဆက်အသုံးပြုထားတာကို ပြရမယ်ဆိုရင် sd card ကိုပဲ ပြပေးရမှာပါ။ sd card ကို android system က /mnt ဆိုတဲ့ directory အောက်မှာ sdcard ဆိုတဲ့ ဖိုလ်ဒါနာမည်နဲ့ mount လုပ်ပေးထားတာပါ။ ဒါကြောင့် sdcard ကို ကျွန်တော်တို့ဟာ settings ထဲကနေသွားပြီး mount ၊ umount လုပ်လိုရတာပါ။ ဒါ mount command မှာ ပြောပြချင်တာက mount command ကိုအသုံးပြုပြီး system partition၊ /data နဲ့ /cust တိုကို သူတို့ရဲ့ default permission ဖြစ်တဲ့ read only အခြေအနေကနေ mount command ကိုအသုံးပြုပြီး read/write ရယူပုံရယူနည်းဖြစ်ပါတယ်။

```
# mount h
mount h
Usage: mount [-r] [-w] [-o options] [-t type] device directory
#
C:\Users\kmc>
```

အပေါ်ကပုံကတော့ mount command အသုံးပြုပုံကိုသိရအောင် "mount h" ဆိုပြီး help ကိုခေါ်ကြည့်လိုက်တာပါ။ ပုံမှန်အားဖြင့်တော့ ကျွန်တော်တို့ဟာ /system ဆိုတဲ့ system partition အပေါ် read/ write permission မရပါဘူး။ ကိုယ်က /system/app အောက်က system app တစ်ခုကို ဖျက်ချင်တယ်ဆို read/write Permission အရင်ယူရပါမယ်။

တကယ်လို /system ကို read/ write ယူချင်တယ်ဆိုရင်တော့ "mount -o rw,remount -t yaffs2 /dev/block/mtdblock4 /system " ဆိုပြီးရေးပေးရမှာပါ။ ဒီနေရာမှာ ရေးလိုက်တဲ့ command အကြောင်းနည်းနည်း ပြောပြချင်ပါတယ်။ အရှေ့ဆုံး mount ဆိုတဲ့ command ကိုသုံးတယ် [ -o ] ခံပြီး options နေရာမှာ rw,remount လိုရေးတယ်။ ကိုယ်လုပ်ချင်တဲ့အကြောင်းပေါ့။ [ -t ] ဆိုတာကတော့ ကိုယ် mount လုပ်မယ့် partition ရဲ့ file system ကိုပြောတာပါ။ sd card တွေဆို FAT/FAT32 တို့ပေါ့။ ဒီမှာတော့ /system ရဲ့ File system ဖြစ်တဲ့ yaffs2 ကိုရေးပေးထားတာပါ။ device directory နေရာမှာတော့ ကိုယ် mount လုပ်မယ့် directory ကိုရေးပေးရမှာပါ။

မေးစရာရှိလာနိုင်တာက yaffs2 ဆိုတာကရော ဘာလဲ တကယ်လို /system

အပိုင်းကို read/write ဖြစ်အောင် mount မလုပ်ချင်ဘဲ /cust အပိုင်းကို ပဲ mount လုပ်မယ့်ဆိုရင်ရော ဘယ်လိုလုပ်မလဲ /system ကို mount လုပ်တိုင်း ဒီcommand ကိုအလွတ်ကျက်ထားရမှာလား၊ မေ့သွားရင် ဘယ်သူမေးမလဲ၊ ဘယ်လိုသိနိုင်မလဲ ဆိုတာတွေပါပဲ။ ဟုတ်ကဲ့ပါ yaffs2 ရဲ့ File System အရှည်ကတော့ "Yet Another Flash File System - Version 2) ဖြစ်ပြီး yaffs က open source File System ပါ။ အရမ်းမြန်ဆန်ဖို့လိုအပ်တဲ့ NAND၊ NOR flash တွေမှာ အသုံးပြုဖို့အတွက် တိတွင်ဖန်တီးထားတာဖြစ်ပါတယ်။ mount command နဲ့ပတ်သက်ပြီး အသေးစိတ် သိချင်ရင် Terminal ထဲမှာ mount လိုပဲ ရိုက်လိုက်ပါ။

```
# mount
mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,relatime,mode=755 0 0
devpts /dev/pts devpts rw,relatime,mode=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpuacct 0 0
tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
tmpfs /mnt/obb tmpfs rw,relatime,mode=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/mtdblock4 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock6 /data yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock5 /cache yaffs2 rw,nosuid,noexec,relatime 0 0
/dev/block/mtdblock8 /cust yaffs2 ro,relatime 0 0
```

အပေါ်ကပုံမှာ ကြည့်လိုက်ရင် /system ဆိုရင်လည်း သူရဲ့ device directory ဟာဘာလဲဆိုတာတို့ အလွယ်တကူ သိနိုင်ပါတယ်။ အားလုံးအတူတူပါပဲ။ အလွတ်ကျက် ထားစရာမလိုတော့ပါဘူး။ partition တွေရဲ့ file system ဆိုရင်လည်း device directory ရဲ့နောက်မှာ ကပ်လျက်ဖော်ပြပေးထားပါတယ်။ အခုလို read/write အနေနဲ့ mount လုပ်ပြီးပြုဆိုရင်တော့ system အောက်ကနေ ကိုယ်ဖျက်ချင်တာ ထပ်ထည့်ချင်တာအားလုံး လုပ်လို့ရသွားပါပြီ။ mount ကို အသုံးပြုဖို့အတွက် root access လိုအပ်ပါတယ်။

## (၁၉) dd

*dd command* ကိုတော့ ပုံမှန်အတိုင်း Command သုံးပြီးဝင်ကြည့်ဖို့ မလွယ်တဲ့နေရာက ဖိုလ်ဒါတွေ၊ ဒါမှုမဟုတ် drive တစ်ခုလုံး၊ partition တစ်ခုလုံးကို တစ်နေရာကနေ တခြားတစ်နေရာ ကူးယူဖို့အတွက် အသုံးပြုပါတယ်။ အသုံးပြုနိုင်မယ့်

ပုံစံကတော့ " dd if=<input file> of=<output file> " အတိုင်းပဲဖြစ်ပါတယ်။ dd command ကိုအသုံးပြုမယ်ဆိုရင် အရမ်းသတိထားရပါမယ်။ " if " ရဲ့ နောက်က input file ဖြစ်ရမယ်၊ " of " ရဲ့နောက်က output file ပါ။ မှားသွားရင် file တစ်ခုလုံး Folder (Directory) တစ်ခုလုံး partition တစ်ခုလုံးပျက်သွားနိုင်ပါတယ်။ ဖုန်းတွေ ဆိုရင်လည်း brick အနေအထားဖြစ်သွားနိုင်ပါတယ်။ ဒါကြောင့် dd ကို နာမည်ပြောင် အနေနဲ့ " disk destroyer " ခေါ်ကြတာပါ။

```
dd if=/dev/block/mmcblk0p8 of=/sdcard/recovery.img
```

အပေါ်က command မှာဆိုရင် partition 8 mmcblk0p8 အောက်မှာ ရှိတဲ့ဖိုင်အားလုံးကို sd card ထဲကို recovery.img ဆိုတဲ့နာမည်နဲ့ ကူးယဉ်လိုက်တာပါ။ HTC Device တွေမှာ stock Recovery ကို Backey ဆွဲတဲ့ ပုံစံကို ပြထားတာပါ။ ကျွန်တော်တို့ Huawei phone တွေကို Firmware တင်မရတဲ့အခါ သူရဲ့ stock recovery ကို ပြန်တင်ပေးပြီးမှ Firmware ကို ပြန်တင်ရပါတယ်။ ဥပမာ Huawei Y320 ဆိုရင် သူရဲ့ firmware က dload နဲ့မလာဘဲ sdcard\_update.zip အနေနဲ့ လာပါတယ်။ ကျွန်တော်တို့က Firmware ပြန်တင်တဲ့အခါ recovery ကနေပြီး update from sd card ကနေတင်ပေးရပါမယ်။ အဲဒီအခါနဲ့ ကိုယ့်ဖုန်းက recovery ထဲ ဝင်မရဘူးဆိုရင်တော့ သူနဲ့အမျိုးအစားတူ Device ကနေ အခုလို Command ကို သုံးပြီး recovery.img ဖိုင်ကို backup ပြန်ယူပြီးမှ Firmware တင်မယ့်ဖုန်းမှာ fastboot ကနေပြီး flash လုပ်ပေးရပါမယ်။ recovery တင်ပြီးပဲဆိုရင်တော့ Firmware ကို ပြန်တင်လိုပါပြီ။ " dd " နေရာမှာ " cat /dev/block/mmcblk0p8 >/sdcard/recovery.img " ဆိုပြီး "cat" command ကိုလည်းအသုံးပြုပြီး backup ဆွဲလိုပါတယ်။ ဒါက နုမ္မနာအနေနဲ့ အသုံးဝင်ပုံကိုပြောပြတာပါ။ ဘယ်ဖုန်းကနေမဆို အခုလိုမျိုး backup ဆွဲလိုပါတယ်။ ဒါဆိုရင် အပြန်အလှန်အနေနဲ့ ကိုယ့်မှာရှိပြီးသား recovery.img ကိုလည်း Fastboot ထဲမဝင်ဘဲ adb ကနေပဲ dd command နဲ့ဖြစ်ဖြစ် တစ်ခါတည်း restore ပြန်လုပ်လိုပါတယ်။

ကျွန်တော့မှာ စိုးရိမ်စိတ်ရှိပါတယ်။ ကျွန်တော်ရောပြတာကို နားမလည်မှာ စိုးရိမ်ပါတယ်။ နားလည်တဲ့သူကလည်း ပေါက်ကရတွေ လျောာက်ရေးထားတယ်လို့ တင်မှာ စိုးရိမ်ပါတယ်။ အပေါ်က " dd " command အသုံးပြုပုံ နုမ္မနာမှာဆိုရင်

ကျွန်တော်ဆိုရင်လည်း စာရေးသူကို မေးစရာရှိပါတယ်။ ဘာလို့ recovery ကို mmcblk0p8 အပိုင်းကနေပဲ backup ဆွဲတာလဲ၊ mmcblk0p8 ဆိုတာက ဘာကို ခေါ်တာလဲ ဆိုတာမျိုးပါ။ အများကြီးရှင်းပြချင်ပါတယ်။ ဒါပေမဲ့ ပို၍၍သွားမှာစိုးလို့ နည်းနည်းတော့ပြောပြုမယ်များ။ dd command နဲ့တော့ မဆိုင်ပါဘူး။ ဒါပေမဲ့ recovery ကို backup ဆွဲတဲ့အခါမြင်ရမယ့် အစိမ်းသက်သက်စကားလုံးတွေကို နားလည်ဖော်လို့ပါ။

Android Device တွေမှာ မတူညီတဲ့ data တွေကိုသိမ်းဖို့အတွက် (ဥပမာ recovery၊ booti၊ system၊ data၊ cache) partition တွေအများကြီးရှိပါတယ်။ Android device တွေဟာ စက်ရုတ်ခုတည်းကထုတ်ရင်တောင် တစ်ခုနဲ့တစ်ခု တည်ဆောက်ပုံ၊ ထုတ်လုပ်ပုံခြင်း မတူညီကြပါဘူး။ ဖုန်းတွေမှာဆိုရင် MTD (Memory Technolog Device) ကိုအခြေခံတဲ့ Device တွေ EMMC (Embedded MultiMedia Card) ကိုအခြေခံပြီးထုတ်တဲ့ Device တွေနဲ့ MMC (MultiMedia Card) ကိုအခြေခံပြီး ထုတ်တဲ့ Device ရယ်ဆိုပြီး အမျိုးမျိုးကွဲပြားကြပါတယ်။ MTD Devices တွေနဲ့ EMMC Devices တွေကတော့ partition ပုံစံကြည့်တာ အတူတူပါပဲ။ MTD Device ဆို /proc အောက်က mtd ဆိုတဲ့ဖိုင်ကို လှမ်းဖတ်ကြည့်ရပြီး EMMC ဆို /proc အောက်က emmc ဖိုင်ကို " cat /proc/emmc " ဆိုပြီး cat command နဲ့ လှမ်းဖတ်ကြည့်ရတာပါ။ အောက်မှာ EMMC Device တစ်ခုဖြစ်တဲ့ HTC Desire S ရဲ့ emmc ဖိုင်ကိုနှုန်းဖတ်ပြထားပါတယ်။

`cat /proc/emmc`

<b>dev:</b>	<b>size</b>	<b>erasesize</b>	<b>name</b>
mmcblk0p17:	00040000	00000200	"misc"
mmcblk0p21:	0087f400	00000200	"recovery"
mmcblk0p22:	00400000	00000200	"boot"
mmcblk0p25:	22dff00	00000200	"system"
mmcblk0p29:	002ffc00	00000200	"local"
mmcblk0p27:	090ffe00	00000200	"cache"
mmcblk0p26:	496ffe00	00000200	"userdata"
mmcblk0p30:	014bfe00	00000200	"devlog"
mmcblk0p31:	00040000	00000200	"pdata"
mmcblk0p28:	09800000	00000200	"lib"

တကယ်လို ကိုယ့်ရဲ Device က MMC base ဖြစ်ခဲ့ရင်တော့ ကိုယ် Device ရဲ Partition ပုံစံကိုသိချင်ရင် /proc အောက်က partitions ဆိုတဲ့ဖိုင်ကို " cat /proc/partitions " ဆိုပြီး ဖတ်ကြည့်ရမှာပါ။

```
shell@android:/ # cat /proc/partitions
cat /proc/partitions
major minor #blocks name
179      0      3817472 mmcblk0
179      1          20 mmcblk0p1
179      2          300 mmcblk0p2
179      3      133120 mmcblk0p3
179      4             1 mmcblk0p4
179      5      12288 mmcblk0p5
179      6          4096 mmcblk0p6
179      7          4096 mmcblk0p7
179      8      3072 mmcblk0p8
179      9      3072 mmcblk0p9
179     10      3072 mmcblk0p10
179     11          4096 mmcblk0p11
179     12          8192 mmcblk0p12
179     13      20480 mmcblk0p13
179     14          4096 mmcblk0p14
179     15      196608 mmcblk0p15
179     16          81920 mmcblk0p16
179     17      524288 mmcblk0p17
179     18      655360 mmcblk0p18
179     19      2154496 mmcblk0p19
shell@android:/ #
```

EMMC Device တွေနဲ့ MTD အခြေခံ Device တွေမှာ နာမည်တွေနဲ့ partition တွေကို တွဲပြီးမပြေားနိုင်ပါဘူး။ အပေါ်ကပုံမှာကြည့်လိုက်ရင် cat/proc/partition command ရဲ result မှာ အပေါ်ဆုံးက mmcblk0 ဆိုတာ "MultiMedia Card Block 0 " ဖြစ်ပါတယ်။ အဲဒီ block0 ကိုထပ်ပြီး partition ခွဲထားတာ 19 ပိုင်းရှုပါတယ်။ တခြား Device တွေမှာ ဒီထက်များချင်လည်း များနိုင်ပါတယ်။ block0 ကို partition ခွဲထားတဲ့ ပထမဆုံးအပိုင်းကို partition 1(p1) လိုအပ်မှတ်ပြီး နောက်ဆုံးအပိုင်းဖြစ်တဲ့ partition 19 ကို (p19) လိုအပ်မှတ်ထားတာပါ။

ကျွန်တော်အခု Huawei C8810 ကို ကွန်ပျော်ဘာနဲ့ ကြိုးတပ်ပြီးသူရဲ့ recovery ကို backup ဆွဲတာ ပြပါမယ်။ အရင်ဆုံး adb shell လိုရှိက်ပြီး terminal ထဲကိုသွားပါမယ်။ Huawei ရဲ recovery partition က ဘယ်အပိုင်းလဲ ဆိုတာကို " cat /proc/mtd " လိုရှိက်ပြီး အရင်စစ်ကြည့်ပါမယ်။

```
cat /proc/mtd
dev:    size   erasesize  name
mtd0:  00500000  00020000  "boot"
mtd1:  00500000  00020000  "recovery"
mtd2:  00140000  00020000  "misc"
mtd3:  00060000  00020000  "splash"
mtd4:  0ba00000  00020000  "system"
mtd5:  03f00000  00020000  "cache"
mtd6:  0a0a0000  00020000  "userdata"
mtd7:  01400000  00020000  "userdata2"
mtd8:  01400000  00020000  "cust"
# -
```

အပေါ်ကပံမှာကြည့်လိုက်ရင် သူရဲ့ recovery partition က mtd1 မှာရှိပါတယ်။ သူကို backup ဆွဲမယ်ဆို dd command ကိုပဲ အသုံးပြုပြီးဆွဲပါမယ်။

```
dd if=/dev/mtd/mtd1 of=/sdcard/recovery.img bs=4096
```

အခုရှိကိုလိုက်တဲ့ command မှာဆို အားလုံးက သိပြီးသားပါပဲ။ နောက်ဆုံးမှာ bs=4096 ဆိုတာပဲ အပိုပါလာတာပါ။ **bs** ဆိုတာကတော့ block size ကိုရည်ညွှန်ပြီး partition တွေကို ကော်ပိကူးတယ်ဆို သူရဲ့ပုံသေ block size က 4096 ဖြစ်ပါတယ်။ အကယ်၍ Disk တစ်ခုလုံးကိုသာကူးမယ်ဆိုရင်တော့ bs=1M လို့ ရေးပေးရပါမယ်။ block size ကတော့ ရေးပေးရင်လည်းရသလို မရေးလည်းဘဲရပါတယ်။

### (j) chmod

အခု Command ကတော့ Terminal ထဲကနေ Android က ဖိုင်တွေရဲ့ Permission ကိုသတ်မှတ်ပေးဖို့အတွက် အသုံးပြုပါတယ်။ ဘယ်ဖိုင်တွေကို ဘယ်သူ ကိုပေးသုံးမယ်၊ ပေးမသုံးဘူးဆိုတာတွေကို သတ်မှတ်ပေးနိုင်ပါတယ်။ Android ကလည်း လုံခြုံရေး အများကြီးလုပ်ထားပါတယ်။ တော်ရုံတန်ရုံနဲ့ အသုံးပြုသူဟာ မဆိုင်တာတွေ မလုပ်မိအောင် ထိန်းချုပ်ထားပေးပါတယ်။ အရင်ဆုံး ကိုယ့် ANDROID SYSTEM မှာ ဘယ်ဖိုင်တွေကို ဘယ်လို့ Permission ပေးထားတယ်ဆိုတာကို သိဖို့အတွက် ls -l နဲ့ကြည့်လိုက်ပါ။

# ls -l
drwxrwx---x system system 1980-01-06 06:30 persist
drwxr--xr-x root root 2012-02-07 17:06 cust
dr-x----- root root 1980-01-06 06:30 config
drwxrwxr-x system cache 2014-02-05 12:04 cache
drwxrwxr-x system sdcard_rw 1970-01-01 06:30 HwUserData
lrwxrwxrwx root root 1980-01-06 06:30 sdcard -> /mnt/sdcard
drwxr--xr-x root root 1980-01-06 06:30 acct
drwxrwxr-x root system 1980-01-06 06:30 mnt
lrwxrwxrwx root root 1980-01-06 06:30 vendor -> /system/vendor
lrwxrwxrwx root root 1980-01-06 06:30 d -> /sys/kernel/debug
lrwxrwxrwx root root 1980-01-06 06:30 etc -> /system/etc
drwxr-xr-x root root 1970-01-01 06:30 wifi
-rw-r--r-- root root 6072 1970-01-01 06:30 ueventd.rc
drwxr--xr-x root root 2014-03-03 18:38 system
drwxr--xr-x root root 1980-01-06 06:30 sys
drwxr-xr-x root root 1970-01-01 06:30sbin
dr-xr--xr-x root root 1970-01-01 06:30 proc
-rwxr-xr-x root root 27193 1970-01-01 06:30 init.rc
-rwxr-xr-- root root 1593 1970-01-01 06:30 init.mem.rc
-rwxr-xr-- root root 6705 1970-01-01 06:30 init.huawei.rc
-rwxr-xr-- root root 1513 1970-01-01 06:30 init.highmem.rc
-rwxr-xr-- root root 1622 1970-01-01 06:30 init.goldfish.rc

စာကြောင်းတွေရဲ့ ရှေ့ဆုံးစာလုံးကိုကြည့်လိုက်ရင် ( d ) နဲ့စတာရှိမယ် ( 1 ) နဲ့စတာရှိမယ် ( b ) နဲ့စတာလည်းရှိပါတယ် ဒီပုံမှာ ( b )နဲ့စတာမပါလိုပါ။ d=Directory (Folder) l=link (Shourtcut) b=binary (configuration file) တွေပဲဖြစ်ပါတယ်။ သူတို့နောက်မှာပါတဲ့ စာလုံး ၉ လုံး (rwxrwxrwx) ကတေသ့ permission သတ်မှတ်ချက် ပဲဖြစ်ပါတယ်။ စာလုံး ၉ လုံးနဲ့ အပ်စု ၃ ခုကိုပေးထားတာပါ။ ပထမ ၃ လုံး ဖြစ်တဲ့ (rwx-----) က ဖိုင် ဖိုလ်ဒါရို့ owner permission ပါ။ ဒုတိယ ၃လုံး (---rwx---) က အဲဒီ ဖိုင် ဖိုလ်ဒါကို ပိုင်တဲ့ Group ရဲ့ permission ဖြစ်ပါတယ်။ နောက်ဆုံး တတိယ ၃လုံး (-----rwx)က Group ထဲမှာမပါတဲ့ အခြား user တွေအတွက်ပါ Windows ကွန်ပူးတာမှာအခါနဲ့ဆို Everyone အတွက်ပါ။

r (Read permission) ဆိုရင် ဖတ်လိုရမယ်၊ ကြည့်လိုရမယ်၊ ကော်ပိကုံးလို ရမယ်။ w (Write permission) ဆို ပြင်လိုရမယ်၊ ဖျက်လိုရမယ်၊ နာမည်ပြောင်းလို ရမယ်၊ ဖိုင်အသစ်၊ ဖိုလ်ဒါအသစ်ဆောက်လိုရမယ်။ x (Execute permission) ဆိုရင် တေသ့ Execute ပါလိုနိုင်ပါတယ်။ ဒါ Command ကတေသ့ ဖိုလ်ဒါတွေပေါ်မမှတည် ပါဘူး။ အမိက command တွေ၊ တကယ်အရေးကြီးတဲ့ configuration file တွေအပေါ်သက်ရောက်ပါတယ်။ Terminal ထဲကနေ ls -l နဲ့ကြည့်လိုသာ အခုလို rwx Format(Alphabetic) နဲ့ပြတာပါ။ ကျွန်တော်တို့ သတ်မှတ်ပေးမယ်ဆိုရင်တေသ့ အခုလို rwx Format နဲ့ သတ်မှတ်ပေးလိုမရပါဘူး။ Numeric method နဲ့ပဲ သတ်မှတ်ပေးရမှာပါ။ အောက်မှာကြည့်လိုက်ပါ rwx ကို ငါကန်းနဲ့ ဘယ်လို သတ်မှတ်တယ်ဆို တေသ့ ..

1	:	--x
2	:	-w-
3	:	-wx
4	:	r--
5	:	r-x
6	:	rw-
7	:	rwx

အခုပြထားတာက rwx အုပ်စုတစ်စုချင်းစီရဲ့တန်ဖိုးကို ရေးပြထားတာပါ။ တကယ် permission ပေးရင်တော့ အုပ်စု ၃ ခုလုံးအတွက် သတ်မှတ်ပေးရမှာပါ။ အသုံးပြုရမယ့်ပုံစံကို လေ့လာကြည့်ပါညီ။

```
#chmod <mode> <file>
```

chmod ကတော့ permission ပေးဖို့အတွက်အသုံးပြုတဲ့ command ပါ။ mode နေရာမှာ ကိုယ်သတ်မှတ်ပေးချင်တဲ့ permission ကို Numeric နည်းနဲ့ သတ်မှတ်ပေးရပါမယ်။ File ဆိုတာကတော့ ကိုယ် Permission သတ်မှတ်ချင်တဲ့ ဖိုင်နာမည်၊ ဖိုလ်ဒါနာမည်ကိုရေးပေးရမှာ။ chmod ကိုအသုံးပြုဖို့အတွက် root access ရှိဖို့လိုအပ်တယ်ဆိုတာကိုတော့ ထည့်မပြောတော့ပါဘူး။ အရင်ခုံးအနေနဲ့ 'test' ဆိုတဲ့ ဖိုင်ကို Permission ပေးကြည့်ပါမယ်။ ဘယ်လိုပေးမလဲဆိုတော့ owner ကို rwx(read/write/execute) ပေးမယ်၊ Group ကိုတော့ (read/execute)၊နောက်ဆုံး Other လိုခေါ်တဲ့ Everyone ကိုလည်း rx(read/execute) သတ်မှတ်ပေးပါမယ်။ ပထမ owner က rwx ဆိုတော့ (7)၊ နောက် Group က rx ဆိုတော့ (5)၊ Other ကိုလည်း rx ဆိုတော့ (5) ပေါင်းပြီးတော့ 755 ဆိုပြီး mode နေရာမှာအစားထိုးပြီး ရေးပေးရပါမယ်။

```
#chmod 755 test
```

အပေါ်က command မှာ test ဆိုတာက execute (run) လုပ်နိုင်တဲ့

ဖိုင်(ဥပမာ - စာ busibox) ဖြစ်နေလို့ execute ကိုပါအောင် permission မှာသတ်မှတ်ပေးတာပါ တကယ်လို့ ကျွန်ုတ်တို့ဟာ execute မလုပ်နိုင်တဲ့ဖိုင်တစ်ခုခဲ့ (ဥပမာ- font: apps) ကိုသာ permission သတ်မှတ်ပေးချင်တယ်ဆို ပုံသေပေးနေကျ ကတော့ 644 ပါ၊ စာဖတ်တဲ့သူတွေနဲ့လည်းမစိမ်းဘူးလို့ထင်ပါတယ်။ 644 ဆိုရင် ပထမကြောင်း (6) က owner အတွက် rw (read/write)၊ ဒုတိယကြောင်း (4) က Group အတွက် r (read only) တတိယကြောင်းက Other အတွက် သူလည်း (4) ပဲမို့လို့ r (read only) ပါ။

```
#chmod 644 /system/app/Superuser.apk
```

*chmod* မှာအပေါ်ကရေးပြုတဲ့ Permission တွေတင်မကဘဲ super permission ဆိုတာ ရှိပါသေးတယ် လူသုံးနည်းပါတယ်။ အသိသာဆုံး ဥပမာပြုရမယ် ဆိုရင် root ဖောက်ပြီးသား android device ရဲ့ /system/bin အောက်က su ဆိုတဲ့ဖိုင်ရဲ့ permission ကို " ls -l /system/bin/su ဆိုပြီးကြည့်လိုက်ပါ။ များသောအားဖြင့် Vroot နဲ့ဖောက်ထားရင်သူရဲ့ permission ဟာ -rwsr-sr-x ဆိုပြီးပြပါတယ်။ သူရဲ့ permission မှာပါတဲ့ (s) ဆိုတာက super permission သုံးထားတယ်ဆိုတာပြတာပါ။ တစ်ခြားနည်းနဲ့ root ဖောက်ထားရင်တော့ su ရဲ့ permission တို့ -rwxr-xr-x (755) ပဲ ပုံသေသတ်မှတ်ပေးကြပါတယ်။ Vroot မှာတော့ (s) ပါလာတော့ Numeric နဲ့ကြည့်ရင် 4755 ပါ အရှေ့ဆုံးက (4) က (s) ရဲ့တန်ဖိုးပါ။ အောက်မှာ သူတို့ရဲ့ တန်ဖိုးတွေကို ထပ်ပြီးရှင်းပြထားပါတယ်။

4000 s-----	suid
2000 s-----	sgid
1000 t-----	sticky

Super permission ဖြစ်တဲ့ (suid, sgid, sticky) တို့ကို ပြန်ပြီး clear လုပ်ချင်ရင်၊ ဒါမှုမဟုတ် ရှိုးရှိုး permission အဖြစ်ပြန်ပြောင်းချင်ရင် ကိုယ်သတ်မှတ်ချင် တဲ့ permission(mode) ရဲ့ အရှေ့မှာ zero (0) ထည့်ပေးရပါမယ်။ အပေါ်က su ကိုကြည့်ပြီး 4755 ကနေ 755 ပြောင်းချင်ရင် mode မှာ 0755 လို့

ရေးပေးမှုရမှာပါ။ တစ်ခါတည်း suid၊ sgid စံ sticky တို့ကြား ခြားနားချက် ကွာခြားချက်တွေကို အနည်းငယ် ပြောပြချင်ပါသေးတယ်။ နည်းနည်းပါ .. သေချာပါတယ်။ နည်းနည်းဆို မဖြီးတော့ ဘူးဆိုတာ။

sticky ကို အချို့နေရာတွေမှာ Stick bit လိုလည်း ဖော်ပြကြပါတယ်။ sticky permission (t) ကိုတော့ များသောအားဖြင့် ဖို့လိုပါတွေအပေါ်မှာ သတ်မှတ်ကြပါတယ်။ sticky permission (t) သတ်မှတ်ပေးထားတဲ့ ဖို့လိုပါအောက်က ဖိုင်တွေကို သူ့ရဲ့ owner ကိုယ်တိုင်မှုသာ ဖျက်လို့ရပါတယ်။ ဥပမာ /tmp ဖို့လိုပါဆိုရင် ဘယ် user မဆို ဖိုင်တွေ လာထည့်ထားလို့ရပါတယ်။ ဒါပေမဲ့ ကိုယ့်ထည့်ထားတဲ့ဖိုင်ကို ကိုယ်ကိုယ်တိုင်ပဲ ဖျက်လို့ရပါတယ်။

suid ကတော့ Execute လုပ်နိုင်တဲ့ ဖိုင်တွေကို run တဲ့အခါအမြှေတမ်း owner နာမည်နဲ့ပဲ အလုပ်လုပ်ပေးပါတယ်။ ဥပမာ ကျွန်တော်က root user ဆိုရင် su ဖိုင်ကို တခြားတစ်ယောက်ကို run ခွင့်ပေးချင်ပေးလို့ရတယ်၏ မပေးချင်လည်းရတယ်။ အသုံးပြုခွင့်ပေးလိုက်ရင်လည်း အဲဒဲ့ command ကို run တဲ့အခါ owner နာမည်က ကိုယ့်နာမည်နဲ့ပဲ အလုပ်လုပ်တယ်။ သူမှာလည်း root access ရှိတယ်ဆိုရင်တောင် owner change လို့မရပါဘူး။

sgid ကတော့ suid နဲ့ အတူတူပါပဲ။ ကွဲပြားချက်ကတော့ သူကိုတော့ ဖို့လိုပါတွေပေါ်မှာလည်း သတ်မှတ်နိုင်ပါတယ်။ sgid နဲ့သတ်မှတ်ထားတဲ့ ဖို့လိုပါအောက်မှာ တစ်ခြား ဖိုင်၊ ဖို့လိုပါတွေလာထည့်ရင်လည်း အားလုံးက Group Ownership ရပါတယ်။ များသွားပြီထင်ပါတယ်။ ဒီလောက်ဆို chmod နဲ့ပတ်သက်ပြီး permission ပေးတာ ပေးတတ်ပြီးလို့ ထင်ပါတယ်။

### (j) chown

chown ဆိုတာ change owner ရဲ့အတိကောက်ဖြစ်ပါတယ်။ ဒီ command ကတော့ ဖိုင်၊ ဖို့လိုပါတွေရဲ့ owner ကိုပြောင်းပေး ပြောင်းယူလုပ်တဲ့အခါ အသုံးပြုပတယ်။ ပထားအနေနဲ့ ဘယ်ဖိုင်တွေဟာ ဘယ် Owner ပိုင်တာလဲ၊ ဘယ် Group က ပိုင်တာလဲဆိုတာ ကြည့်ကြည့်လိုက်ပါ။ ဒါမှသာလျှင် ကိုယ်က Owner ပြောင်းပြီးသွားရင် တကယ်ပြောင်း၊ မပြောင်း ပြန်စစ်ကြည့်လို့ရမှာပါ။ ဒီအခန်းမှာတော့ root user နဲ့ သာမန် user ပြောင်းယူတာပဲ ရေးပြပေးမှာပါ။ ကိုယ့် ANDROID DEVICE

မှာလည်း user တွေကိုကြိုက်သလောက်ဖွဲ့လိုပါတယ်၊ ဖွဲ့ပြီးရင်လည်း ပြန်ဖျက်လို့ရ ပါတယ် ဆိုတာကို ကျွန်တော် အရှေ့ကအခန်းမှာ "adb shell pm create-user <USER\_NAME>" ဆိုပြီး package manager အသုံးပြုခြင်းအခန်းမှာ ရေးပြခဲ့ပါ တယ်။ ဒါအခန်းရောက်မှ User တွေကို စမ်းသပ်တဲ့အနေနဲ့ဖွဲ့ကြည့်ပြီးအသုံးပြုကြည့်ပါ။ အရင်ဆုံး ဘယ်ဖိုင်ကို ဘယ်သူပိုင်တာလဲဆိုတာသိရအောင် ls -l နဲ့(/) အောက်ကိုပဲ ကြည့်လိုက်ပါ။

shell@android_id:/ # ls -l					
drwxr-xr-x	root	root	970-01-01 07:00	acct	1 -> /sys/kernel/debug
drwxrwxr-x	root	system	2014-03-25 00:42	cache	
drwxr-xr-x	root	root	970-01-01 07:00	charged	
drwxr-xr-x	root	root	970-01-01 07:00	config	
drwxr-xr-x	root	root	970-01-01 07:00	cust	
drwxrwxr-x	root	system	1970-01-01 07:00	data	
drwxr-xr-x	root	root	2014-03-22 00:24	default.prop	
drwxr-xr-x	root	root	118 1970-01-01 07:00	dev	
drwxrwxr-x	root	root	1970-01-01 07:00	etc -> /system/etc	
drwxr-xr-x	root	root	1970-01-01 07:00	init	
drwxr-xr-x	root	root	102896 1970-01-01 07:00	init.goldfish.rc	
drwxr-xr-x	root	root	2344 1970-01-01 07:00	init.huawei.rc	
drwxr-xr-x	root	root	20446 1970-01-01 07:00	root	

နံပါတ် (1) နေရာက ဖိုင်၊ ဖိုလ်ဒါတွေအတွက် permission။ နံပါတ် (2) က ဖိုင်၊ ဖိုလ်ဒါတွေရဲ့ Owner နာမည်။ နံပါတ် (3) က Owner ရဲ့ Group။ နံပါတ် (4) က File size။ နံပါတ် (5) က ဖိုင်၊ ဖိုလ်ဒါတွေရဲ့ Modified Date နဲ့ နံပါတ် (6) ကတော့ ဖိုင်နာမည်၊ ဖိုလ်ဒါနာမည်ပဲ ဖြစ်ပါတယ်။ အခြောင်းမှုအပိုင်းက chmod (ပိုင်ရှင်ပြောင်းတာ) ဖြစ်လို့ နံပါတ် (2) နဲ့ သူနဲ့ဆက်စပ်နေတဲ့ နံပါတ် (3) အကြောင်းကို ဆက်ရေးပြပါမယ်။

chmod ကိုအသုံးပြုမယ်ဆို သူရဲ့ အသုံးပြုပဲ command ကတော့

#chown <user>[.GROUP] <FILE1> <FILE2> ...
---

ဆိုပြီး အပေါ်က ပြထားတဲ့အတိုင်းရေးပေးရမှာပါ။ ရွှေ့က အခန်းတွေမှာ ရှင်းပြခဲ့ပါတယ်။ <->နဲ့ [-] ကွာခြားချက်ကို <->ထဲက command က မဖြစ်မနေ ထည့်ပေးပြီး [--] ထဲက command ကတော့ မထည့်လည်းရတယ်၊ ထည့်လည်းရတယ်။ Optional ပေါ့။

#chown root test.txt
----------------------

test.txt ဆိုတဲ့ ဖိုင်ရဲ့ owner ကို root ဖြစ်အောင် ပြောင်းတာပါ။ ဖိုင်အများကြီးကို တစ်ခါတည်း ပြောင်းနိုင်ပါတယ်။ တကယ်လို ဖိုင်ရဲ့ ownership တင်မကဘဲ Group ownership ပါ တစ်ခါတည်းပြောင်းချင်တယ်ဆိုရင်တော့?

```
#chown root.root test.txt
```

ဆိုပြီးရှုက်ပေးရပါမယ်။ user name နဲ့ permission ပေးနိုင်သလို user id၊ Group id နဲ့လည်းပေးနိုင်ပါတယ်။ **ID** ကြည့်တာကတော့ terminal ထဲမှာ id လိုရှိက်ထည့်ပြီး ကိုယ့် android device ရဲ့ user id၊ Group id တွေကို ကြည့်လိုရပါတယ်။ root ဖောက်ပြီးသား ဖုန်းတွေရဲ့ uid၊ gid တွေဟာ များသောအားဖြင့် 0 (zero) ဖြစ်နေတက်ပြီး root မဖောက်ရသေးရင် များသောအားဖြင့် uid=2000၊ gid=2000 အဖြစ်အတွေ့များပါတယ်။

```
#chown 0.0 /system/app/RootExplorer.apk
```

## (JJ) svc

အခုရေးပြုမယ့် svc command ကတော့ ကျွန်တော်တို့ရဲ့ Android Device ကို ကွန်ပျံဗျာမှာ ကြိုးထိုးပြီး အသုံးပြုနေစဉ် wi-fi ကို ဖွင့်မယ်၊ ပိတ်မယ်၊ Mobile Data ကိုဖွင့်မယ်၊ ပိတ်မယ်၊ power ပိုင်းနဲ့ပတ်သက်ပြီးလည်း "stay awake" ကို ဖွင့်မယ်၊ ပိတ်မယ် အစရှိသဖြင့် အသုံးပြုနိုင်ပါတယ်။ root access ရှိဖို့တော့လိုအပ်ပါတယ်ခင်ဗျာ။ Terminal ထဲကနေအသုံးပြုနိုင်တဲ့ command တွေက အများကြီးကျွန်ပါသေးတယ်။ ဒီအခန်းမှာတော့ တကယ်အသုံးများပြီး အသုံးတည့်မယ့် command တွေပဲ ရေးပေးထားတာကြောင့် မပြည့်မစုံ ဖြစ်နေနိုင်ပါတယ် ဆိုတာကို ကြိုးတင်ပြီးတောင်းပန်ထားပါရမွေး။ အရင်ဆုံး svc command ကနေ ဘာတွေအသုံးပြုနိုင်မလဲလို သိရအောင် Terminal ထဲမှာ svc လိုပဲ ရှိက်ကြည့်ပါမယ်။

```
# svc
svc
Available commands:
    help      Show information about the subcommands
    power     Control the power manager
    data      Control mobile data connectivity
    wifi      Control the Wi-Fi manager
# -
```

အပေါ်ကပုံမှာဆိုရင် svc command မှာ အသုံးပြနိုင်တာတွေကို ပြပေးထားပါတယ်။ " svc help " လို့ ရိုက်ရင်လည်း svc ချဉ်းသက်သက်ရိုက်လို့ ကျလာမယ့် result နဲ့ အတူတူပါပဲ။ svc command မှာ ဘာတွေအသုံးပြနိုင်တယ်၊ ဘယ်လို့ အသုံးပြနိုင်တယ်ဆိုတာကို ပြပေးပါတယ်။ ဒုတိယတစ်ခုအနေနဲ့ " svc power " command ကတော့ ဖုန်းအားသွင်းနေတဲ့အချိန် Screen လင်းနေဖို့၊ ပိတ်နေဖို့အတွက် အသုံးပြုတာပါ။ အောက်မှာ " svc power " ဆိုပြီး power command နဲ့ပတ်သက်ပြီး အသုံးပြနိုင်တာတွေကို ပုံနဲ့တက္က ပြပေးထားပါတယ်။

```
# svc power
svc power
Control the power manager

usage: svc power stayon [true|false|usb|ac]
        Set the 'keep awake while plugged in' setting.
```

အသုံးပြနိုင်ပုံကတော့ " svc power stayon " လို့ရှိပြီး သူနောက်ကနေ true ဆိုရင် မိမိဖုန်းဟာ အားသွင်းနေတဲ့အချိန် Screen လင်းနေမယ်၊ false ဆိုရင် မလင်းဘဲ ပိတ်နေမယ်၊ usb ဆိုတာကို ထည့်ရှိက်မယ်ဆို ကိုယ့်ဖုန်းက ကွန်ပျူးတာမှာ usb cable ကနေ ချိတ်ထားတဲ့အချိန်လင်းနေမယ်၊ အားသွင်းကြိုးနဲ့ပဲလပ်ထိုးပြီး အားသွင်းနေတဲ့အချိန် မလင်းဘူး၊ အပြန်အလှန် ac command ကိုအသုံးပြုမယ်ဆို အားသွင်းနေတဲ့အချိန်လင်းမယ်၊ ကွန်ပျူးတာနဲ့ချိတ်သုံးနေတဲ့အချိန်မလင်းဘူး။ ကျွန်တော်တို့ ပုံမှန်အတိုင်း Settings > Developer options ထဲကနေသာ သွားပြီး လုပ်မယ်ဆို သူက stay awake ကို အဖွင့်၊ အပိတ် ၂ ခုပဲလပ်ပေးနိုင်ပါတယ်။ အခဲလို Terminal ထဲကဆို ကိုယ့်စိတ်ကြိုက်ပို့ပြီး အလုပ်လုပ်နိုင်ပါတယ်။

```
# svc data
svc data
Control mobile data connectivity

usage: svc data [enable|disable]
        Turn mobile data on or off.

svc data prefer
        Set mobile as the preferred data network
```

svc ကနေအသုံးပြနိုင်တဲ့ ဒုတိယ Command ကတော့ Mobile Data ကို Terminal ထဲကနေ Command အသုံးပြုပြီး အဖွင့်အပိတ် လုပ်နိုင်ပါတယ်။ သူက တော့ရှင်းပါတယ် mobile data ဖွင့်မယ်ဆို " svc data " ရဲ့ နောက်ကနေ enable ထည့်ရှုက်ပေးပြီး၊ ပိတ်မယ်ဆို disable ထည့်ရှုက်ပေးရဲ့ပါ။ နောက် svc ကနေအသုံးပြု နိုင်တဲ့ တတိယ command ကတော့ ကိုယ့်ဖုန်းရဲ့ wifi ကို အဖွင့်အပိတ်လုပ်နိုင်တာပါပဲ။

```
# svc wifi
svc wifi
Control the Wi-Fi manager

usage: svc wifi [enable|disable]
        Turn Wi-Fi on or off.

svc wifi prefer
        Set Wi-Fi as the preferred data network
```

ဆက်ရှင်းပြန့်တောင် မလိုတော့ဘူးထင်ပါတယ်။ ဖွင့်မယ်ဆို enable ပိတ် မယ်ဆို disable ပေါ့။ ဥပမာ - " svc wifi enable " ဒါမှာမဟုတ် " svc wifi disable " ပါ။

### (၂၃) *sqlite3*

*sqlite3 command* ကလည်း တကယ်ဆို Android Device တွေမှာတစ်ခါတည်း builtin ပါလာတာမဟုတ်ပါဘူး။ su တို့လိုပဲ ကိုယ့်ဘာသာ path တစ်ခုခဲ့အောက်ကို ကူးထည့်ပြီး တစ်နည်းအားဖြင့် မိမိ Device ထဲမှာ Install လုပ်ပြီးမှ အသုံးပြနိုင်တာပါ။ cd အခွဲထဲမှာ တစ်ပါတည်း ထည့်ပေးလိုက်ပါတယ်။ မိမိတို့ စိတ်ကြိုက် command အသုံးပြုပြီး /system/bin အောက်ခါမှုမဟုတ် /system/xbin အောက်ကိုပဲ ကူးကူး GUI ကနေပြီး " root explorer " အသုံးပြုပြီးပဲ ကူးကူး

ကူးထည့်နိုင်ပါတယ်။ တစ်ခုပဲသတိထားစေချင်တာက sqlite3 က command ဖိုင် ဖြစ်တာမလို့ Permission ပေးတဲ့အခါ 644 မရပါဘူး။ execute ပါလုပ်နိုင်အောင် 755 ဖြစ်ဖြစ် 4755 : 777 ဖြစ်ဖြစ် ပေးရပါမယ်။ ဘာကြောင့် အခုလိုပေးရတယ်ဆိုတာ ကတော့ ကျွန်ုတ် Chmod Command အသုံးပြုပုံမှာရှင်းပြထားပြီးဖြစ်ပါတယ်။ အခုဆို Google Play Store မှာ sqlite3 installer ဆိုတာလည်းရှိနေပါပြီ။ root access ရှိတယ်ဆို သူကိုအသုံးပြုနိုင်ပြီး sqlite 3 ကို မိမိ Device မှာ Install လုပ်နိုင်ပါတယ်။

Sqlite3 ဆိုတာကတော့ sql base command တစ်ခုပါ Windows နဲ့လည်း backwork compatable ဖြစ်ပါတယ်(အလုပ်လုပ်နိုင်ပါတယ်)sql လို့ပြောကတည်းက database ဆိုတာကိုတွဲမြင်ပြီးဖြစ်ပါလိမ့်မယ် ထင်ပါတယ်။ sqlite3 ဆိုတာကလည်း ကျွန်ုတ် Chmod Command ထဲမှာရှိတဲ့ database ဖိုင်တွေကိုပြင်ဖို့အတွက် အသုံးပြုတာပါ။ ဒီ Command အကြောင်းကိုဘာလို့ပြောရသလဲဆိုတော့ နောက်ပိုင်း ကိုယ့် Device ထဲကနေ password ဖျက်တဲ့အခါ service တွေကို enable၊ disable လုပ်တဲ့အခါ ပိုမိုလွယ်ကူပြီး ပိုအသုံးဝင်လို့ပါ။ နောက်ပြီး အရေးကြီးတဲ့ command တစ်ခုအကြောင်းကိုလည်း စာဖတ်သူတို့အတွက် ပဟ္မသုတဖြစ်အောင် ပြောပြချင်တာကြောင့် ပါ။ sqlite3 ဆိုတာ database ဖိုင်တွေကိုပြုပြင်ပေးနိုင်တာမလို /data/data အောက်မှာရှိတဲ့ package တွေရဲ့အောက်က database ဖိုင်တွေကိုပဲ ပြင်ပေးနိုင်ပါတယ်။ " data/ data/com.android.providers.settings/databases/settings.db " ဆိုပြီး ပြထားတဲ့ ဖိုင်လမ်းကြောင်းမှာဆုံး sqlite3 ပြင်ပေးမှာတ settings.db (\*.db) ဖိုင်ကိုပဲဖြစ်ပါတယ်။

လက်တွေအသုံးပြုပုံအနေနဲ့ ကိုယ့် Android ဖုန်းမှာ sqlite3 ကိုအသုံးပြုပြီး wifi ကိုလှမ်းဖွင့်တာ နှမူနာပြပေးပါမယ်။

```
>adb shell
#slite3 /data/data/com.adndroid.providers.settings/
databases/settings.db
#slite3> updatesecure set value=1 where name='wifi_on';
#sqlite3> .exit
#exit
```

Android မှာ am(activity manager) ကိုအသုံးပြုပြီး services (bluetooth, wifi) တွေကိုလှမ်းဖွင့်ရင် တကယ်ဖွင့်မှာလား၊ မဖွင့်ဘူးလားဆိုတဲ့ confirmation တောင်းပါတယ်။ အခုလို sqlite3 ကိုအသုံးပြုပြီး လုမ်းဖွင့်ရင် ဘာ confirmation မှ မတောင်းပါဘူး။ တန်းပြီးဖွင့်ပေးသွားတာပါ။ နောက်ဆုံးအခန်းရောက်ရင်လည်း sqlite3 ကိုအသုံးပြုပြီး screen lock ကို ကျော်တာကို ဆက်ရေးပြပေးသွားပါမယ်။

### (၂၄) **wipe / data | system | all**

wipe command ရဲနောက်မှာ (data/ system/ all) ဆိုပြီး ကိုယ့် wipe လုပ်ချင်တဲ့ position ကို ထည့်ပေးရပါတယ်။ wipe ဆိုတာကို မြင်တာနဲ့ recovery mode ထဲရောက်ရင် မြင်နေကျ "wipe cache partition" ဆိုတာကိုပဲ ပြီးမြင်မိမယ် ထင်ပါတယ်။ တကယ်တော့ ကျွန်တော်တို့ဟာ /data partition । /system partition တွေကို Terminal ထဲကနေလည်း တစ်ခါတည်းတန်းပြီး ဖျက်ချပစ်လို့ရပါတယ်။ wipe command အသုံးပြုဖို့ရင်လည်း root access ရှိဖို့လို့အပ်ပါတယ်။ ဒါနဲ့ သတိပေးလိုက်ပါဉိုးမယ်၊ ကိုယ့် device က ကောင်းကောင်းမွန်မွန် အလုပ်လုပ်နေတဲ့အခါန် "wipe system" နဲ့ "wipe all" ဆိုပြီး သွားမစမ်းလိုက်ပါနဲ့။ /system အောက်က အကုန်လုံးပျက်သွားပြီး terminal ထဲမှာအသုံးပြုနိုင်တဲ့ command တွေပါ အသုံးပြုလို့ မရနိုင်တော့ဘူး ဖြစ်သွားမှာပါ။ စမ်းသပ်ချင်ရင် "wipe data" ကိုပဲစမ်းပါ။ ဒါ Command က Terminal ထဲကနေ ကိုယ့် Android Device ကို Factory reset ချုတဲ့ Command ပဲ သတိထားပြီး အသုံးပြုစေချင်ပါတယ်။

### (၂၅) **exit**

exit ဆိုကတည်းက ထွက်တော့မယ်လို့ ဆိုလိုတာပါ။ terminal ထဲကနေ အသုံးပြုနေရင်း ပြန်ထွက်တော့မယ့်အခါ command prompt ထဲကနေအသုံးပြုနေရင်း ပြန်ထွက်တော့မယ့်အခါ အသုံးပြုပါတယ်။ မရေးပြုရင်လည်းသိကြမှာပါ။ သို့ပေမဲ့ မပြည့်စုံမှာစိုးလို့ ထည့်ရေးပေးလိုက်တာပါ။

ယခုရေးခဲ့တဲ့ Terminal ကနေအသုံးပြုတဲ့ command တွေအကြောင်းမှာဆိုရင် ကော်ပီကူးနိုင်တဲ့ "mv" နဲ့ "cp" မှာဆို အနည်းငယ်ကဲလွှာမှုရှိပါတယ်။ ဥပမာ sd card ထဲကနေဖုန်းရဲ့ system partition တစ်ခုခုအောက်ကို ကူးမယ်၊ ရွှေ့မယ်ဆို mv ကိုအသုံးပြုလို မရပါဘူး။ cp ကိုပဲ အသုံးပြုပေးရမှာပါ။ mv က မတူညီတဲ့ file system တွေကြား ကောင်းမွန်စွာ အလုပ်မလုပ်နိုင်ပါဘူး။ ဘာလိုလဲဆိုတော့ sdcard က FAT32၊ FAT ဖြစ်ပြီး system partition တွေကတော့ များသောအားဖြင့် ext2၊ ext3၊ ext4၊ yaffs2 တို့ဖြစ်နေလိုပါ။ နောက်တစ်ခုကတော့ ကိုယ့် android device က root ဖောက်ပြီးသား ဟုတ်လား မဟုတ်လား ဆိုတာကိုလည်း **id command** ရိုက်ပြီး ကြည့်လိုရပါတယ်။ root ဖောက်ပြီးသားဆို **uid** နောက်မှာ (root) လိုပြုပေးမယ်၊ မဖောက်ရသေးရင်တော့ (shell) ဆိုပြီးပြုပေးပါတယ်။ အခုအခန်းမှာ အသုံးပြုတဲ့ command တွေကို "adb shell <command>" ဆိုပြီး adb ကနေ တစ်ခါတည်းအသုံးပြုနိုင်ပါတယ်။ ခွဲပြီးရေးရတဲ့အကြောင်းကတော့ အရှေ့ကအခန်းက command ကိုပဲ အသုံးပြုရင်လည်း adb အတွက် ဗဟိုသုတ လုံလောက်ပြီးသားဖြစ်နေလိုပါ။

ဒီအခန်းမှာဆို "dd" command အသုံးပြုပဲ၊ "mount" command အသုံးပြုပဲနဲ့ "chmod" အသုံးပြုပဲတွေကို အကျယ်တဝါးရေးပြထားပါတယ်။ ဒါတွေကတကယ်ကို အရေးကြီးလိုပါ။ ကျွန်တော်နောက်အခန်းရောက်ရင် တကယ့် problem solving ပြုလုပ်ပုံတွေကို adb ထဲ terminal ထဲကနေပဲ ဖြေရှင်းပုံရေးပြုပေးမှာပါ။ ကိုယ့်ဘာသာ အလွတ် root ဖောက်ပဲ၊ pattern lock ဖျက်ပုံတွေကို ရေးပြပေးပါမယ်။ ဘာဖြစ်လိုလဲဆိုတော့ ကျွန်တော်တို့လေ့ကျင့်ခဲ့တဲ့ command တွေကို အပြင်မှာ လက်တွေဘယ်လိုအသုံးချတယ်ဆိုတာကို သိစေချင်ပါတယ်၊ ပိုပြီး နားလည်စေချင်ပါတယ်။ ပညာရပ်တစ်ခုကို အပေါ်ယုံကြည့်ရင် လွှာယ်တယ်လိုထင်တတ်ပြီး တကယ်အတွင်းကျကျ လေ့လာတဲ့အခါမှ ဘယ်လောက်ခက်ခဲတယ်၊ ဘယ်လောက်အသုံးဝင်တယ်ဆိုတာ တွေ့ရမှာပါ။

အခန်း ( ၆ )

## Fastboot Command အသုံးပြုပါ

*Fastboot Command* များအသုံးပြုပုံကတော့ Samsung Device တွေ အတွက် အသုံးမဝင်ပါဘူး၊ Huawei device၊ HTC device၊ Sony၊ Google Nexus နဲ့ China android phone တဲ့မှာပဲ အသုံးဝင်မှာပါ။ Fastboot ဆိုတာကို Google Nexus ဖုန်းတွေမှာ စတင်အသုံးပြုပြီး developer တွေအတွက် သိုးသန့်သဘော ဖန်တီးထားတာမလို သူကို developer mode လိုတောင် တင်စားခေါ်ကြပါတယ်။ Samsung မှာက download mode နဲ့ recovery mode ဆိုပြီး (j) ခုပဲ ပါပါတယ်။ Huawei device တွေမှာတော့ fastboot mode တစ်နည်းအားဖြင့် bootloader mode လိုလည်းခေါ်ပါတယ်။ နောက်ပြီး recovery mode၊ firmware တင်တဲ့အခါ ဝင်ရတဲ့ firmware installation mode (firmware update mode) ဆိုပြီး (r) ခုရှုပါတယ်။ HTC device တွေမှာလည်း Huawei device တွေမှာလိုပဲ fastboot mode၊ recovery mode ဆိုတာတွေရှိလို အခုအခန်းမှာရေးမယ့် Fastboot Command တွေကတော့ များသောအားဖြင့်အဲဒီ device တွေပေါ်မှာပဲ မူတည်၊ မြှို့ခိုပြီးရေးထား တာပါ။

Fastboot Command တွေအကြောင်း စမပြောခင် သိထားရမှာက Fastboot mode ဆိုတာကို ဘာကိုပြောတာလဲ၊ ဘာတွေလုပ်နိုင်လဲ၊ ဘာလို fastboot mode လိုခေါ်တာလဲ ဆိုတာတွေကို သိထားစေချင်ပါတယ်။ ဒါမှာလည်း ဒီစာတွေဖတ်ပြီးသွား တာနဲ့ မမေ့တော့မှာပါ။ Fastboot ဆိုတာကတော့ ကိုယ့် android device ရဲ့ partition တွေကိုပြန်ပြီး flash (အသစ်ရေး) လုပ်ဖို့အတွက် အသုံးပြုနိုင်ပါတယ်။ fastboot ဆိုတာ Toolkit တစ်ခုဖြစ်ပြီး (Adb ကဲသို့) Android SDK မှာ တစ်ပါတည်းပါဝင်လာပါတယ်။ Fastboot ထဲမှာဆိုရင် ကိုယ့် Android Device ရဲ့ File System images တွေကို ပြန်ပြင်နိုင်တယ်၊ ပြန်ဖျက်နိုင်ပါတယ်၊ ပြန်ရေးနိုင်ပါတယ်။ Fastboot ကို ကိုယ့်ဖုန်းက Android OS လိုခေါ်တဲ့ firmware ကို ကိုယ့်ဖုန်းပေါ်ဆွဲမတင်ခင် (OS loader အလုပ်မလုပ်ခင်) အခြေအနေမှာတင် အသုံးပြုနိုင်ပါတယ်။ fastboot ရှိတဲ့ device တွေမှာဆို firmware ပျက်နေရင်၊

မတင်ထားရင်တောင် fastboot ကိုဝင်လိုပါတယ်။

Fastboot mode ကနေ ကိုယ့် system ကို Update လုပ်တဲ့အခါ အရမ်းမြန်ပြီး recovery mode ထဲကိုမသွားဘဲနဲ့ကို လုပ်နိုင်ပါတယ်။ လူသုံးများ၊ လူပြောများတဲ့ CWM (Clockwork Mod) recovery ကိုလည်း fastboot ကနေ flash လိုပါတယ်။ ကိုယ် Flash လုပ်မယ့်ဖိုင်ကို Security အရ မှန်၏၊ မမှန်၏ မစစ်ဆေးဘဲနဲ့ တစ်ခါတည်းတန်းပြီး Flash လုပ်နိုင်ပါတယ်။ ဒါကြောင့် သတိထားရမှာက ကိုယ် Flash လုပ်တဲ့ဖိုင်မှားရင်၊ အဆင်မပြောင် ဖုန်းက Firmware တက်မှာ မဟုတ်တော့ပါဘူး။ ကျွန်ုတ်တော်တို့ recovery mode ကိုသွားပြီး "update from sdcard" ကနေ update လုပ်တဲ့အခါ ကိုယ် update လုပ်မယ့်ဖိုင်ဟာ system က သတ်မှတ်ထားတဲ့ သတ်မှတ်ချက်နဲ့မကိုက်ညီရင် စစ်ဆေး၊ Digitally sign လုပ်ကတည်းက error ပြုပြီး installation ကို ပပ်ဖျက်လိုက်ပါတယ်။ ဒါကြောင့် ကိုယ်က recovery mode ထဲကနေသာဝင်ပြီး update လုပ်ရင် အမှားနည်းနိုင်ပါတယ်။ အမှားကင်းတယ်လို့ မဆိုလိုပါ Fastboot ထဲကနေ partition တွေကိုဖျက်လိုပါတယ်။ အသစ်ပြန် ရေးလိုပါတယ်။ ဒါကြောင့် Fastboot ဟာဆိုရင် Software နဲ့ပတ်သက်တဲ့ ပြဿနာတွေဖြေရှင်းတဲ့အခါဖြစ်ဖြစ်၊ Hardware နဲ့ပတ်သက်တာတွေကိုဖြေရှင်းတဲ့ အခါဖြစ်ဖြစ် အသုံးဝင်တဲ့ Tool တစ်ခု ဖြစ်ပါတယ်။

Fastboot ကို များသောအားဖြင့် Huawei၊ HTC၊ Sony၊ Google Nexus ဖုန်းတွေမှာ bootloader unlock လုပ်တဲ့အခါတွေမှာ အသုံးများပါတယ်။ Fastboot ကိုဝင်မယ်ဆို adb command ကနေ "adb reboot-bootloader" ဒါမှုမဟုတ် "adb reboot bootloader" လိုခိုက်ပြီးဝင်နိုင်ပါတယ်။ ဒါကတော့ ကိုယ့် device က ပုံမှန်အတိုင်းအလုပ်လုပ်နေတဲ့ အခြေအနေမှာပါ firmware လည်းတက်တယ် ပုံမှန်ပဲ။ တကယ်လို့ ကိုယ့် device က firmware မတက်တော့တဲ့အခြေအနေမျိုးမှာ fastboot ထဲဝင်မယ်ဆိုရင်တော့ ကိုယ့်ဖုန်းအမျိုးအစားပေါ်မှုတည်ပြီး သူတို့သတ်မှတ်ပေးထားတဲ့ ckey Combination အတိုင်းဝင်ရမှာပါ။ များသောအားဖြင့် ဖုန်းတော်တော်များများက Vol down + Power တွဲနှိပ်ပြီး ဝင်ရတာပါ။

Fastboot ကိုအသုံးပြုဖို့အတွက်လည်း Adb လိုပဲ Install လုပ်ပေးဖို့လိုပါတယ်။ အဲဒါကြောင့် ကျွန်ုတ်က အစကတည်းက Adb install လုပ်ကတည်းက fastboot ကိုရော တစ်ပါတယ်။ ကော်ပိုကုံးပြီး ကွန်ပူးတာရဲ့ Windows directory ထဲကို

ထည့်ခိုင်းခဲ့တာပါ။ ကူးထည့်ထားခဲ့တယ်ဆိုရင်တော့ တစ်ခါတည်း Command Prompt ခေါ်ပြီး Fastboot လို့ တန်းရှိက်ပြီးအသုံးပြုနိုင်ပါပြီ။ တကယ်လို့ မကူးထည့်ခဲ့ရဘူးဆိုရင် လည်း အခုအချိန်မှာ ကူးထည့်လို့ရပါတယ်၊ မကူးထည့်ချင်ရင်လည်း fastboot.exe ရှိတဲ့ နေရာကိုသွားပြီး သူရှိတဲ့ directory ထဲမှာ shift + right click နိုင်၊ "open command window here" ကိုကလစ်ပြီး လည်းအသုံးပြုနိုင်ပါတယ်။ အခုပထမဆုံး fastboot command အနေနဲ့ "fastboot" လို့ရှိက်ကြည့်ပြီး fastboot မှာ ဘာတွေ အသုံးပြုနိုင်တယ်ဆိုတာကို အရင်ကြည့်ကြည့်လိုက်ပါ။

```
C:\Users\Phyo Ngeli>fastboot
usage: fastboot [ <option> ] <command>

commands:
update <filename>                                reflash device from update.zip
flashall                                         flash boot + recovery + system
flash <partition> [ <filename> ]                  write a file to a flash partition
erase <partition>                                 erase a flash partition
getvar <variable>                                display a bootloader variable
boot <kernel> [ <ramdisk> ]                      download and boot kernel
flash:raw boot <kernel> [ <ramdisk> ]            create bootimage and flash it
devices                                           list all connected devices
reboot                                         reboot device normally
reboot-bootloader                               reboot device into bootloader

options:
-w                                              erase userdata and cache
-s <serial number>                            specify device serial number
-p <product>                                  specify product name
-c <cmdline>                                  override kernel commandline
-i <vendor id>                                specify a custom USB vendor id
-b <base_addr>                                specify a custom kernel base address
-n <page size>                                specify the nand page size. default:
2048
```

Fastboot Command အသုံးပြုပုံကို သူရဲ့ usage မှာဖြပေးထားပါတယ်။ ပြန်ရေးပြုမယ် နည်းနည်းလေးပြောစရာရှိလာလိုပါ။

***fastboot [ <option> ] <comand>***

ဆိုပြီးသူက အသုံးပြုပုံကိုနူးပေးထားပါတယ် အားလုံးသိပြီးသား <--> နဲ့ပြထားရင် command မှာမဖြစ်မနေထည့်ပေးရပြီး [ -- ] ဆိုမထည့်ပေးလည်းရတယ်။

အပေါ်က command မှာဆို option ကိုသူက [-] ရော <--> နဲ့ပါပေးထားပါတယ်။ အခြားတော့ တချို့ command တွေမှာ မထည့်ပေးလည်းရတယ်၊ တချို့ command တွေမှာ option ထည့်ကို ထည့်ပေးရမယ်လို့ ဆိုလိုတာပါ။ သူမှာလည်း commands ရင်၊ options ရယ်ဆိုပြီး (j) ပိုင်း ပိုင်းပြထားပေးပါတယ်။ အရင်ဆုံး command တွေကို ရှင်းပြပေးပါမယ်၊ တစ်ဆက်တည်း option (switch) ဆိုပြီး အသုံးပြုပုံတွေကိုပါ ရောပြီး ရေးပြပေးသွားပါမယ်။

### (o) *fastboot devices*

ကွန်ပျူးတာမှာ ချိတ်ဆက်ထားတဲ့ Fastboot mode (bootloader mode) ဝင်ထားတဲ့ devices) တွေကိုကြည့်ဖို့အတွက် အသုံးပြုပါတယ်။ ဘာကြောင့်ကြည့်ဖို့လဲဆိုတော့ ကိုယ့် Device က ကွန်ပျူးတာမှာ ချိတ်ဆက်မိ၊ မမိရယ် နောက်ပြီးကိုယ်က ကွန်ပျူးတာမှာ Device ကိုတစ်ခုထက်ပိုပြီး ချိတ်ဆက်ထားတယ်ဆိုရင် ကိုယ်ကဘယ် device ကို flash လုပ်ချင်လဲဆိုတာ ကိုရွေးချယ်ပေးဖို့အတွက်ရယ် ကြည့်ရတာပါ။ Adb command အသုံးပြုပုံမှာဆိုရင်လည်း ကိုယ်က Adb device ကိုတစ်ခုထက်ပိုချိတ်ဆက်ထားရင် adb -s <ID> ဆိုပြီး device တစ်ခုတည်းကိုပဲ သီးသန့်ရွေးချယ် အလုပ်လုပ်နိုင်သလို fastboot မှာဆိုရင်လည်း fastboot -s <serial number> ဆိုပြီး device တစ်ခုတည်း ကိုပဲ သီးသန့်ရွေးချယ် အလုပ်လုပ်နိုင်ပါတယ်။

### (j) *fastboot update <filename>*

Update.zip ဖိုင်ကိုအသုံးပြုပြီး System ကို Flash လုပ်ချင်တဲ့အခါ အသုံးပြုပါတယ်။ <filename> နေရာမှာကိုယ့်ရဲ့ update.zip ဖိုင်ရှုတဲ့လမ်းကြောင်းကိုရေးပေးပါ။ mouse ကိုအသုံးပြုပြီး Drag and drop လုပ်ရင်လည်းရပါတယ်။ ရောသွားအုံမယ် fastboot မှာအသုံးနိုင်တဲ့ update.zip နဲ့ recovery mode မှာ အသုံးပြုနိုင်တဲ့ update.zip ဖိုင်က မတူပါဘူး။ တော်ကြာ စမ်းလို့မရတာနဲ့ မရဘူးထင်နေမှာစိုးလိုပါ။ Fastboot ကနေ update လုပ်နိုင်တဲ့ update.zip နဲ့ recovery mode ကနေ update လုပ်နိုင်တဲ့ update.zip နဲ့က မတူပါဘူး။

## (၃) *flash <partition> / <filename>*

Recovery partition၊ System partition၊ Data partition တွက် USB ကြိုးထိုးပြီး flash လုပ်တဲ့အခါအသုံးပြုပါတယ်။ ပုံမှန်အတိုင်း Android Device တစ်ခုကို terminal ထဲကနေ root directory ( / ) အောက်ကို ls -l နဲ့ကြည့်ရင် System၊ Data ဆိုတာတွေက Directory (ဖိုလ်ဒါ) အနေနဲ့ တွေ့နေရတာပါ။ ဘာဖြစ်လို့ partition လိုပြောလဲ မေးစရာရှိလာပါလိမ့်မယ်။ ဥပမာဏျာ အရှေ့က အခန်းတွေမှာလည်း ပြောပြခဲ့ဖူးပါတယ်။ Android ရဲ့ System ဟာ Data တွေအများအပြား သိမ်းနိုင်ဖို့ အတွက် Partition တွေအများတွေ့ခွဲထားရတယ် /system ဆိုတာကတည်း mtdblock0 ရဲ့ partition အပိုင်းတစ်ပိုင်းဖြစ်ပြီး mount လုပ်တဲ့ Command လေ့ကျင့်ခန်းမှာဆို သူရဲ့ block နံပါတ်က mtdblock4 တစ်နည်းအားဖြင့် partition နံပါတ် ၄ ဖြစ်ပြီး ကျွန်ုတ်တို့နားလည် လွယ်အောင် /system ဆိုပြီးနာမည်နဲ့ Shortcut သဘေးပေးထားတာပါ။ ပြောရရင်တော့ partition 4 အောက်မှာဆိုရင် system နဲ့ပတ်သက်တွေ့ သိမ်းထားလို့ /system ဆိုပြီးနာမည်ပေးထားတာပါ။ Flash command ကို အသုံးပြုပြီး Fastboot ကနေ partition တွက် flash လုပ်ပုံတွက် အောက်မှာ ကြည့်လိုက်ပါ။

```
fastboot flash userdata data.img
fastboot flash system system.img
fastboot flash boot boot.img
fastboot flash recovery recovery.img
fastboot flash bootloader bootloader.img
fastboot flash splash splash.img
```

အပေါ်ဆုံးက command ကတော့ Userdata ဖြစ်တဲ့ Data partition ကို Flash လုပ်တာပါ။ ဒုတိယတစ်ခုက system partition ကို flash လုပ်တယ် တတိယတစ်ခုက boot partition ကို flash လုပ်တယ် နောက်ပြီး အတွေ့ရအများဆုံး ကတော့ လေးခုမြောက်ဖြစ်တဲ့ recovery partition ကို flash လုပ်တာပါပဲ။

ကျွန်ုတ်တို့ဟာ Custom recovery (CWM) ကိုတင်ချင်တဲ့အခါ ကိုယ့် Device မှာ Recovery ပျက်သွားလို့ stock recovery ပြန်တင်ချင်တဲ့အခါ အခုလို

Command ကနေ flash လုပ်ပေးရပါမယ်။ နောက်တစ်ခုကတော့ bootloader ကို flash လုပ်တာပါ။ နောက်ဆုံးတစ်ခုကတော့ splash image ကို flash လုပ်တာပါ။ စာဖတ်သူတွေအနေနဲ့ /system၊ /data partition တွေအကြောင်းကောင်းကောင်းသိနေပေး၊ bootloader အကြောင်းတော့ အတွေ့နည်းမှာမိုလို ဒီနေရာမှာ ကျွန်တော် Bootloader အကြောင်းနည်းတော့ ပိုရေးပြပါမယ်။

Fastboot Mode ကို တစ်နည်းအားဖြင့် Bootloader Mode လို့လည်းခေါ်ပါတယ်။ ဒါကတော့ သိပြီးသားဖြစ်မှာပါကျွန်တော်တို့ Android device တွေမှာ တစ်ခုနဲ့တစ်ခု bootloader အသုံးပြုပုံချင်းမတူပါဘူး။ bootloader ဆိုတာ ဖုန်းတွေမှာ ပထမဦးဆုံး အလုပ်လုပ်ရတဲ့ program တစ်ခုဖြစ်ပြီးသူကို processor (cpu)၊ board တွေပေါ်မှုတည်ပြီး သီးသန့်ဖန်တီးပေးရတာပါ။ bootloader တွေအမျိုးမျိုးရှုပါတယ်။ android device ထုတ်လုပ်သူတွေဟာ redboot၊ uboot၊ qi bootloader တွေကို အသုံးများပါတယ်။ အခုနောက်ပိုင်းတော့ ကိုယ်ပိုင် Bootloader တွေကို ဖန်တီးပြီး အသုံးပြုနေကြပါပြီ။ ဒါကြောင့်လည်း Huawei၊ HTC၊ Sony၊ Google Nexus တို့ဟာ Bootloader ကို lock လုပ်ထားကြပြီး Developer mode လို့ခေါ်တဲ့ Fastboot Mode ကနေပြီး ဝင်ရောက် အသစ်ရေးလို့မရအောင် ပိတ်ထားကြပါတယ်။ ကျွန်တော်တို့အနေနဲ့ လည်း device တစ်ခုနဲ့တစ်ခု unlock လုပ်ရတာချင်းမတူကြပါဘူး။ အဲဒါကတော့ သူတို့တွေရဲ့ ကိုယ်ပိုင် bootloader တွေကြောင့်ပါပဲ။

ပို့ဆောင်သွားပြီထင်တယ်။ Redboot၊ uboot၊ qi bootloader တွေအကြောင်း ပြောလိုက်လိုပါ။ အဲဒါမှာ Redboot ဆိုတာ eCos Operating System ပေါ်ကိုအခြားထားတာပါ။ သူက linux အခြားမဟုတ်ပါဘူး။ ဒါပေမဲ့ မည်သည့် OS Platform ကိုမဆို အထောက်အပံ့ပေးနိုင်ပါတယ်။ uboot ဆိုတာကတော့ Universal Bootloader ဖြစ်ပြီး သူကတော့ linux OS ကိုအခြားပါတယ်။ qi bootloader ကတော့ uboot ကိုပို့ကောင်းလာအောင် ပြင်ဆင်ထားတဲ့ bootloader ဖြစ်ပါတယ်။ ဘယ်လိုပြင်ထားလဲဆိုတော့ uboot ဆိုရင် သူရဲ့ image ဖိုင်ကို system memory ပေါ်ကနေပဲယူပြီးအလုပ်လုပ်နိုင်ပါတယ်။ qi bootloader ကတော့ sdcard ထဲကနေပါယူသုံးနိုင်ပါတယ်။ ဘယ် Bootloader ပဲဖြစ်ဖြစ် သူလုပ်ရမယ့်အလုပ်တွေကို အဆင့် ၂ ဆင့်နဲ့ လုပ်ဆောင်ရပါတယ်။ ပထမအဆင့်မှာ External RAM ကိုရှာပြီး Program တွေ တင်ပေးရပါတယ်။ ဒုတိယအဆင့်အနေနဲ့ကတော့ kernel ကောင်းမွန်စွာအလုပ်

လုပ်နိုင်ဖို့အတွက်လိုအပ်တဲ့ memory၊ network စတာတွေကို စီစဉ်ပေးရပါတယ်။ bootloader ရဲ့ အဓိကလုပ်ဆောင်ရတဲ့ တာဝန်ကတော့ kernel ကောင်းမွန်စွာ အလုပ် စတင်နိုင်စေဖို့ပဲ ဖြစ်ပါတယ်။

Android Device တွေမှာ Bootloader တွေ မတူညီတာကြောင့် ကိုယ်က bootloader ကိုသာ ပြန်ပြီး flash ချင်ရင် တစ်ခါတစ်ရုံမှာ ကိုယ့် device ရဲ့ bootloader ပေါ်မှုတည်ပြီး flash လုပ်ပေးတဲ့ command တွေ ကွဲသွားနိုင်ပါတယ်။ ဥပမာ - "*fastboot flash hboot hboot.img*" ဆိုတဲ့ပုံစံမျိုးပါ။ အဲဒီမှာဆိုရင် သူရဲ့ bootloader က hboot ဖို့လို့ hboot.img ဖိုင်ကိုသုံးပြီး flash လုပ်ပေးရတာပါ။ HTC ဖုန်းတွေမှာ hboot ကိုအသုံးပြုကြပြီး အကယ်၍ HTC wildfire A Series တွေမှာဆို bootloader unlock လုပ်ဖို့အတွက် hboot ဗားရှင်းကို မဖြစ်မနေမြင်ပေးရပါမယ်။ partition တွေကို အခုလို flash တဲ့အခါ ကွန်ပျူတာ screen မှာ ပက်ဆောင်ရွက်လေးလာပြပေးမှာပါ။ ကိုယ်က အဲဒီမက်ဆောင်ကိုကြည့်ပြီး flash လုပ်တာ အောင်မြင်လား၊ မအောင်မြင်ဘူးလား ဆိုတာ ဆုံးဖြတ်ရတာပါ။ ဥပမာ recovery ကို flash မယ်ဆို . . .

```
sending 'recovery' (4930 KB)...
OKAY [ 0.521s]
writing 'recovery'...
OKAY [ 0.489s]
finished. total time: 1.10s
```

ပထမဆုံး Recovery ဖိုင်ကို Device ထဲ လှမ်းပိုလိုက်တယ်၊ ပြီးတော့ ရေးတယ် (flash လုပ်တယ်)၊ အောင်မြင်တော့ OKAY လိုပြပြီး ကြာတဲ့အချိန်ပါပြပေး ပါတယ်။

### (၅) *fastboot erase <partition>*

ဒီတစ်ခါတော့ System Partition တွေကို fastboot ထဲကနေလှမ်းပြီးဖျက် တာကိုပြောပါမယ်။ ဖျက်မယ်ဆို သူရဲ့ command က "fastboot erase <partition>" ဆိုတော့ ကျွန်တော်တို့ <partition> နေရာမှာ ကိုယ်ဖျက်ချင်တဲ့ Partition

နာမည်ကိုရေးပေးရမှာပါ။ ကိုယ်က system partition ကိုဖျက်ချင်ရင် <partition> နေရာမှာ system ဆိုတာနဲ့ အစားထိုးပေးရမှာဖြစ်ပြီး၊ ကိုယ်က data partition ကိုဖျက်ချင်ရင်တော့ data ဆိုတာနဲ့ အစားထိုးရေးပေးရမှာပါ။

*fastboot erase system  
fastboot erase boot  
fastboot erase recovery  
fastboot erase userdata  
fastboot erase cache*

အပေါ်က Command တွေက နမူနာပြုပေးထားတာပါ။ အားလုံးတစ်ပြိုင် နက်သုံးရမှာမဟုတ်ပါဘူး။ ကိုယ်ဖျက်ချင်တဲ့ partition အတွက် ကိုယ်အသုံးပြုသင့်တဲ့ command ကိုပဲအသုံးပြုပေးရမှာပါ။ erase အကြောင်းပြောရင်းနဲ့ တစ်လက်စ တည်း "fastboot -w" ဆိုတဲ့ [-w] switch ခံပြီး ရိုက်တာလေးလည်း ပြောပြလိုက်ပါမယ်။ ကျွန်ုတ်တို့ fastboot ထဲကနေ data | cache တွေကို command တစ်ခုချင်းစိုကာနေ ဖျက်လို့လည်းရသလို "fastboot -w" ဆိုပြီး တစ်ကြောင်းတည်းရိုက်လိုက်ရင်လည်း Android device ဟာ /data အပိုင်းကို ဖျက်ပြီးသား /cache ကိုလည်း ဖျက်ပြီးသား ဖြစ်သွားပါတယ်။ တစ်နည်းအားဖြင့် factory reset ချပေးတဲ့ command လိုပဲ မှတ်ထားစေချင်ပါတယ်။ အတွေ့အကြုံတစ်ခုလောက် ပြောပြချင်ပါတယ်။ နည်းနည်းတော့ သည်းခံပါ။ ဘာလို့လဲဆိုတော့ နည်းနည်းဆို များတော့မှာမြိုလိုပါခင်ဗျာ။

China android (clone) တစ်လုံးက fi:firmware မတက်ဘူး၊ လိုဂို ကနေ တက်လိုက် ကျလိုက် တက်လိုက်ကျလိုက်ဖြစ်နေလို့ recovery mode ထဲဝင်ပြီး factory reset ချဖြော်ပြီးစားပေမဲ့ recovery mode ဝင်လို့မရဘူးဖြစ်နေတယ်။ fastboot mode ထဲလည်းဝင်ဖို့ ကြိုးစားကြည်လည်းမရဘူး။ firmware ကို ပြန်တင်ကြည့်ဖို့ ROM တစ်ခုခု ပြန်တင်ကြည့်ဖိုးစားပေမဲ့ အွန်လိုင်းမှာ ရွှေမရနဲ့ စိတ်ပေါက်ပေါက်နဲ့ ပစ်ထားလိုက်တာ ၂ ရက်၊ ၃ ရက်လောက်ကြောသွားတယ်။ နောက်နေ့ မနက်ကျမှ ဖြစ်ချင်တာဖြစ် စမ်းတာပေါ့ဆိုပြီး Vol(-) ကို မလွှတ်ဘဲ နှိပ်ပြီး ကွန်ပူ၍တာမှာ ကြိုးထိုးလိုက်တာ သူက တစ်ခါတည်း fastboot mode ထဲ ရောက်သွားရော့ဖျား။ အဲဒါနဲ့

ကျွန်တော်လည်း "fastboot -w" ရှိက်ပြီးတော့ ဖုန်းကို reboot ချုပြန်တက်လာတော့ ကောင်းသွားရော့ဖွား။ ပြောပြချင်တာက "fastboot -w" ဆိုတာ ရှိက်ရတာသာစာလုံးနည်းတာ အသုံးတကယ်ဝင်တယ်ဆိုတာ ပြောပြချင်တာပါ။

ကျွန်တော်အခုပြောတဲ့ "fastboot -w" ဆိုတာ "fastboot erase cache" ရယ် "fastboot erase data" ဆိုတဲ့ Command ၂ ခု ရှိက်တာနဲ့ အတူတူပါပဲ၊ တကယ်လို့ ကျွန်တော်တို့က /cache၊ /data၊ /system ၏ ခုစွမ်းကို ဖျက်ချင်ရင်တော့

*fastboot erase system -w*

လို့ ရှိက်လိုက်ရုပါပဲ။ သူက

*fastboot erase system  
fastboot erase data  
fastboot erase cache*

ဆိုတဲ့ command (၃) ခုနဲ့ အတူတူပါပဲ။ ဘာလို့ ခွဲပြောရလဲဆိုတော့ တစ်ခါတစ်လေ ကိုယ့် Device က Cache တွေများနေလို့ စက်သုံးရတာဘာ နှေးနေတယ်ဆိုရင်၊ အင်တာနက်အသုံးများလို့ cache တွေများလို့ နှေးသွားတယ်ဆိုရင် "fastboot erase cache" ဆိုတဲ့ command အသုံးပြပြီး /cache partition ကို ရှင်းပေးလို့ရပါတယ်။

### (၅) *fastboot oem <command>*

fastboot oem လို့ ခေါင်းစဉ်တပ်ပြီးရေးပြထားတာနဲ့ တစ်ခါတည်း မှန်းလိုက်ပြီးထင်ပါတယ် bootloader unlock လုပ်တော့မယ်ဆိုပြီး။ ဟုတ်ပါတယ် ကျွန်တော်တို့ဟာ "fastboot oem" ကို Huawei၊ HTC၊ Sony၊ Nexus ဖုန်းတွေကို bootloader unlock လုပ်ဖို့အတွက်သာ အသုံးများခဲ့လိုပါ။ တကယ်တော့ *oem* ကိုအသုံးပြပြီး တစ်ခြား လုပ်ဆောင်နိုင်တာတွေ အများကြီးရှိပါသေးတယ်။ တတ်နိုင်

သလောက် ရေးပြေပေးသွားပါမယ်။ oem ဆိုတာကတော့ "Original Equipment Manufacturer" ရဲ့အတိုကောက်ပဲဖြစ်ပါတယ်။ အရင်ဆုံး oem ကိုအသုံးပြုပြီး Huawei ဖုန်းတွေမှာ bootloader unlock လုပ်တာ ရေးပြေပေးပါမယ်။ အခုနောက်ပိုင်းတော့ Vroot၊ Eroot တွေက Huawei ဖုန်းတွေကို root ဖောက်တဲ့အခါ bootloader ကို unlock လုပ်ပေးစရာမလိုတော့ပေမယ့် အရင်ကဆို သူရဲ့ bootloader ကို unlock လုပ်ပေးပြီးမှာသာ CWM တင်ပြီး root ဖောက်လိုရတာပါ။ HTC၊ Nexus ဖုန်းတွေမှာတော့ bootloader unlock လုပ်ပေးပြီးမှာသာ root ဖောက်လိုရမှာပါ။ HTC မှာက bootloader unlock လုပ်ပြီးမှ S-ON ကနေ S-OFF လုပ်လိုရတာပါ။ unlock လုပ်မယ်ဆို အရေးကြီးဆုံးက မလုပ်ခင် ဖုန်းထဲမှာရှိတဲ့ data အားလုံးကို backup ခွဲထားဖိုပါပဲ။ မဟုတ်ရင် အကုန်ပျက်ကုန်ပါလိမ့်မယ်(အရေးကြီးပါတယ်)။ အောင် ဒါနဲ့ S-ON၊ S-OFF ဆိုတာ security on၊ security off ပါ။

*fastboot oem unlock <unlock code>*

unlock code ဆိုတဲ့နေရာမှာ Huawei ကနေ ပြန်ပိုပေးတဲ့ ကုဒ် (၁၆) လုံးကို ရှိက်ထဲပေးရမှာပါ။ မှန်ရင် OKAY လိုပြုပြီး ကိုယ့်ဖုန်းထဲမှာရှိတဲ့ user data တွေကိုပါ တစ်ခါတည်း ဖျက်ပေးသွားမှာပါ။ တကယ်လို ကိုယ်က unlock ဖြစ်နေတဲ့ bootloader ကို lock ပြန်ချချင်တယ်ဆိုရင်တော့ unlock command ရဲ့ ပြောင်းပြန် "lock" command ကိုအသုံးပြုရမှာပါ။

*fastboot oem lock*

ဒါနဲ့ ပြောစရာကျွန်းနေတာတာက Huawei ကပြန်ပိုပေးတဲ့ ကုဒ် (၁၆)လုံးဆိုတာ Huawei က သူဘာသာသူ လာပိုပေးတာမဟုတ်ပါဘူး။ သူရဲ့ Unlock Code တောင်းတဲ့ webpage ကနေ ကိုယ့်ဖုန်းရဲ့ Model No၊ IMEI or MEID၊ Product name နဲ့ Email Account တွေဖြည့်ပြီး Huawei company ကိုပိုပေးတော့မှာ Huawei ကနေ ကိုယ့်ထဲပေးလိုက်တဲ့ အီးမေးလ်ထဲကို unlock code ပြန်ပိုပေးတာပါ။ သူပြန်ပိုပေးတဲ့ ကုဒ်နံပါတ်တွေနဲ့ ကိုယ်က unlock လုပ်ပေးရမှာပါ။

Sony ဖုန်းတွေကို bootloader unlock လုပ်ပုံဟာလည်း Huawei နဲ့ တူပေါ့ အသုံးပြုရတဲ့ command ကတော့ အနည်းငယ် ကွဲလွှဲပါတယ်။ ကျွန်တော်က fastboot အသုံးပြုပုံဆိုပြီး ဘာဖြစ်လို့ bootloader unlock လုပ်တာတွေပြောနေလဲဆိုရင် ကျွန်တော်တို့ အသုံးပြုနေကျ မြင်နေကျ command တွေရဲ့ အသေးစိတ်ကို ရှင်းရှင်း လင်းလင်းသိခေါ်ချင်လိုပါ။ အရင်ဆုံး ဖုန်းကို fastboot mode ဝင်ထားပါမယ်။ များသော အားဖြင့် Vol (-) ကို ဖိတ်ပြုး ကွန်ပူးတာမှာကြိုးထိုးပေးလိုက်ရင် မီးအစိမ်းလေးလင်း နေပြီး ဖုန်းက fastboot mode ထဲ ရောက်သွားပါတယ်။ အသုံးပြုရမယ့် command ကတော့

```
fastboot -i 0x0fce oem unlock 0xKEY
```

ဆိုပြီးရိုက်ပေးရမှာပါ၊ **KEY** နေရာမှာတော့ Sony ကနေ အီးမေးလ်ထဲပြန်ပိုပေးတဲ့ key တွေကို ထည့်ပေးရမှာပါ။ Huawei မှာလိုပါပဲ Sony မှာလည်း unlock key ကို သူရဲ့ website ဖြစ်တဲ့ <http://unlockbootloader.sonymobile.com> မှာသွားပြီး ကိုယ့် ဖုန်းရဲ့ IMEI or MEID । ကိုယ့်နာမည်၊ အီးမေးလ်လိပ်စာတွေပေးပြီး key ကိုတောင်းယူနိုင်ပါတယ်။ ဆက်စပ်မိလို့ ပြောပြတာပါ။ တကယ်ပြောချင်တာက sony ကို unlock လုပ်တဲ့ command မှာဆို fastboot နောက်မှာ [ -i ] switch ခံပြီး သူနောက်မှာ " vendor id " ထည့်ပေးရပါတယ်။ 0x0fce ( zero x zero fce ) ဆိုတာက Sony ရဲ့ vendor id ပါ vendor id မလို့ Sony အားလုံးအတူတူပါပဲ တကယ်လို့ Device id သာဆို မတူနိုင်ပါဘူး။ ကျွန်တော်က fastboot <command> တွေနဲ့ fastboot [< option >] တွေကို တွဲပြီးရေးပြပေးတာက တစ်ခုချင်းစီ ခွဲပြန့် ခက်လိုပါ။ HTC ဖုန်းတွေ bootloader unlock လုပ်ပုံကိုတော့မရေးပြတော့ပါဘူး။ အခုံလောက်ဆို " fastboot oem unlock " အသုံးပြုပုံကို သိလောက်ပြီးထင်ပါတယ် တစ်ခြား **oem** နဲ့ တွဲသုံးနိုင်တဲ့ command တွေလည်းရှိပါတယ် အသုံးနည်းတာကြောင့် အကျယ်မရှင်းပြတော့ပါဘူး။ " fastboot oem " နောက်မှာ ဘယ်လို့ command တွေဆက်သုံးလို့ရလ ဆိုတာကို " fastboot oem h " command ကိုအသုံးပြုပြီး ကြည့်လို့ရပါတယ်။

```
C:\Users\Phyo Ngel>fastboot oem h
... INFO command list
INFOkeytest
INFOheap
INFOboot
INFOreset
INFOpowerdown
INFOrebootRUU
INFOheap_test
INFOtask
INFOtask
INFOenableqxdm
INFOgencheckpt
OKAY [ 0.026s ]
finished. total time: 0.026s
```

အပေါ်မှာပြထားတဲ့ပုံကိုကြည့်ရင် INFO နောက်က စာသားတွေက အသုံးပြုခိုင်တဲ့ Command တွေပြောပြီးပါတယ်။

### (၆) *fastboot reboot or fastboot continue*

Command ၂ ခု ကို ခေါင်းစဉ်တစ်ခုတည်းအောက်မှာ ရေးပြထားပါတယ်။ ထထမ command ကတော့ fastboot mode ထဲရောက်နေလဲ device ကို reboot ချမယ်ဆို "fastboot reboot" ဆိုပြီး ဖုန်းကို Reboot ချမယ်၊ ပြီးရင်ပုံမှန်အတိုင်း ပြန်တက်လာအောင် လုပ်ပေးနိုင်ပါတယ်။ ကိုယ်က recovery partition + system partition တွေကို flash လုပ်ပြီးတဲ့အခါ reboot မဖြစ်မနေ ချေပေးရပါမယ်။ ဒါဆို "fastboot contine" ကို ဘယ်နေရာမှာသုံးမလဲဆိုတာ ခန့်မှန်းလို့ရလောက်ပါပြီ။

တကယ်လို့သာ ကျွန်တော်တို့က fastboot mode ထဲဝင်ပြီး ဘယ် image ဖိုင်ကိုမှ flash မလုပ်ဘူးဆိုရင်တော့ ဖုန်းကိုပုံမှန်အတိုင်း ပြန်တက်လာအောင် reboot command ကိုအသုံးပြုစရာမလိုပါဘူး "fastboot contine" ဆိုပြီး လက်ရှိရောက်နေလဲ fastboot mode ထဲကနေတစ်ခါတည်း တန်းပြီး ပုံမှန်အတိုင်း ပြန်တက်သွားမှာပါ။ (ဒါ process Boot remming လို့ ခေါ်ပါတယ်)

```
C:\Users\Phyo Ngei>fastboot continue
      resuming boot... OKAY [ 0.002s ]
finished. total time: 0.003s
```

reboot ကျပြီးမှ ပုံမှန်အတိုင်းပြန်တက်လာမယ့် ကြောချိန်နဲ့ continue ကနေတက်လာမယ့် ကြောချိန်ကတော့ စလာနှင့်ပိုင်းလေးပဲကွာတာပါ။ မိမိနှစ်သက်ရာ အသုံးပြုနိုင်ပါတယ်။ ကျွန်တော်ကတော့ အသုံးပြုနိုင်တဲ့ command တွေပဲ ပြောပြုတာပါ။

## (၃) *fastboot reboot-bootloader*

မပြောလည်း သိပါပြီ။ အခုရောက်နေတဲ့ fastboot mode ထဲကနေ ဖုန်းကို reboot ကျပြီး နောက်တစ်ကြိမ် Fastboot Mode (Boot Loader Mode) ထဲ ပြန်ဝင် စေချင်တဲ့အခါ အသုံးပြုပါတယ်။

Fastboot Command အသုံးပြုပုံကိုတော့ ဒီလောက်ဆို လုံလောက်ပြီလို ထင်ပါတယ်ခင်ဗျာ။ ကျမ်းကျင် အဆင့်ရောက်အောင်တော့ မိမိဘာသာ အားထုတ်ပြီး ကြိုးစားကြည့်ပါ။ Fastboot command တွေမှာ သတိထားရမှာက unlock မလုပ်ခင် backup မဖြစ်မနေဆဲပါ။ စမ်းသပ်တဲ့အခါ fastboot mode က ဖုန်းတိုင်းမှာ မပါဘူး၊ Huawei၊ HTC နဲ့ Sony ဖုန်းတွေမှာသာ ပါလာတာပါ။ အခုဆက်ပြီး android ဖုန်းနဲ့ပတ်သက်ပြီး တွေ့ရတတ်တဲ့ အခက်အခဲတွေကို adb command အသုံးပြုပြီးဖြေရှင်းပုံကို နောက်အခန်းမှာ ဆက်လေ့လာကြည့်ပါ၌ဦးခင်ဗျာ။

အခန်း ( ၃ )

## Adb command ဖြင့် လက်တွေ့ပြသနာဖြေရှင်းနည်း

ဒီအခန်းမှာတော့ adb command ကနေ လက်ရှိအတွေ့များတဲ့ အခက်အခဲတွေကို command အသုံးပြုပြီး ဖြေရှင်းပုံတွေကို ရေးပြပေးမှာပါ။ သိပ်တွေတွေထူးထူးတော့ မဟုတ်ပါဘူး။ Android device တွေ pattern lock၊ password lock တွေကျော်တာ၊ Email account တွေဖျက်တာ၊ root ဖောက်ပုံ ဥပမာနဲ့ တခြား ရောက်တတ်ရာရာလေးတွေပါ။ အခုံလိုပဲ ပြသနာတိုင်းပြသနာတိုင်းမှာ ဖြေရှင်းလိုရမယ့်နည်းလမ်းတော့ရှိပါတယ်။ မတွေ့သေးလို့ ခက်ခဲနေတာပါ။ ကျွန်တော်ကလည်းတစ်ခုခုဆို အသေးစိတ် အပိုတွေပါ လိုက်ရှင်းပြနေတာကလည်း ကျွန်တော်ဖတ်ဖူးတဲ့ အဂ်လိပ်စကားပုံ " A little knowledge is too much dangerous " ဆိုလိုပါစမ်းသပ်ပါ အားပေးပါတယ် မစမ်းသပ်ခင်လေ့လာပါ အကျိုးချည်းသက်သက် သိရှုမလေ့လာဘဲ အပြုစိန္တပါသိအောင် လေ့လာပါ။ အဲဒီလိုမှုလည်း အပ်နဲ့ထွင်းရမယ့် အနာပုံဆိန်နဲ့မထွင်းရမှာပါ။ တစ်ခုခုမလုပ်ခင်အမြဲတစ်း backup ဆွဲထားသင့်ရင် ကြိုး ဆွဲထားပါ။ ကျွန်တော်ထဲ့ IT သမားတွေအတွက် data ဆိုတာ အသက်တစ်မျှ ပါပဲ။

### (၁) pattern / password / pin lock problems

Method (1)

- (၁) root access ရှိရန် လိုအပ်ပါတယ်။
- (၂) adb driver သိနေဖို့လိုပါတယ်။ မသိရင် adb driver ပြန်တင်ပေးလိုက်ပါ။
- (၃) command prompt ကိုဖွင့်ပြီး အောက်က command တွေကိုတစ်ကြောင်းစီ ရှိက်ပေးပါ။

*adb remount*

*adb shell rm /data/system/password.key*

*adb shell rm /data/system/gesture.key*

*adb reboot*

တကယ်လိုသာ adb remount လုပ်လိုမရရင်တော့ adb ကနေတန်းဖျက်လို မရပါဘူး။ Terminal ထဲကိုသွားပြီးမှ ဖျက်ပေးလိုရမှာပါ။

```
adb shell
$su
#rm /data/system/password.key
#rm /data/system/password.ke
#reboot
```

reboot ကျပြီးပြန်တက်လာရင် pattern၊ password၊ pin တွေ ပျက်နေပါပြီ။ လိုသလိုသုံးလိုရနေပါပြီ။

Method (2)

Pattern lock pass by using sqlites.

### (J) *Google Account Remove*

ဒီတစ်ခါ ကိုယ့်ဖုန်းထဲမှာ ဝင်နေတဲ့ တွေားသူရဲ့ အီးမေးလုံအကောက်တွေကို ဖျက်ကြည့်ပါမယ်။ အရင်ကဆို ကိုယ့်ဖုန်းထဲက သူများအီးမေးလုံအကောက် ဖျက်စိုး အတွက် ဖုန်းကို Factory reset ချတဲ့နည်းလမ်းကိုပဲ အသုံးပြုကြတာပါ။ နောက်မှ စမ်းရင်း စမ်းရင်း factory reset ချစရာမလိုဘဲ ဖြေရှင်းနိုင်သွားကြတာပါ။

- (o) **root access** ရှုဖို့လိုပါတယ်။
- (J) GUI ကဆို root explorer တင်ပါ။ Adb ကနေဖျက်မယ်ဆို Command Prompt ကိုဖွင့်ပါ။

```
adb remount
adb shell rm /data/system/account.db
adb shell rm /data/system/sync/accounts.xml
adb reboot
```

password ဖျက်သလိုပါပဲ "adb remount" နဲ့ အဆင်မပြုရင် Terminal ထဲသွားပြီး ဖျက်ပေးလိုက်ပါ။

## (၃) China Tablet (clone) stop at Android logo

ဒီတစ်ခါရေးပြုမှာကတော့ dalvik-cache တွေကိုဖျက်ပေးခြင်းအားဖြင့် တစ်ခါတစ်ရုံ Android Device တွေဟာ factory reset ခုစရာမလိုဘဲ start-up အခြေအနေမှာ ရပ်နေတာတွေကို ဖြေရှင်းပေးနိုင်တယ်ဆိုတာပဲ ဖြစ်ပါတယ်။ ကိုယ့်ရဲ့ Android Device ဟာ လိုဂိုကနေ ရှူးဆက်မတက်နိုင်ဘူးဆို ဖြစ်နိုင်ချေကတော့ device ရဲ့ start-up ပရိုဂရမ်တွေမှာ ပြဿနာရှိနေလို့ ဖြစ်နိုင်ပါတယ်။ အဲဒါဆိုသူရဲ့ dalvik-cache တွေကို ဖျက်ပေးလိုက်ခြင်းအားဖြင့် ဖြေရှင်းပေးနိုင်ပါတယ်။ ပထမဆုံး ကိုယ့် Android Device ကို ကွန်ပျိုးတာနဲ့ချိတ်လိုက်ပါ။ Adb Driver သိနေပြီးသား ဖြစ်နေရပါမယ်ခင်ဗျာ။

```
adb remount
adb rm /data/dalvik-cache/*
adb reboot
```

ပြန်တက်လာရင် လိုဂိုမှာ ရပ်မနေတော့ဘဲ system ဟာ ချောချောမောမော အလုပ် လုပ်နေတာ တွေ့ရမှာပါ။ ဒီနေရာမှာ ပြောစရာရှိလာနိုင်တာ ကျွန်တော် ဖျက်လိုက်တာ cache တွေပါ။ ဒါဆို /cache ကို erase လုပ်လိုက်ရင်ပြီးတာပဲလိုပြော

နေပါတီးမယ်။ တချို့ tablet တွေမှာ fastboot mode မပါပါဘူး။ နောက်ပြီး terminal၊ adb ကနေ " wipe " command ကိုအသုံးပြုမယ်ဆို " cache data " ၊ " wipe system " ၊ " wipe all " ဆိုတာပဲအသုံးပြုနိုင်ပါတယ်။ " wipe cache " လိုအသုံးမပြုနိုင်လိုပါ။

## (၄) Viber Activation Problem

*Viber Activation Problem* ဆိုလို viber activate လုပ်တာများကွာလွယ်လွယ်လေးကို ရေးသီးမယ်လို့ မထင်စေချင်ပါဘူး။ ကျွန်တော်အခုရေးပြုမှာက customer က နိုင်ငံခြားမှာ viber ကိုအသုံးပြုခဲ့တယ်၊ နိုင်ငံခြားက ဖုန်းနံပါတ်နဲ့ Activate ယူခဲ့တယ်၊ ဒီရောက်တော့လည်းဆက်သုံးနေတယ်။ အခု ဖုန်းက error တွေတက်နေပြီး firmware ပြန်တင်ရတော့မယ် အခြေအနေ။ သူက နိုင်ငံခြားကဖုန်းနဲ့ activate ယူထားတာဖြစ်လို့ ဒီနိုင်ငံမှာ နိုင်ငံခြားနံပါတ်နဲ့ ပြန်ပြီး activate ပြန်လုပ်လို့မရနိုင်လို့ သူ viber ကိုမပျက်စေချင်ဘူး။ viber ထဲက call logs၊ messages၊ contacts တွေကိုလည်း ဒီအတိုင်းရှုံးစေချင်တယ်ဆို ခေါင်းနည်းနည်း ရှုပ်သွားမလား။ ဟုတ်ကဲပါ။ အားလုံးကို backup ဆွဲထားလို့ရပါတယ်။ Firmware ပြန်တင်ပြီးရင် backup ဆွဲထားတာကို restore ပြန်လုပ်ပေးလိုက်ရုံပေါ့။

```
adb pull /data/data/com.viber.voip D:\com.viber.voip
```

ဆိုပြီး ကိုယ့် ကွန်ပျူးတာရဲ့ D:\ အောက်ကို ကော်ပီကူးယူထားပြီး Firmware ပြန်တင်ပြီးတဲ့အခါ

```
adb push d:\com.viber.voip /data/data/com.viber.voip
```

ဆိုပြီး restore ပြန်လုပ်ပေးနိုင်ပါတယ်။ backup ဆွဲတဲ့အခါ /sdcard ထဲကိုလည်း

ဆွဲလိုရပါတယ်။ ကိုယ် Install လုပ်ထားတဲ့ application တွေရဲ့ data ဖိုင်တွေအကုန်လုံး ဟာ /data/data အောက်မှာရှိပါတယ်။ တခြား data တွေကိုလည်း မိမစိတ်ကြိုက် backup ဆွဲထားလို့ရပါတယ်။ အပေါ်က Viber ကတော့ ကိုယ်တွေ့မြှုလို ရေးပေးလိုက်တာပါ။

## (၂) Gain Root Access

ဒီတစ်ခါခေါင်းစဉ်ကတော့ ပြဿနာဖြေရှင်းပုံ မဟုတ်ပါဘူး။ ကျွန်တော်အရှေ့မှာ ရေးပြခဲ့သမျှ command တွေကိုပဲအသုံးပြုပြီး root access ရယူပုံကို ရေးပြခဲ့တာပါ။ ဘာရယ်မဟုတ်ပါဘူး။ root ဖောက်တဲ့ အခြေခံသဘောတရားရယ်၊ root ဖောက်တဲ့အခါ လုပ်ဆောင်ပုံ အဆင့်ဆင့်ရယ်ကို သိစေချင်ပါတယ်။ နောက်ပြီး ကျွန်တော်ရေးပြတဲ့ command တွေကိုကြည့်ပြီး adb command အသုံးဝင်ပုံကိုလည်း စိတ်ဝင်စားစေချင်လိုပါ။

Android version 2.1 - 2.2 root

```
adb devices
adb push psneuter /data/local/tmp/psneuter
adb shell
$ cd /data/local/tmp
$./psneuter
```

psneuter ကို run လိုက်တဲ့အခါန်မှာ shell ကအလိုလိုပိတ်သွားပါလိမ့်မယ်။ ခကေလေးစောင့်ပြီး နောက်တစ်ကြိမ် command prompt မှာ adb service ကိုပြန်ပြီး restart လုပ်ပေးဖို့လိုပါတယ်။

```

adb kill-server
adb devices
adb remount
adb push busybox /system/bin/busybox
adb push su /system/bin/su
adb push Superuser.apk /system/app/Superuser.apk
adb shell chmod 4755 /system/bin/busybox
adb shell chmod 4755 /system/bin/su
adb shell chmod 644 /system/app/Superuser.apk
adb reboot

```

အခုဆို ဖုန်းပြန်တက်လာရင် Superuser icon နဲ့ Root access ရနေပြီဆိုတာ တွေ့ရမှာပါ။ ဒါကတော့ SuperOneClick ဆောင်နဲ့ Root access ရယူပုံကို အသေးစိတ် မြင်သာအောင် ပြန်ရေးပြတာပါ။ ကျွန်ုတ်တို့ မသိတဲ့ command တွေမပါပါဘူး။ မသိဘူးလို့ ပြောလို့ရတာဆိုရင် " ./sneuter " ပရှိမယ်ထင်ပါတယ်။ psneuter ကိုအသုံးပြုပြီး Android System ကို exploit လုပ်တာပါ။ Android မှာ program တွေ run ရင် ./<filename> ဆိုပြီးပဲ run ပေးရပါတယ်။ exploit လုပ်တဲ့ ဖိုင်ပကဲတာပါ။ root access ယူပုံကတော့ အားလုံးနီးပါး အတူတူပါပဲ။ အားလုံးနီးပါး လို့ပြောရတာ အကြောင်းရှိပါတယ်။ exploit လုပ်တဲ့ ဖိုင်မလိုဘဲနဲ့ လွယ်လွယ်ကူကူ root ဖောက်လိုရတဲ့ device တွေလည်း ရှိပါတယ်။ ဥပမာ - Huawei T8830၊ Wellcom Mobile ver 2.3 တို့ပါ။ ဘယ် android device မဆို /data/local/tmp ဖို့လိုအပါဘာ root access မရခိုင်ဖြစ်ဖြစ်၊ ရှိပြီးရင်ဖြစ်ဖြစ် system ကိုခွင့်တောင်းစရာမလိုဘဲ ဖိုင်လှမ်းထည့်လို့ ရပါတယ်။

အောက်မှာ Huawei T8830 ကို root ဖောက်ပုံကို ရေးပြပေးထားပါတယ်။ အမိကရည်ရွယ်ချက်ကတော့ adb command တွေကို ဘယ်လိုအသုံးပြုသွားတယ် ဆိုတာကို သိစေချင်လိုပါ။

```

adb shell mv /data/local/tmp /data/local/tmp.bak
adb shell ln -s /data /data/local/tmp
adb reboot
adb shell rm /data/local.prop > nul
adb shell "echo \"ro.kernel.qemu=1\" > /data/local.prop"
adb reboot
adb shell id

```

ID ကိုကြည့်လို ၀ (Zero) ဆိုလျှင် ဆက်ပြီး အောက်က command တွေဆက်ရှိကြသွားပါ။  
 ၀ (Zero) မဟုတ်ရင်တော့ root ဖောက်တာ မအောင်မြင်ပါဘူး။ ရှုံးက command  
 တွေ ဆက်ရှိက်ဖို့မလိုတော့ပါဘူး။

```

adb remount
adb push su /system/bin/su
adb shell chown 0.0 /system/bin/su
adb shell chmod 06755 /system/bin/su
adb push busybox /system/bin/busybox
adb shell chown 0.0 /system/bin/busybox
adb shell chmod 0755 /system/bin/busybox

```

```

adb push Superuser.apk /system/app/Superuser.apk
adb shell chown 0.0 /system/app/Superuser.apk

adb shell chmod 0644 /system/app/Superuser.apk
adb shell rm /data/local.prop
adb shell rm /data/local/tmp
adb shell mv /data/local/tmp.bak /data/local/tmp
adb reboot

```

အပေါ်က command တွေမှာ ကျွန်တော်တို့မသိတဲ့ command တွေမပါ  
 ပါဘူး။အားလုံးကို နိုးရှင်းစွာအသုံးပြုပြုသွားတာပါ။ လေ့လာမယ်ဆို အများကြီးရှိပါတယ်။

ဥပမာ - DoomLord Root bin4ry root တို့ဆိုရင်လည်း လွယ်လွယ်ကူကူ root ဖောက်ပြထားတာပါ။ ကိုယ်တိုင်လေ့လာကြည့်ဖို့ တိုက်တွန်းပါတယ်။ အနည်းဆုံးတော့ ကျွန်တော်တို့ဟာ Android Device တွေကို root ဖောက်တဲ့အခါ အွန်လိုင်းမှာတက်ရှာစရာ မလိုဘဲ ကိုယ့်မှာရှုပြီးသားနည်းလမ်းတွေကို စပေါင်းပြီး စမ်းသပ်လို့ရတာပေါ့။

အတွေ့အထူးမဟုတ်တော့တဲ့ "adb push" command အကြောင်း တစ်ခုလောက်ပြောပြချင်ပါတယ်။ ကျွန်တော်အပေါ်ကရေးခဲ့သမျှ push လုပ်ပုံလုပ်နည်းတွေက "adb push <filename> <destination/filename>" ပဲဖြစ်ပါတယ်။ ကိုယ်က Superuser.apk ဖိုင်ကို အခြားနေရာကိုကူးယူတဲ့အခါ နာမည်ပြောင်းချင်ရင် ပြောင်းနိုင်အောင် နောက်ဆုံးမှာ filename ပြန်ရေးပေးထားတာပါ။ တကယ်လို့ ကျွန်တော်တို့ Superuser.apk ကိုသူနာမည်အတိုင်းပဲ အခြားနေရာကိုကူးယူမယ်ဆိုရင်တော့ filename ပြန်ရေးစရာမလိုဘဲ dot(.) ပဲရေးပေးရင်လည်းရပါတယ်။ ဥပမာ - "adb push Superuser.apk /system/app/." ပါ။ "Gain Root Access" ကတော့ ဒီလောက်ပဲရေးပြုပါတော့မယ်။

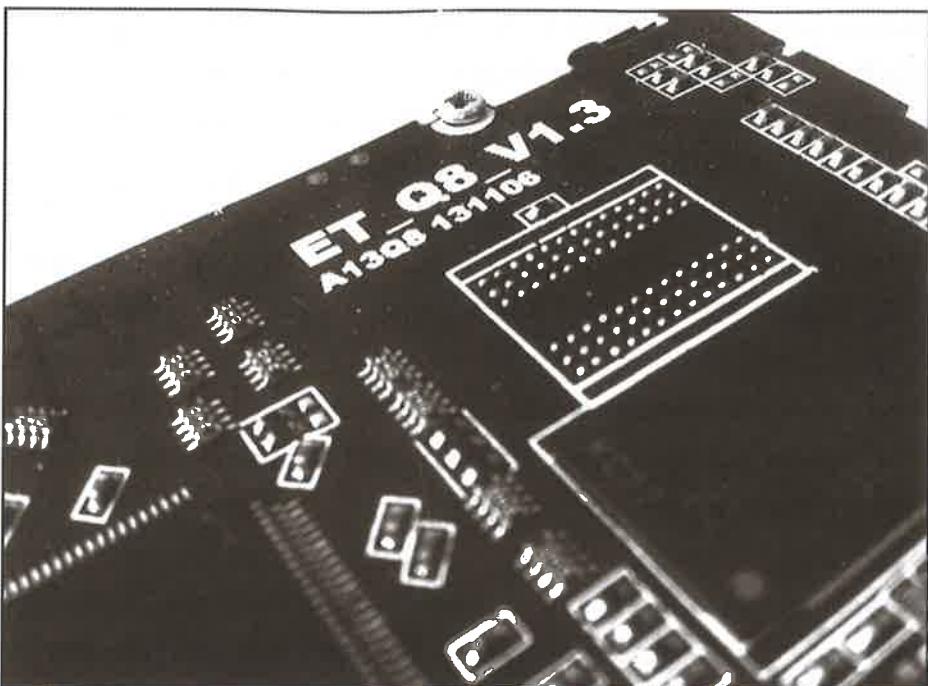
## (၆) China Tablet မှာ: Firmware ရေးနည်း

**ADB** နဲ့တော့ ဒီခေါင်းစဉ်က မဆိုင်ပါဘူး။ ဒါပေမဲ့ ကျွန်တော်စာအုပ်ဖတ်ရင်းနဲ့ Tablet တွေ ကိုဘယ်လို့ Firmware ရေးလည်းဆိုတာလေးပါတယ်ခါတည်းသိ သွားစေချင်လိုပါ။ ဈေးကွက်မှာရှိသမျှ Tablet တွေအားလုံးအတွက် Firmware ရေးနည်းကို သိုးသန္ဓာဇ်ပုံပါ။ ထုတ်ဖို့ရည်ရွယ်ပေမဲ့ ဒီစာအုပ်ဖတ်တဲ့သူ တွေအတွက် အကျိုးရှိအောင် တစ်ပါတယ်း ရေးပေးလိုက်တာဖြစ်လို့ အသုံးဝင်လိမ့်မယ် လို့ မျှော်လင့်ပါတယ်ခင်ဗျာ။ Mobile Service ဆိုင်တော်တော်များများကလည်း China Tablet ဆော့ဝဲကျတယ်ဆို ပြန်ရေးပေးဖို့ ခဲယဉ်းနေကြတာတွေ့နေလိုပါ။ ရတယ်ဆိုရင်လည်း Service Charges ကို အဆမတန် တောင်းတာမျိုးလည်း ဖြော်ဖြော်ပါတယ်။ ပြီးတော့ စမ်းသပ်ချင်တဲ့သူတွေအတွက်လည်း စမ်းတဝါးဝါး ဖြစ်မနေစေချင်တာကြောင့် ဒီခေါင်းစဉ်ကို Adb နဲ့ မဆိုင်ပေမဲ့ ရေးဖြစ်အောင် ရေးပေးလိုက်တာပါ။ အခုံဆိုရင် ဈေးကွက်မှာလည်း China Tablet တွေက တော်တော်များများ

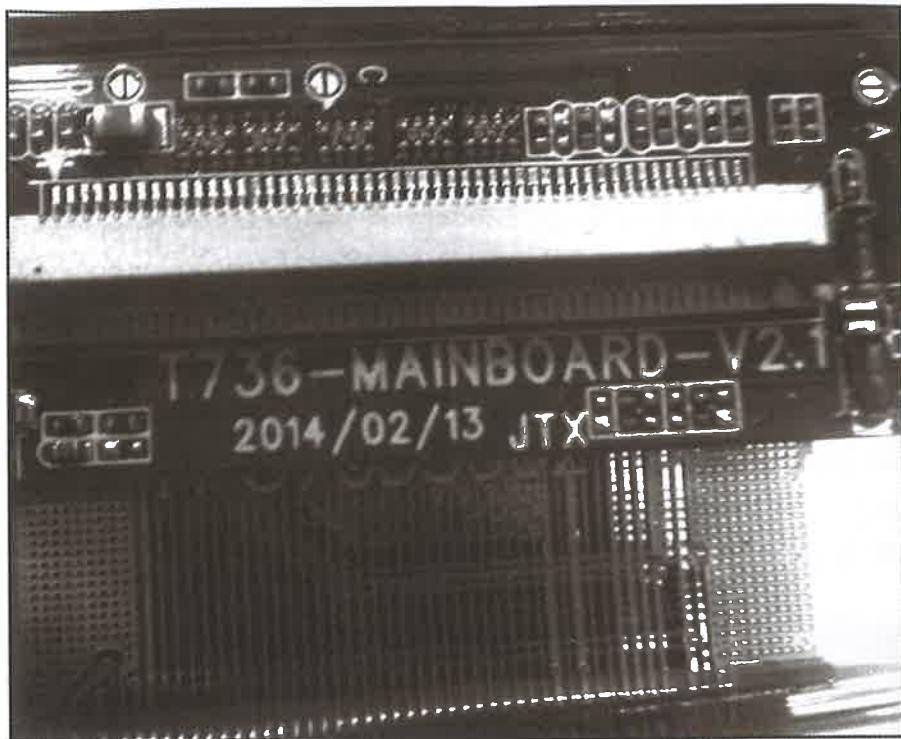
နေရာယူလာတာ တွေ့ရပါတယ်။ အဲဒီအတွက်လည်း Firmware ပြဿနာကို ကြိုးကြိုးမှားမား စိတ်ထဲမထားဘဲ လွယ်လွယ်ကူကူ ပိမိဘာသာ လေ့လာစမ်းသပ်စေခဲ့ပါတယ်။ ပုံမှန်အားဖြင့် အတွေ့များတဲ့ Tablet အမျိုးအစားတွေကတော့ အခုံ လက်ရှိ လူသုံးများတဲ့ MOMO၊ ONDA၊ China Samsung Galaxy တို့ဖြစ်ပြီး ကျွန်တော်တို့ မသိသေးတဲ့ နာမည်တွေနဲ့ Tablet မော်ဒယ်တွေလည်း အများကြိုးရှိနိုင်ပါတယ်။ Tablet တွေမှာ မော်ဒယ်တွေ အမျိုးအစားတွေ ကဲ့လွှဲနေသလို သူတို့၏ CPU အမျိုးအစား ဖြစ်တဲ့ A10၊ A13၊ A23 ဆိုပြီးကဲ့နေပေမဲ့ ရေးပုံရေးနည်းကတော့ အားလုံးနီးပါး တူညီကြပါတယ်။

Tablet Firmware ရေးတော့မယ်ဆိုအရင်ဆုံးအခက်အခဲက ဒီ Tablet ကို ဘယ် Firmware နဲ့ရေးရမှာလဲ ဘယ်လို့ရေးရမှာလဲဆိုတာပါပဲ။ MOMO၊ ONDA၊ Huawei၊ Samsung အစရိုသဖြင့်နာမည်ရှိပြီးသား Tablet တွေအတွက် အခက်အခဲ မရှိနိုင်ပါဘူး။ Firmware ကလည်းသူ့မော်ဒယ်နဲ့ သူရှာရတာလွယ်ပါတယ်။ ရေးပုံ ရေးနည်းကလည်းသူ့ Software နဲ့သူဖြစ်လို့ရေးရတာအခက်အခဲမရှိနိုင်ပါဘူး။ တကယ်လို့သာ အောက်ကပုံမှာပြထားသလို Tablet မျိုး China ကလာတဲ့ Samsung တံဆိပ်တပ်ထားတဲ့ Clone တွေဆိုနည်းနည်းစဉ်းစားရတော့မှာပါ။ Firmware ကို ဘယ် Firmware နဲ့ ရေးပေးရမှာလဲ Firmware ဘယ်ကနေရမလဲ ဘယ်လို့ ရေးပေးရမလဲဆိုတာမျိုးပေါ့။ ဟုတ်ကဲ့ပါ။ အရမ်းကြိုးမခက်ပါဘူးလို့ ကြိုးပြောထားပါရစေ။

ပထမဆုံး နောက်ဖုံးကာဗာကိုဖွင့်ပြီး CPU ရဲ့မော်ဒယ်ကိုကြည့်မယ်။ Mainboard ရဲ့ မော်ဒယ်ကိုကြည့်ပါမယ်။ CPU က A1s3၊ Mainboard မော်ဒယ် က Q8\_V 1.3 ဆိုပြီးတော့တွေပါမယ်။ အဲဒါက ကျွန်တော်တို့ ရှာပြီးရေးပေးရမယ့် Firware နာမည်ပါ။



Tablet တွေရဲ့ Mainboard မော်ဒယ်ကို နေရာအမျိုးမျိုးမှာ တွေ့နိုင်ပါတယ်။ တစ်ခါတလေ Board ရဲမြင်သာတဲ့နေရာမှာရှိပါတယ်။ တစ်ခါတလေ LCD ကြီးရဲ့ အောက်မှာရှိတတ်ပြီး ကြိုးဖြေတ်ပြီးကြည့်ရပါတယ်။ တချို့ Tablet တွေမှာ မော်ဒယ် ရှာမတွေ့တာမျိုးလည်း ဖြစ်တတ်ပါသေးတယ်၊ ရှိပါတယ်။ သူတိုက Mainboard ရဲ့ အောက်ဘက်မှာရေးထားလိုပါ။ မှတ်မှတ်ရရရ ဂ လက်မ China Sasung Galaxy Tab ဆိုတဲ့ Clone တွေမှာပါ။ အဲဒါကြောင့် တစ်ခါတလေ အပေါ်ကာဗာဖွင့်ကြည့်ရုံနဲ့ မမြင်ရတဲ့အခါ ဘုတ်မှာတပ်ထားတဲ့ Screw လေးတွေကို သေသေချာချာဖြေတ်ပြီး အောက်ဘက်မှာ လျှန်ကြည့်ပါ။ အောက်ကပုံမှာတော့ A23 CPU မှာဆို သူမော်ဒယ်ကို LCD ကြီးရဲ့ အောက်မှာရေးထားတာပါ။ များသောအားဖြင့် Tablet တော်တော် များများမှာတော့ သူတို့ရဲ့မော်ဒယ်ကို မြင်သာတဲ့ Mainboard ပေါ်မှာပဲ ရေးပေးထားလေ့ ရှိပါတယ်။



ဒါဆို မော်ဒယ်သိပြီမို့ ကျွန်တော်တို့လိုချင်တဲ့ Firmware တွေကို အင်တာနက် ပေါ်ကနေ ဒေါင်းလုပ်ဆွဲပြီး ရေးလို့ရနိုင်ပြီပေါ့။ Google မှာရှာမယ်၊ 4shared မှာ ရှာ့ရမယ်။ များသောအားဖြင့် Tablet Firmware တော်တော်များများဟာ [www.4shared.com](http://www.4shared.com)၊ [www.q88.com](http://www.q88.com) မှာရှိပါတယ်။ ကိုယ်က T730 Firmware လိုချင်တယ်၊ အပေါ်မှာပြခဲ့တဲ့ T736-Mainboard-V2.1 ကိုရှာချင်တယ်ဆို ပထမဆုံး [www.4shared.com](http://www.4shared.com) ကိုသွားမယ်၊ အပေါ်က Search box မှာ လိုချင်တဲ့ မော်ဒယ်ရှိက်ထည့်မယ်။ T736 Mainboard လို ရှိက်ထည့်လိုက်တာနဲ့ result မှာ သူရှာလိုတွေတာတွေ လာပြပါလိမ့်မယ်။ ကိုယ်လိုချင်တဲ့ V2.1 ကိုရွေးပြီး ဒေါင်းလုပ် လုပ်လိုက်ပါ။ Firmware လည်းရပြီဆိုရင်တော့ ရေးဖို့ပဲကျွန်ပါတော့တယ်။ ဘယ်လိုရေးမလဲ၊ ဟုတ်ကဲ သိပ် မခက်ပါဘူး။ A10၊ A13၊ A23 CPU နဲ့အလုပ်လုပ်တဲ့ Tablet တွေကို Firmware ရေးဖို့အတွက် မိမိကျွန်ပူးတာမှာ LiveSuit|PhoenixSuit|AML\_FlashTool ဆိုတဲ့ Firmware ရေးတဲ့ Tool တစ်ခုခုရှိဖို့လိုပါတယ်။ မရှိသေးရင် အင်တာနက်ကပဲ

ဒေါင်းလုပ်ဆွဲလိုက်ပါ။ တစ်ခါတလေ မဟုတ်ပါဘူး၊ များသောအားဖြင့် \*.rar | \*.zip နဲ့ ဒေါင်းလုပ်ဆွဲယူရတဲ့ Firmware ဖိုင်တွေမှာ Upgrade Tool တစ်ခုခု (ဥပမာ - LiveSuit or PhoenixSuit) ပါပြီးသားဖြစ်ပါတယ်။ တစ်ချို့ ONDA Tablet တွေကိုတော့ AML\_FlashTool ကိုအသုံးပြုပြီးရေးပေါ်ပါတယ်။ DVD အခွဲထဲ မှာအားလုံးထည့်ပေးလိုက်ပါတယ်။ ဘာပဲပြောပြာ မရှိသေးရင်ဒေါင်းလုပ်ဆွဲပေါ့။ တကယ်လို့ T736 ကိုရေးတဲ့အခါ V2.1 နဲ့ ရေးလို့မရရင် သူထက် ဘားရှင်းမြင့်တဲ့ ဖိုင်နဲ့ ပြောင်းရောကြည့်ပါ (e.g V2.3)။

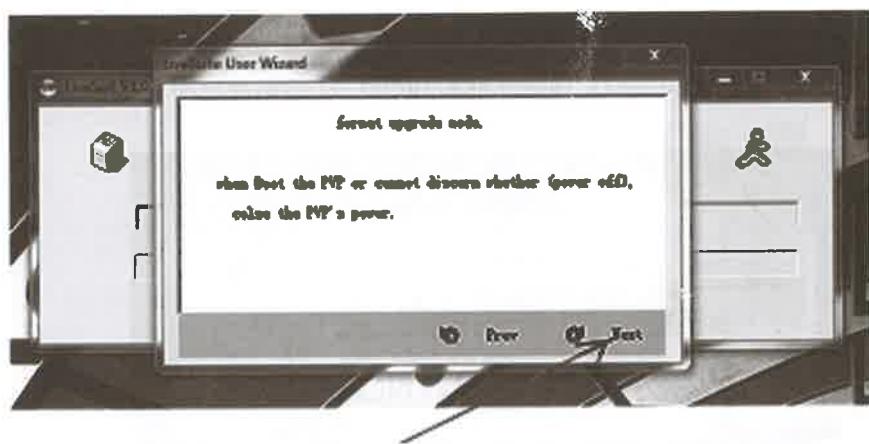
အခုခို အားလုံးပြည့်စုံနေပြုဖြစ်လို့ စတင်ပြီး Firmware ရေးလို့ရပါပြီ။ အရင်ဆုံး Livesuit ကိုအသုံးပြုပြီးရေးပြုပေးပါမယ်။ နှစ်ခုစလုံးကတော့ အတူတူ ပါပဲ။ ပိုလွှယ်မယ်ထင်တာ၊ ကိုယ်နဲ့ပိုအဆင်ပြေတာနဲ့အသုံးချ နိုင်ပါတယ်။ ဒေါင်းလုပ် ဆွဲယူထားတဲ့ LiveSuit ဖိုလ်ဒါတဲက Livesuit.exe ဖိုင်ကို ဖွင့်လိုက်ပါ။ ပထမဆုံး "Welcome box" ကျလာရင် "Yes" ကိုရွေးပါ။



"Please select upgrade mode" ဆိုတဲ့ မက်ဆောင်ဘောက်ကျလာရင် "Format" ကိုရွေးပေးပါ။



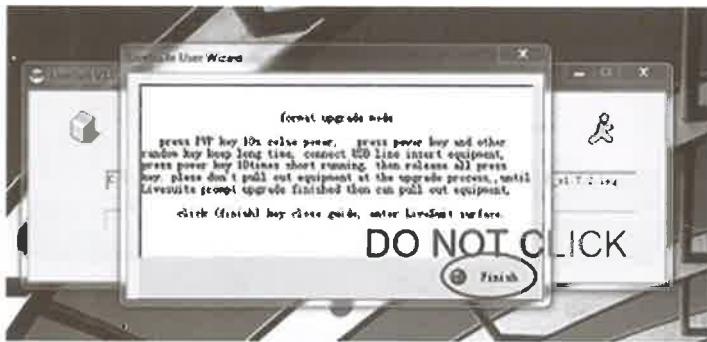
"Format upgrade mode" ဘောက်ကျလာရင် "Next" ကိုပဲ ရွှေးပေးပါ။



နောက်တစ်ဆင့်က ကိုယ်တင်မယ့် Firmware ကိုရွှေးခိုင်းတာဖြစ်လို့ "Yes" ကိုနှိပ်ပြီး ကိုယ်ရေးမယ့် \*.img ဖိုင်(တစ်နည်း - ဒေါင်းလုပ်ဆွဲထာတဲ့ဖိုင်) ကိုရွှေးပေးပါ။



နောက်တစ်ဆင့်က Firmware ရေးမယ့် Tablet ကို ကွန်ပျူတာနဲ့ ချိတ်ဆက် ခိုင်းတဲ့ အပိုင်းပါ။ "Finish" ဆိုတာကို မနှစ်ပါနဲ့မြို့။ Tablet ကိုပါဝါပိတ်ပါ။ Vol + ဒါမဗဟိုတ် Vol - တစ်ခုခုကို ဖိနှိပ်ထားပြီး Tablet ကို ကွန်ပျူတာမှာ ချိတ်ဆက်ပါ။



**STOP - It is time to plug your device into your PC.**

ကွန်ပျူတာနဲ့ ချိတ်ဆက်ပြီးတာနဲ့ ဖိနှိပ်ထားတဲ့ Vol ကိုကိုမလွတ်ဘဲ အမြား တစ်ဖက်နဲ့ Power ဆလှတ်ကို ၁၀ ချက်လောက် ဆက်တိုက်နှိပ်ပေးထားလိုက်ရင် Tablet ဟာသူအလိုလို Upgrade mode ထဲရောက်သွားပြီး Firmware ရေးနှိုင်တဲ့ အနေအထားကို ရောက်သွားပါလိမ့်မယ်။

သတိထားစရာ အနေနဲ့ကတော့ LiveSuit နဲ့ရေးတဲ့အခါပထဆုံး Tablet အတွက် Driver တောင်းပါလိမ့်မယ်။ Livesuit က Portable ဖြစ်တာကြောင့် Driver တောင်းခဲ့လျှင်ကိုယ်တိုင်ပြန်တင်ပေးရပါမယ်။ ကွန်ပျူးတာရဲ့ Device Manager ထဲသွားပြီး Driver မသိသေးတဲ့ Device (အဝါရောင် Exclamation Sign (!) နဲ့ပြနေတဲ့ Device) ပေါ်ကို Right click နှိပ်ပြီး "Update Driver Software" ကိုရွေးပေးပါ ပြီးရင် "Browse my computer for driver software" ကိုနှိပ်ပြီး ဒေါင်းလုပ်ဆွဲထားတဲ့ Upgrade tool(LiveSuit) ဖို့လ်ဒါကို ညွှန်းပေးလိုက်ရင် သူဘာသာ ကွန်ပျူးတာကနေ Driver ကိုတင်သွားပါလိမ့်မယ်။

Driver တင်ထားပြီးသား ကွန်ပျူးတာမှာဆိုအရင်ဆုံး ကွန်ပျူးတာ Taskbar မှာ "Installing driver" ဆိုပြီး မက်ဆွဲချုပြပေးပါလိမ့်မယ်၊ ဘာမှာမလုပ်သေးဘဲ စောင့်နေလိုက်ပါ။



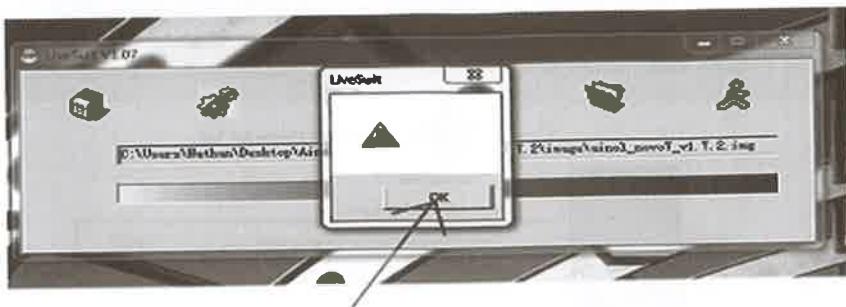
Driver သိသွားပြီဆို ပထမဆုံး "Mandatory Format" ဆိုတဲ့ ဘောက်လေး ကျလာပါလိမ့်မယ်။ ဒီနေရာမှာသတိထားရမှာပါ၊ သူက "Yes or No" နှစ်ခုမေးပါတယ်၊ Yes ကိုရွေးမယ်ဆို ကိုယ်ရေးမယ့် Tablet ထဲက အရင်ဒေတာတွေကို အကုန်ဖျက်ပြီး တင်ပေးသွားမှာဖြစ်ပြီး၊ No ဆိုရင်တော့ အရင်ဒေတာအဟောင်းတွေကို မဖျက်ဘဲ Upgrade အနေနဲ့သာ တင်ပေးသွားမှာပါ။



အကောင်းဆုံးအကြံ့ဗြိုက်ကတော့ No နဲ့တင်ပါ၊ အရင်အဟောင်းတွေလုံးဝမယ်ပါဘူး။ ဒါပေမဲ့ တစ်ခါတစ်ရု Upgrade လုပ်လိုအဆင်မပြေတာမျိုးတွေရှိတတ်ပါတယ်။ ဥပမာ MOMO P702 Tablet ဆိုပါတော့၊ ကိုယ်က သူရဲ့ Original Firmware ကို တင်လိုအဆင်မပြေတဲ့အခါ Tablet ကို ကာာာဖွင့်ပြီး ဘုတ်ရဲ့မော်ဒယ်ကို ကြည့်ရေးရတဲ့အခါမျိုးနဲ့ကြုံရင် Original မဟုတ်တာကြောင့် Upgrade လုပ်ရင်ဘယ်လိုမှ အဆင်မပြေနိုင်ပါဘူး။ အဲလိုအချင်မှု Yes ကိုပဲမဖြစ်မနေရွေးပေးပြီး အရင်အဟောင်းတွေကို ဖျက် Firmware ပြန်ရေးပေးရပါမယ်။ No ဆိုရင်တော့ နောက်ထပ် မက်ဆွဲချုံကောက် ကျ မလာတော့ဘဲ Firmware တန်းပြီးရေးပေးသွားမှာပါ။ Yes ဆိုရင်နောက်တစ်ဆင့် အနေနဲ့ တကယ်ဖျက်မှာလားဆိုတဲ့ မက်ဆွဲချုံ ပြပေးပါလိမ့်မယ်။ Yes ကိုပဲရွေးပေးလိုက်ပါ။ LiveSuit ကနေ Tablet ကို Firmware ရေးနေတာကို အစိမ်းရောင် ဘားတန်းလေးနဲ့ ပြပေးပါလိမ့်မယ်။



ရေးနေတဲ့ အစိမ်းတန်းဆုံးသွားရင် Ok ဆိုတဲ့ မက်ဆွဲချုံကောက်လေးကျလာပါလိမ့်မယ်။ Firmware ရေးတာအောင်မြင်ပါတယ်ဆိုတဲ့ မက်ဆွဲချုံပဲဖြစ်ပါတယ်။

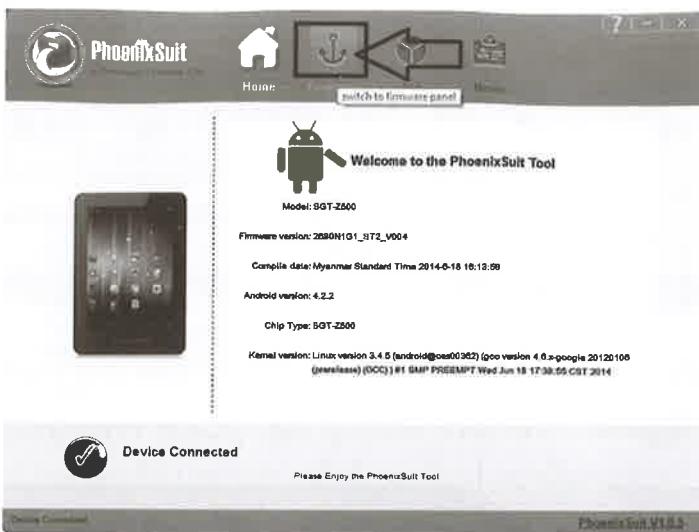


Ok ကိုနှိပ်ပြီး ကွန်ပူးတာနဲ့ ချိတ်ဆက်ထားတဲ့ Tablet ကို ဖြော်ဖြစ်၊ LiveSuit ကို Exit နဲ့ထွက်။

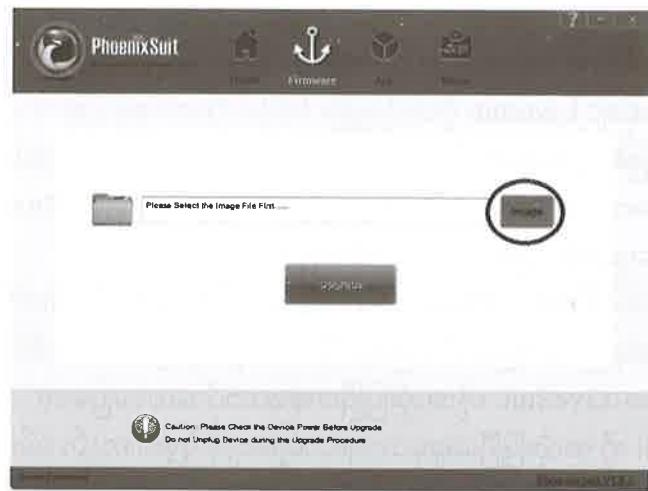


အခုခိုရင် Livesuit ကိုအသုံးပြုပြီး Tablet Firmware ရေးတာအောင်မြင်စွာ ပြီးဆုံးသွားပြီဖြစ်လို Tablet ကို ပါဝါဖွံ့ဗြည့်လိုက်ပါတော့။ မိမိ Tablet ဟာ ပုံမှန် အတိုင်းပဲ အလုပ်လုပ်နေတာကို ပျော်ဆွဲဖွယ်ရာမြင်ရမှာပါ။ PhoenixSuit နဲ့ Firmware ရောပုံတို့လည်း အနည်းငယ်ရောပြေားပါဉီးမယ်ခင်ဗျာ။

အခုထပ်ပြီးတော့ PhoenixSuit ကိုဘယ်လိုအသုံးပြုပြီး Firmware ရေးတယ်ဆိုတာကိုအနည်းအကျဉ်းထပ်ရေးပေးပါမယ်။ သူကပိုလွယ်မယ်ထင်ပါတယ်။ တချို့ Tablet တွေက LiveSuit ကိုအသုံးပြုပြီးရေးရတာပိုအဆင်ပြေပေမဲ့ တချို့ကိုတော့ PhoenixSuit ကို အသုံးပြုပြီးရေးမှုသာရတာမျိုးတွေ ကြံ့ဖူးပါတယ်။ မိမိကွန်ပူးတာမှာ တော့ ဆော့ဝဲနှစ်ခုစလုံးရှိထားသင့်ပါတယ်။ PhoenixSuit ကိုအသုံးပြုပြီး Firmware ရေးဖို့အတွက် ပထမဆုံး ကွန်ပူးတာမှာ PhoenixSuit ကိုအရင် Install လုပ်ပါ။ ကွန်တော် DVD အခွေထဲမှာ ထည့်ပေးထားပါတယ် အင်လိပ်ဟားရှင်းကို တင်လိုက်ပါ။ တင်ပြီးရင် PhoenixSuit ကိုဖွင့် Firmwaer ကို ကလစ်လုပ်လိုက်ပါ။

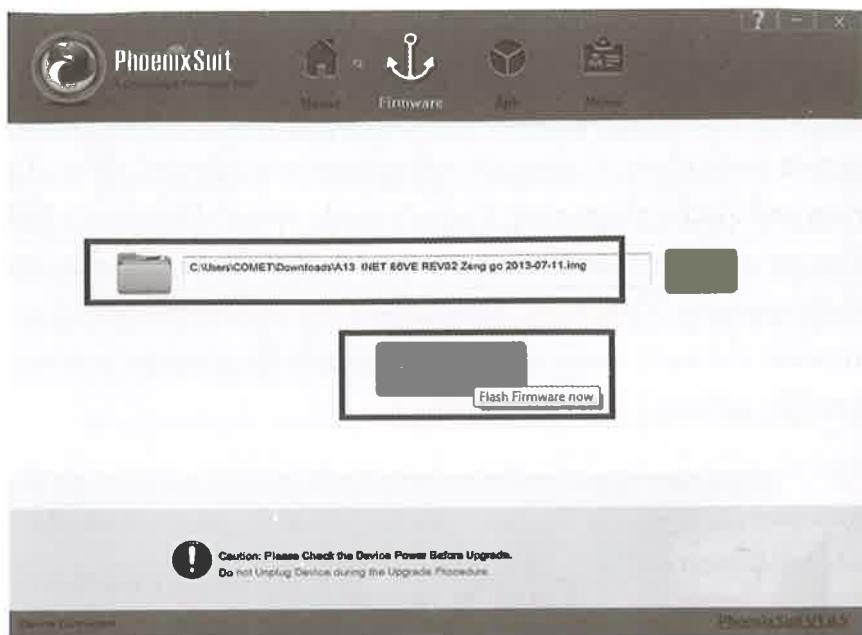


Firmware ကိုကလစ်လုပ်လိုက်တာနဲ့ ဘယ်နားက Image ဖိုင်ကိုထည့်မှာ လဲ ဆိုတဲ့ Firmware တင်မယ့်ဖိုင် (\*.img) ကို Image ကို ကလစ်နှင်းပြီး ရွေးပေးရ ပါမယ်။

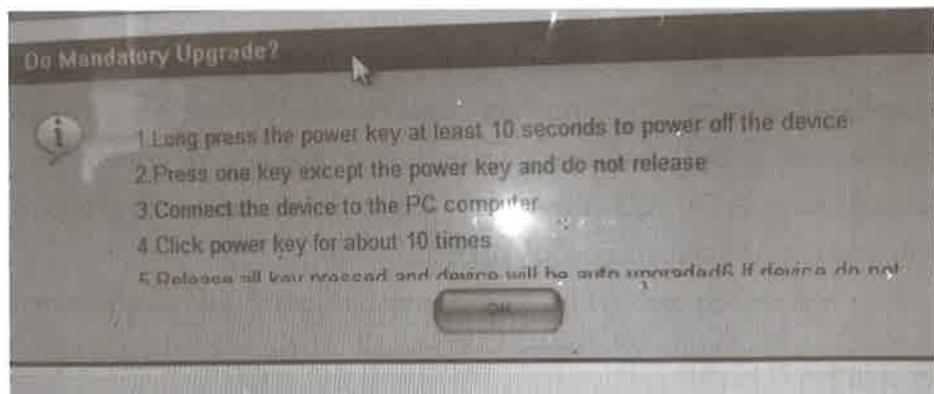


Firmware တင်မယ့် ဖိုင်ကိုရွေးပြီးတာနဲ့ အောက်မှာ Upgrade ဆိုတာကို နှင့်လိုက်ပါ။ မက်ဆွဲချုံဘောက်တွေ တစ်ခုပြီးတစ်ခု ကျလာပါလိမ့်မယ်၊ အားလုံးကို OK ပဲရွေးပေးပါ။

**THS တဗ္ဗိုလ်**



အခုလို Upgrade ကို ကလစ်နှင့်လိုက်တာနဲ့ မက်ဆွဲချုပ်ဘောက် များစွာကျေလာမှာဖြစ်ပြီး အားလုံးကို အိုကေပဲရှေ့ပေးပါ။ နောက်ဆုံး Tablet နဲ့ ကွန်ပျူးတာကို ကြီးနဲ့ချိတ်ဆက်ခိုင်းတဲ့ အနေအထားမျိုးရောက်ရင် ok မလုပ်သေးဘဲ Firmware ရေးမယ့် Tablet ကို Upgrade mode ထဲဝင်ပြီး ကွန်ပျူးတာနဲ့ ချိတ်ဆက်ပေးရပါတွေ့မယ်။



**THS စာအုပ်တိုက်**

Upgrade mode ကိုဝင်တာကတော့ LiveSuit နဲ့ Firmware ရေးတုန်းက ဝင်သလိုပါပဲ။ Vol+ ဒါမှမဟုတ် Vol- ကိုဖိန့်ထားရင်း Tablet ကိုကွန်ပျူးချိတ်ဆက် မယ်၊ ပါဝါခလုတ်ကို (၁၀)ချက်လောက် အဆက်မပြတ်နိုင်ပေးလိုက်ရင် Tablet ဟာ Upgrade mode ကိုရောက်သွားမှာပါ။ ကွန်ပျူးတာက ပထမဆုံး သက်ဆိုင်ရာ Tablet Driver တင်ပါမယ်။ သိသွားတာနဲ့ Upgrade mode အတွက် Mandatory Format လုပ်မလား မလုပ်ဘူးလားမေးပါမယ်။ အားလုံးဟာ LiveSuit ကအတိုင်းပါပဲ။ အားလုံး စိတ်တိုင်းကျရွေးချယ်ပြီးရင်တော့ PhoenixSuit က အစိမ်းရောင်ဘားတန်းလေးနဲ့ Firmware တင်နေတဲ့ အခြေအနေကိုပြပေးနေမှာပါ။ ပြီးသွားရင်တော့ အောက်က ပုံစံအတိုင်း မက်ဆောင်းပြပေးပါလိမ့်မယ်။



အခုခိုရင်တော့ PhoenixSuit ကိုအထုံးပြုပြီး Tablet ကို Firmware ရေးတာ အောင်မြင်ပြီးဆုံးသွားပါပြီ။ Firmware ရေးတာ အောင်မြင်ကြောင်းနဲ့ အချိန် ဘယ်လောက် ကြာခဲ့တယ်ဆိုတာကို ပြပေးပါတယ်။ ဒီလောက်ဆို စာဖတ်သူတွေ အနေနဲ့ A10၊ A13၊ A23 အစရှိသဖြင့် China Tabet တော်တော်များများကို Firmware ရေးနိုင်လောက်သွားပြီလို့ ယုံကြည်ပါတယ်။

ကျွန်တော်က ဘယ်လို Firmwaer ရေးရတယ်ဆိုတာကို အကြမ်းဖျင်းလောက် ပဲ ရေးပြပေးနိုင်ပါတယ်။ ကိုယ်တကယ်လုပ်တဲ့အခါ လက်တွေ့နဲ့ လွှဲခေါ်နေနိုင်ပါ

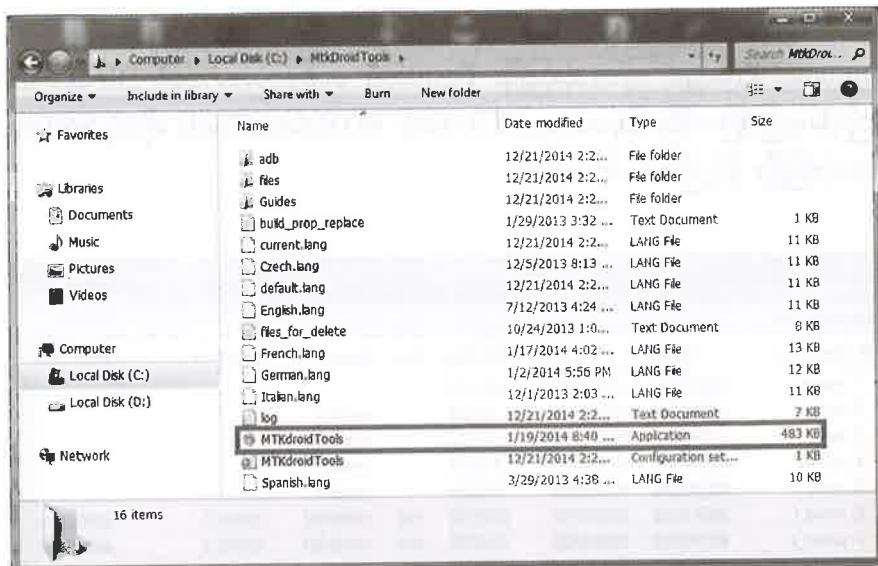
တယ်။ ဥပမာ တချို့ China Tablet တွေမှာ Upgrade mode ဝင်ဖို့အတွက် Volume ခလုတ်တွေမပါဘူး။ Home key နဲ့ Power Key နှစ်ခုပဲပါတာတွေကိုလည်း ကြံ့ရမှာပါ။ အဲအခါကျရင် ပထမဆုံး Home key ကိုဖိနိုင်ထားရင်း Tablet ကို ကွန်ပျူးတာမှာ ကြိုးထိုးပါ၊ ကြိုးထိုးပြီးတာနဲ့ ပါဝါခလုတ်ကိုပုံမှန်အတိုင်း ဆယ်ချက်လောက် အဆက် မပြတ်နိုင်ပေးထား ခြင်းဖြင့် Upgrade mode ကိုဝင်နိုင်ပါတယ်။ AML\_Flash Tool ကိုအသုံးပြုပြီး ONDA Tablet (e.g ONDA V971D) တွေကို Firmware ရေးပုံရေးနည်းကိုတော့ ဒီစာအုပ်မှာ မရေးပြတော့ပါဘူး။ တကယ်လို့ ကြံ့လာခဲ့ပြီ ဆိုရင်လုပ်ကြည်ပါ။ မခက်ပါဘူးလို့ ပြောပါရစေခင်ဗျာ။

အားလုံးအတွက် လက်ဆောင်စကားအနေနဲ့ကတော့ အခက်အခဲတွေ့လှုပ် စိတ်ဓာတ်မကျပါနဲ့ အကြိမ်တစ်ရာလုပ်လို့ မအောင်မြှင့်ခဲ့ရင် ဒီနည်းလမ်းတွေဟာ မအောင်မြှင့်နိုင်တဲ့ နည်းလမ်းတစ်ရာပဲလို့ သတ်မှတ်ပြီး ဖြစ်နိုင်မယ့်အရာတွေကို ဖြစ်နိုင်ချေတွေကိုဆက်လုပ်ပါ။ ကျွန်ုတ်တို့ မူလတန်းကတည်းကသင်ခဲ့တဲ့ သင်ခန်းစာထဲက ဆောင်ပုဒ်တွေလို့။ “င့်တိမိသတိုင် တက်နိုင်ဖျားရောက်၊ လိုလျှင်ကြံ့ဆ နည်းလမ်းရ . . .” ဆိုတဲ့အတိုင်းပါပဲ။ အားလုံးအဆင်ပြောပါစေလို့ ဆုမွန်ကောင်းတောင်းချွဲလိုက်ပါတယ်။

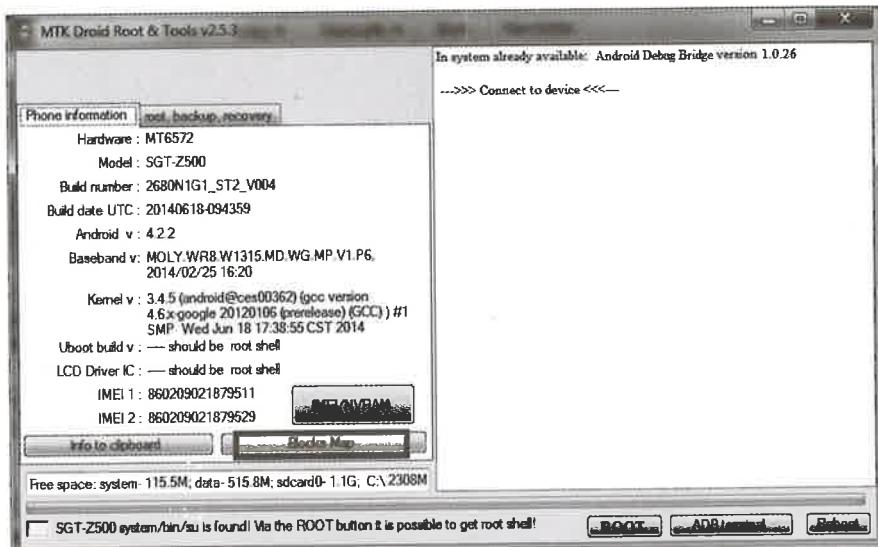
## (၈) MTK Chips Device များအတွက် Scatter ဖိုင်ထုတ်ယူခြင်း

MTK (MediaTek) Chips ကိုအသုံးပြုထားသော ဖုန်းများကို Firmware တင်မယ်ဆိုရင် ကျွန်တော်တို့ဟာ SP FlashTool ကိုအသုံးပြုပြီးတင်ကြရပါတယ်၊ လွယ်ကူမှုလည်းရှိပါတယ်။ ကျွန်တော်တို့ SP FlashTools နဲ့ တင်မယ်ဆိုရင် အခိုက အရေးကြီးတာကတော့ Scatter File ပါပဲ။ Scatter File မှာသတ်မှတ်ထားတဲ့အတိုင်း FlashTool ကမိမိတို့ Device တွေကို Firmware ပြန်တင်ပေးမှာပါ။ တစ်ခါတစ်ရုံ Scatter File မှားနေလို့ ရေးမပေါ်ဘူးဆိုတဲ့ မတဲ့ဆိုမျိုးတွေ့ရတက်ပါတယ်။ ကိုယ်က MTK6582 ရဲ့ Scatter File နဲ့ ရေးရမယ့်ဟာကို ကျွန်တော်တို့က MTK6592 ဆိုတဲ့ Scatter File နဲ့ သွားရေးနေတယ်ဆို ဘယ်လို့မှာအဆင်မပြောနိုင်ပါဘူး။ အဲဒီ အခါ Scatter File ကိုပဲ သီးသန့်ရှာပြီး ပြန်ရေးပေးရပါမယ်။ ကျွန်တော် ဒီမှာ ပြောပြချင်တာကတော့ ကိုယ်လို့ချင်တဲ့ Scatter File ကို ဘယ်လို့ ကိုယ့်ဘာသာကို လုပ်ယူမလဲဆိုတာပါပဲ။ ဥပမာအနေနဲ့ G610 ကို SP FlashTool နဲ့ Firmware တင်မယ်ဆို သူ့အတွက် Scatter File ကို သူနဲ့ အမျိုးအစားတဲ့ ဖုန်းတစ်လုံးတဲ့ကနေ ပြန်ပြီးယူလို့ရပါတယ်။

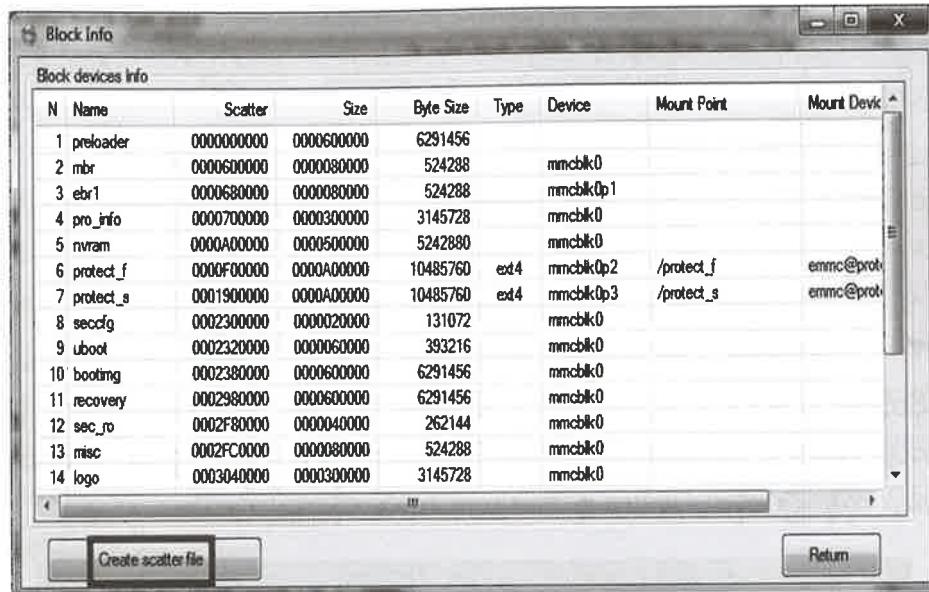
Scatter File ကို ထုတ်ယူဖို့အတွက် ကိုယ့်မှာ ကိုယ်သုံးမယ့်ဖုန်းနဲ့ အမျိုးအစား တဲ့ ဖုန်းရှိပေါ်မယ်။ ဖုန်းဆိုင်၊ ဆားပစ်ဆိုင်တွေအတွက်တော့ အမျိုးအစားတဲ့ ဖုန်းက အပိုရှိနေမှာဖြစ်လို့ စိတ်ပူဇာရာမလိုဘူးထင်ပါတယ်။ ပိမိကွန်ပျူးတာမှာ ချိတ်ဆက်ထားတဲ့ ဖုန်းအတွက်တော့ Driver တင်ပြီးသားဖြစ်ရပါမယ်။ Driver မသိသေးရင်တော့ Driver ကိုအရင်ဆုံးတင်ပေးလိုက်ပါ။ MTKdroid Tools ဆိုတဲ့ ဆော့ဝဲကိုအသုံးပြုပြီး Scatter File ကိုယူမှာဖြစ်လို့ ဆော့ဝဲလိုအပ်ပါတယ်။ MTKDeviceTools ကို Useful DVD ထဲမှာထည့်ပေးလိုက်ပါတယ်။ ပထမဆုံး MTKDeviceTools ကို မိမိ ကွန်ပျူးတာ မှာဖြည့်ချလိုက်ပြီး MtkdroidTool ဆိုတဲ့ ဖိုလ်ဒါလဲကနေ MTKdroidTools.exe ဖိုင်ကို ကလစ်နှစ်ချက်နှိပ်ပြီး ဖွင့်လိုက်ပါ။



ဖိုင်ဖွင့်ပြီး ခဏာလောက်စောင့်ကြည့်နေလိုက်ပါ။ Phone information ဆိုတဲ့အောက်မှာ ကွန်ပျိုးတာနဲ့ ချိတ်ဆက်ထားတဲ့ မိမိ ဖုန်း ဒါမှာမဟုတ် Device ရဲ့ အချက်အလက် အစုံအလင်မြင်ရမှာဖြစ်ပါတယ်။ ညာဘက်အောက်နားမှာ Blocks map ကိုနှိပ်ပြီး ကိုယ့် Device ရဲ့ Partition Table ကိုကြည့်ပါမယ်။



ကျွန်တော်တို့အရှေ့က အခန်း(၅) "dd command" အသုံးပြုပုံအခန်းမှာတော့ Command Prompt ကိုအသုံးပြုပြီး Partition တစ်နည်းအားဖြင့် Blocks map ကြည့်ပုံရေးပြုပြီးပါပြီ။ ဒီမှာတော့ GUI ကနေ MTKdroidTools ဆိုတဲ့ Software ကိုအသုံးပြုပြီး ကြည့်တာပါ။



Blocks map ကိုတွေပြုဆို ဘယ်ဘက်အောက်နားက "Create scatter file" ဆိုတာကို ကလစ်နှစ်၊ ကိုယ်နှစ်သက်ရာနေရာတို့ရေးပေးပြီး သိမ်းလိုက်ရင် ကျွန်တော် တို့ပေးလိုက်တဲ့နာမည်နဲ့အတူ ကျွန်တော်တို့လိုချင်တဲ့ MTK Device အတွက် Scatter file ရပါပြီခင်ဗျာ။

သီဟစိုး

## မာတိကာ

	Page
အခန်း (၁) ADB Commands အသုံးပြုရန်ပြင်ဆင်ခြင်း	7 to 10
အခန်း (၂) ADB Commands များကိုလေ့လာခြင်း	11 to 17
အခန်း (၃) ADB Commands များကိုအသုံးချခြင်း	18 to 27
အခန်း (၄) အဆင့်မြင့် ADB Commands များ	28 to 46
အခန်း (၅) Android Terminal အသုံးပြုခြင်း	47 to 64
အခန်း (၆) Fastboot Commands အသုံးပြုပုံ	70 to 82
အခန်း (၇) Adb Commands ဖြင့်လက်တွေပြသုနာ ဖြေရှင်းနည်း	83 to 103
အခန်း (၈) MTK Chips Device များအတွက် Scatter ဖိုင်ထုပ်သူခြင်း	140 to 106