

Bild 8a



Bild 8b

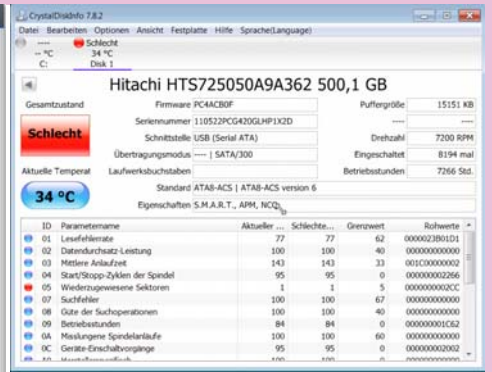


Bild 8c

mal bei 3.000 Euro an. Wenn Du Glück hast. Der Plan ist daher immer der gleiche. Erstelle einen Festplattenklon, sichere diesen weg und dann bearbeite diesen mit Recovery-Tools. Arbeite nie direkt auf der defekten Platte mit Tools. Das wird nichts. Auf dem Klon kann ich hingegen auch mit verschiedener Software mein Glück versuchen. Soweit der Plan.

Es gibt eine Phalanx von Windowsbasierten Datenrettungsprogrammen, die Festplatten mit schlechten Sektoren klonen können. Zumindest am Prospekt. Ich persönlich hatte bisher nie gute Erfahrungen mit dieser Klonsoftware gemacht. Entweder waren unnötig große Sprünge über defekte Sektoren bei „bad blocks“ oder gleich ganze Programmabstürze inklusive Einfrieren meine Begleiter. Selbst wenn ich einen Klon hinüberrettete, waren die Ergebnisse immer recht bescheiden. Und ein abgebrochener Klonvorgang kann zudem nicht dort fortsetzen, wo er aufgehört hat.

Das Hauptproblem mit Windowssoftware ist die eingeschränkte Ansteuerung zu der defekten Festplatte. Das hängt zum großen Teil mit der Treiberarchitektur und dem Host Controller von Windows zusammen. Dadurch kann die Software benötigte Timeouts beim Lesen von defekten Bereichen nicht steuern und manipulieren. Die Kontrolle über die Hardware fehlt eben. Ein Guru hat mir das einmal bei einem Kaffee erklärt. Zumindest habe ich es noch so in Erinnerung.

Unter Linux gibt es das Programm ddrescue. Dieses ist genau für solche Fälle entwickelt worden. Eines vorweg: Ist nichts für Anfänger. Hol Dir einen linuxkundigen Freund oder mach einen Crashkurs in Linux. Ein falscher Befehl kann Dir eine Platte und deren Daten unwiederbringlich löschen. Wenn Du Glück hast, ist es nicht Deine Systemplatte. Sofern Dir deine Daten lieb sind, investiere drei Stunden in die Linux-Basics oder ruf einen kompetenten Freund an, und Du hast gute Chancen, Deine Daten zurück zu bekommen.

Also ddrescue brauchen wir. Ist üblicherweise schon auf jeder Linuxdistribution drauf. Generell sind die üblichen Datenrettungs- und Klon-

programme (diskpart, parted, dd, ddrescue, photorec, testdisk, clonezilla...) auf jeder Linuxdistri schon drauf. Für das Auslesen von beschädigten Festplatten haben wir nun ddrescue. Das Tool kann defekte Sektoren (bad blocks) überspringen, die Festplatte weiters von hinten auslesen und hat auch eine Protokollfunktion. Diese ermöglicht einen beliebigen Abbruch des Klonvorganges. Vielleicht, weil man die Festplatte abkühlen lassen oder von der anderen Seite her den Klonvorgang fortsetzen will. Das Programm fängt nach einem Neustart dort an, wo es mit dem Klonen aufgehört hat. Das ist ein unschätzbare Vorteil gegenüber herkömmlicher Software.

Eine Datenrettung über einen SATA/USB Adapter ist nicht so prickelnd. Das zusätzliche USB Protokoll und der verbaute China-Chip im Adapter hat mir noch nie viel Glück gebracht. **Siehe Bild 9.**

Wenn zudem die Festplatte nicht mehr schnell reagiert, verliert der Adapter die Verbindung zum defekten Laufwerk. Zudem ist das USB Protokoll langsam, was wiederum den Klonprozess in die Länge zieht. Mehr Hitze, längere Bearbeitungszeit, mit USB noch eine zusätzliche Hardwareschicht dazu...alles nicht gut. Nach zwei Minuten reaktivierte ich einen alten Laptop mit einer Linux Live Cd und stöpselte das Laufwerk direkt an den SATA Port. **Siehe Bild 10 und 11.**

Auf **Bild 11** sieht man schön den abgesetzten Befehl zum Klonen. Das Image kann entweder auf eine andere Festplatte oder in eine Datei geschrieben werden. Ich entschied mich für Letzteres. Hier der Befehl `> ddrescue -fnR /dev/sda2 /Pfad/zur/Datei.img /Pfad/zur/Logdatei.log`. Das wars.

"f" steht für force, also ziehs durch. "n" verwendet man beim ersten Durchgang. Dieser Parameter bewirkt, dass man defekte Sektoren überspringen soll. Damit soll erreicht werden, dass das Image so schnell wie möglich erzeugt wird, ohne dass man jetzt mehrmals bei den defekten Sektoren "herumtut" und das Ganze eventuell noch verschlimmert. Erst wenn das Image fertig ist, kann man das Ganze nochmal

ohne dem Parameter "n" durchlaufen lassen. Das bedeutet, es werden jetzt nur mehr die defekten Sektoren näher betrachtet, aufgeteilt und jedes Bit herausgekitzelt. Der letzte Parameter "R" besagt eigentlich nur, dass ich die Erzeugung des Images von hinten startete, weil es von der vorderen Seite her schon sehr viele defekte Sektoren gab. Also versuchte ich mein Glück von hinten. Natürlich habe ich die Imagevorgänge auch öfters abgebrochen, wenn das Klicken zu laut wurde oder ich der Meinung war, die Platte braucht eine kleine Pause. Durch das Anlegen einer Logdatei (siehe oben den Befehl) weiß ddrescue genau, wo es nach einem abgebrochenen Klonvorgang wieder anfangen muss.

Somit ist das Logfile unverzichtbar und gleichzeitig eine Steuerdatei für ddrescue. Auf **Bild 11** sieht man mit 450 Tagen Restzeit schon, dass die Platte die Hufe streckt. Ein guter Indikator für den Gesamtzustand. Die Lesegeschwindigkeit variiert auch dementsprechend, je nachdem wie viele Sektoren kaputt sind. Das Programm bestimmt selber die Geschwindigkeit. Einzig beim Lesen von defekten Blöcken kann man die Leseversuche noch manipulieren. Also wie oft er probieren und die Sektoren aufteilen soll. Der Parameter „r5“ weist das Programm beispielsweise an, bei defekten Sektoren fünf mal zu lesen versuchen. Das ist wie erwähnt keine gute Idee beim ersten Klonvorgang. Später kann man das aber machen. Vorausgesetzt, es gibt schon ein Image. Dieses wird dann mit den Ergebnissen dieses Parameters vergrößert.

Wenn zum Schluss Lesefehler aufgelistet werden (**650 oben in Bild 11**), dann schickt man ddrescue ohne Option „n“ nochmal drüber. Die zuvor erwähnten Lesefehler beziehen sich natürlich nur auf den vorigen Durchgang. Das sei noch zur Vorsicht erwähnt. Die Gesamtzahl der Lesefehler beträgt weit über 150.000.

Damit wird versucht, diese fehlerhaften Blöcke noch weiter aufzuteilen, den Bereich zu überspringen und von rückwärts zu lesen, um Daten aus diesem Bereich zu retten. Mein erster Klonvorgang dauerte knapp 12 Tage für die 500 GB Platte. Ich bin mir sicher, dass dies auf



Bild 9



Bild 10

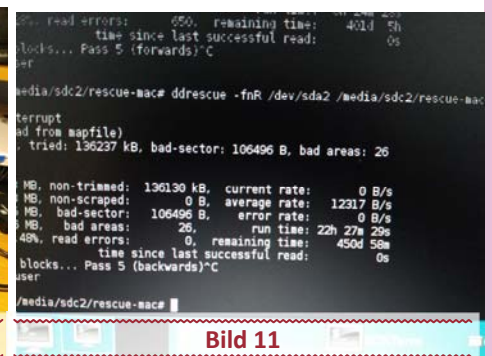


Bild 11