

Oprindelig: <http://www.kermitproject.org/kermit.html>

Om Kermit

Frank da Cruz, fdc@kermitproject.org



Indhold

- [HVAD ER KERMIT?](#)
- [KERMIT SOFTWAREN](#)
- [KERMIT-PROTOKOLLEN](#)
- [EKSEMPEL PÅ FILOVERFORSEL](#)
- [KONTROVERSER](#)
- [URLs](#)

NYHEDER

- 30 oktober 2013: Kermit Projekt **Mundtlig Historie Udskrifter** på [Computerhistorisk museum](#):
 - [Kermit Projekt Mundtlig Historie Panel](#)
 - [Joe Doupnik Mundtlig Historie](#) (MS-DOS Kermit)
- 29 oktober 2013: Den [Kermit Projektet Dokument Arkiv](#) på Computerhistorisk museum [[catalog](#)].
- Udvikling af C-Kermit [fortsætter](#) , men på et reduceret tempo.

Effektiv 1 Juli 2011...

- Kermit-projektet ved Columbia University blev aflyst
- Et nyt Kermit Projekt-websted er åbnet på <http://www.kermitproject.org>.
- Al Kermit-softwaren har fået en [Open Source-licens](#).
- Kermit ftp-arkivet og webstedet i Columbia forbliver på plads (men ændres ikke).

Velkommen til det nye Open Source-Kermit Projekt.

[Meddelelse](#) [Overgang køreplan](#) [Kermit 95](#) [C-Kermit](#) [E-Kermit](#) [Anden Kermit-softwaren](#)

HVAD ER KERMIT?

Kermit er navnet på en fil-overførsel og -styringsprotokol og en suite af edb-programmer til mange typer af computere, der gennemfører, der protokol samt andre kommunikations-funktioner spænder fra terminal emulering til at automatisering af kommunikation opgaver gennem en høj-niveau cross-platform scripting sprog. Softwaren er transport-uafhængig, driver over TCP/IP-forbindelser i traditionelle klar-tekst-mode eller sikret ved SSH, SSL/TLS, eller Kerberos IV eller V, samt via seriel-portforbindelser, modemer, og andre kommunikationsmetoder (X. 25, DECnet, forskellige LAN-protokoller såsom LPR og LAT, parallelle porte, osv, på især - platforme).

Den Kermit-Projektet blev grundlagt på Columbia University Edb-Center (nu [CUIT](#)) i 1981 for at opfylde et [specifikt behov](#), og indtil midten - til slutningen af Af 1990'erne, Kermit blev Columbia ' s standard desktop-tilslutning softwaren, der anvendes universelt af studerende, fakultet og personale til at oprette forbindelse fra skrivebordet mikrocomputere, PC, Macintosh og Unix-arbejdsstationer til den centrale design og indretning: [IBM mainframes](#) (1963-2017), den [DECSYSTEM-20s](#) (1977-1988), [CLIO](#) (Columbia første online biblioteksinformationssystem, 1984-2003), og Cunix (vores Unix-baserede servere, 1986-i dag), og at afdelingerne VAXes, PDP-11s, Sole, og andre minicomputere. I de tidlige dage af pc 'er og Pc' er men før en udbredt anvendelse af lokale netværk og på skrivebordet arbejdsstationer, der er forbundet til dem, Kermit softwaren forbundet skrivebordet e-mail, bulletin boards, fildeling, sms-behandling, beskeder og andre aspekter af den nye on-line kultur, der er nu taget for givet, lang før de erfaringer, der var tilgængelige på de fleste andre institutioner. På Columbia, DEC 20'erne og afdelingerne minicomputere er længe gået og IBM mainframes er nu kun for backoffice bruge, men Kermit softwaren er stadig anvendes til SSH-sessioner fra skrivebordet til CUNIX, og af den tekniske personale til system-og netværksadministration opgaver, for eksempel, [konfiguration af stativer fuld af HP blade-servere](#) , som de ankommer, [forvaltning af University ' s telefon-system](#), [CGI scripting](#), [alphapersonsøgning](#) af on-call medarbejdere, og så videre. Plus, selvfølgelig, af old-timere, der bare foretrækker sikkerheden og effektiviteten af [tekst-tilstand shell sessioner for e-mail](#) og for at få deres arbejde gjort, for eksempel, softwaren udvikling og ledelse.

I løbet af de år, den **Kermit-Projektet** vokset til en verdensomspændende kooperative nonprofit softwaren udvikling og distribution indsats, som har hovedkvarter i og koordineret fra [Columbia University](#), som Kermit softwaren blev porteret til eller udvikles til flere og flere computere og operativsystemer ([se liste](#)). Den Kermit Projektet er dedikeret til produktion af cross-platform, langvarig, **stabil**, standarder-kompatible, interoperable kommunikation softwaren, og har været aktivt engageret i at de [standarder proces](#). Kermit softwaren bruges over hele verden i alle sektorer af økonomien: det nationale regering, statslige og lokale myndigheder, akademiske, medicin og sundhedspleje -, ingeniør -, rumfart, non-profit og kommercielle.



Selvom terminalemulering i vid udstrækning er blevet erstattet af Internettet til onlineadgang, **Kermit softwaren fortsætter med at spille en rolle** i andre applikationer, såsom remote sensing og data indsamling, forvaltning og fejlfinding af netværks-og telekommunikationsudstyr, back office arbejde, fragt-og lagerstyring, medicinske forsikringskrav, indsendelse, elektronisk overførsel af midler, og online indgivelse af selvangivelsen. Kermit softwaren er indlejret i netværk, routere og switches, i mobiltelefon tårne, i medicinsk diagnostik og overvågning af udstyr, selv i pacemakere, ikke for ikke nævne kasseapparater af en hel stor-navn forhandlere. I 2002 Kermit fløj på den [Internationale Space Station](#) og Kermit softwaren er den kommunikation, der anvendes ved [EM APEX havet flyder](#) (venstre) leverer realtime data til orkanen forskere og trackers til denne dag (orkanen projektet ind i en ny udvidet fase i 2010 er baseret på en ny version af [Indlejrede Kermit](#)).



Siden 1980'erne, Kermit-protokollen og softwaren, har været brugt på **fabrikken** i programmerbare udstansning, skal du trykke på bremsen, laminering, fladskærms roll, klipning, metal - og plast-behandling, træbearbejdning, og andre maskiner. For eksempel, i den fremstilling af den **Boeing 787**, hvor Kermit er anvendt til at styre en [Tape Lag](#), der danner visse krop komponenter. Du kan læse mere om, hvordan Kermit er brugt på fabriksgulvet, [her](#) og [her](#).



1990'erne Kermit softwaren blev brugt i [USA Post Office automatisering](#), det spillede en central rolle i den [Brasilianske nationale 1994 valg](#) (den største i historien af verden op til, at tid), og det var centralt, at [FN 's nødhjælp mission til Bosnien](#), "forbinder hele spektret af projektet drift, fra mainframe, minicomputer, Pc ' er, til håndholdte enheder og strekcode læsere."



I 1980'erne robustheden af den Kermit-protokollen passer det ideelt til tjeneste i [Grøn Revolution](#) i Afrika, den fælles Europæiske USSR [Giotto plads mission](#), og måske mest bemærkelsesværdigt er med til at genetablere data kommunikation mellem OS forskningsstationer på Antarktiske fastlandet, efter at de var cut-off i 1986 i en computer uheld i løbet af den 9-måneders Antarktiske vinter. I 1989 en [international konference om Kermit](#) blev afholdt i Moskva, SOVJETUNIONEN, og Kermit sessioner blev featured på andre konferencer gennem 1980'erne i Tokyo, Bern, Paris, Nashville, og andre steder.



Den Kermit-protokollen og softwaren, der er opkaldt efter **Kermit the Frog**, stjerne i tv-serien The Muppet Show, navn Kermit er anvendt af tilladelse af Henson Associates, Inc. *Hvorfor er det opkaldt efter KermitFrøen?* I Maj 1981 havde vi allerede første implementeringer af protokol arbejde, men vi ikke har et navn til den protokol eller den softwaren endnu. En gruppe af os var at diskutere det (mig, Bill Catchings, Bill Schilit, Jeff Damens, jeg tror, det var den gruppe), uden rent faktisk at omsorgen for meget da vi har aldrig forventet, at den softwaren, til at sprede sig over hele verden,

og i sidste i årtier. Jeg tilfældigvis at være over den mur, der havde en Muppets kalender på det, og da mine børn var så store fans af the Muppet Show jeg sagde, Hvordan om Kermit? Tredive år senere (Maj 2011) fandt jeg kalenderen side, der var jeg kigger på, når jeg sagde, at du kan se det på venstre og du kan klikke på det for at se et større billede.

KERMIT SOFTWAREN

Kermit softwaren er blevet skrevet til [hundredvis](#) af forskellige computere og operativsystemer, noget af det af frivillige programmører over hele verden, noget af det, som Kermit Projekt professionelt personale. De vigtigste funktioner af de mest populære Kermit-programmer er:

- Forbindelse med oprettelse og vedligeholdelse af en bred vifte af forbindelse metoder (TCP/IP, X. 25 -, LAN -, seriel port, modem osv.).
- [Terminal emulering](#).
- Fejl-gratis fil overførsel.
- Internet protokoller, herunder Telnet, Rlogin, [FTP](#) og HTTP.
- [Internet security metoder](#) herunder Kerberos, SSL/TLS, SSH, og SRP.
- [Tegn-sæt konvertering](#) i løbet af både terminal emulering og fil transfer – en unik funktion af Kermit softwaren.
- Numeriske og alfanumeriske personsøger.
- Script-programmering til at automatisere komplicerede eller gentagne opgaver.

Kermit ' s brugerflade og script programmering sprog er konsistente på tværs platforme og metoder til kommunikation, så investering i at lære at pay-off tid og igen, efterhånden som du bevæger dig fra den ene platform til den anden, kommunikation metode til en anden.

Vores premiere Kermit softwaren implementeringer er:

- [Kermit 95](#) til Windows 95/98/ME, Windows NT/2000/XP/Vista/7/8/10, og OS/2;
- [C-Kermit](#) til [UNIX](#), VMS, VOS, og flere andre operativsystemet familier;
- [E-Kermit](#) til indlejring.
- [MS-DOS Kermit](#) til DOS og Windows-3.x;
- [IBM Mainframe Kermit](#) til VM/CMS, MVS/TSO, og CICS.

[C-Kermit](#) og [IBM Mainframe-Kermit](#) er host-baserede pakker, med en enestående vifte af alsidighed. [Kermit 95](#) og [MS-DOS Kermit](#) er fuldt udstyret desktop kommunikation softwaren programmer stridig kvaliteten af noget andet (eller fra) markedet, måske med undtagelse af i flashiness af brugergrænseflade: Kermit programmer følg tekst-tilstand spørg-og-kommando stil af gårsdagens, som er forvirrende at nogle mennesker indtil de indser fordelene:

- Den kommando, der er nogenlunde ensartet på tværs af alle platforme, og næsten helt konsekvent på tværs af moderne platforme såsom Windows, Mac OS X, Linux, og FOS. Lære det en gang, skal du bruge det overalt.
- Kommandoer kan være samlet i "makroer" eller "programmer" for at automatisere alle opgave, der kan gøres i hånden, som beskrevet [her](#). Faktisk i C-Kermit og Kermit 95, kommando-sproget er en fuld-

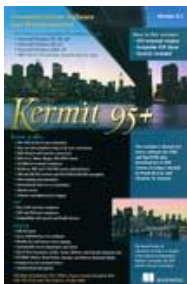
blæst programmeringssprog med variable, kontrolstrukturer, funktioner, "subrutiner", plus et par overraskelser.

- Du behøver ikke at kende kommandoer på forhånd eller skrive dem fuldt ud. Kommandoen stil kaldet "kontekstafhængig menu on demand" (du kan se de tilgængelige valgmuligheder, når du skriver et spørgsmålstegn), og søgeord kan være forkortet. Der er masser af indbyggede hjælp, og masser af hjælp til at findes på Kermit hjemmeside, for eksempel [C-Kermit tutorial](#) og [den Kermit 95 tutorial](#), bare til at starte med.
- Berøringsskrivere kan arbejde hurtigere, når de ikke behøver at flytte hænderne væk fra hjemmetasterne, og de får mindre gentagne belastningsskader.
- Visse ting kan bare ikke gøres effektivt eller ved hjælp af en GUI interface. Her er et helt tilfældigt eksempel, men det gør det punkt:

På en PC, jeg har en mappe, der indeholder tusindvis af billeder, sammen med deres miniaturebilleder. For hvert billede *xxx.jpg* der er en miniature *xxx-t.jpg*. Jeg ønsker at indlæse alle miniaturer i Photoshop. Hjælp musen, det ville tage hele dagen. Med Kermit du kan gøre det som denne (ved Kermit kommandoprompt):

```
mkdir thumbnails  
rename *-t.jpg thumbnails/
```

Og derefter på miniaturer undermappe, skal du Ctrl-A for at "vælge alle" og træk til Photoshop (og derefter, hvis det ønskes, træk miniaturerne tilbage til den oprindelige mappe med en mus bevægelse, eller omdøb dem tilbage med en Kermit-kommando).



Kermit 95 blev udviklet, ikke kun for at opfylde Columbia 's behov for forbindelse fra Windows 95 (og senere) til den centrale tekst-baseret tjenester, men også for at rejse penge til at støtte den Kermit-Projektet. I modsætning til andre Kermit programmer, K95 var strengt kommercielle, som findes i både en detail shrinkwrapped version (til højre) og i [bulk lige-til-at kopiere licenser](#). Fra udgivelsen i 1995 til midten af 2011, over en kvart millioner bulklicenssæder blev købt i 1000 licenser licenser, der spænder i størrelse fra 100 pladser til 10.000. Omkring 30.000 krympede eksemplarer blev solgt i mange tusinde flere, der købes til download fra [e-academy](#), og K95 var site-licens med over 100 universiteter samt hele statewide universitetet systemer såsom SUNY (64 campusser med omkring 400.000 studerende).

Den Kermit-Projektet blev sat på en selvstændig finansiering grundlag i 1984, og fra så indtil sin opsigelse i 2011, det indså **\$8,894,912.00** i indtægterne til Universitetet, plus et udstyr, tilskud (den [Eremit - Projektet](#)) til en værdi af **\$3,000,000.00**.

[Kermit bøger](#) [Kermit 95](#) [C-Kermit](#) [E-Kermit](#) [G-Kermit](#) [Aktuelle softwaren-versioner](#)

KERMIT-PROTOKOLLEN

Siden starten i 1981, den **Kermit-protokollen** har udviklet sig til en avanceret, kraftfuld og *udvidelig* transport-uafhængige værktøj for fil overførsel og forvaltning, herunder blandt andet:

- Overførsel af flere filer i en enkelt operation.
- [Fil attribut transmission](#) (størrelse, dato, tilladelser, osv.)
- Fil navn, registrerings-format, og [tegn-sæt](#) konvertering
- Fil kollision muligheder, herunder en "update" funktion
- Gendannelse af filoverførsel (genoptagelse af en afbrudt overførsel af binær tilstand fra fejlstedet)
- Automatisk upload og download
- Klient/Server-fil overførsel og forvaltning
- [Automatisk per-fil tekst/binary mode skift](#) under fil-gruppen transmission
- [Rekursive directory-tree-overførsel](#), selv mellem i modsætning til platforme
- Ensartet service på serielle forbindelser og netværk
- En [Internet Kermit Daemon Service](#)

Kermit-protokollen bruger veldefineret, sekventeret, fejl-kontrolleret pakker i hver retning af effekten af en fil-overførsel, følgende almindelige regler for protokol lagdeling. Pakker, der er designet for maksimal gennemsigtighed, så de kan passere selvom enhver kommunikationsform, uanset hvor restriktive. Half-duplex (stop og vente), fuld-duplex (skydevinduer med selektiv retransmission), og løbende streaming transport kan anvendes til at tilpasse sig nogen forbindelse.

Den funktion, der adskiller Kermit-protokollen fra de fleste andre, er dens bred udvalg af indstillinger til at tillade tilpasning til enhver type og kvalitet af forbindelsen mellem to former for edb — pakkelængde, pakke-kodning, vinduets størrelse, tegnsæt, fejl-metode til påvisning, timeouts, pauser. De fleste andre protokoller er designet til at arbejde kun på visse typer eller kvaliteter- forbindelser, og/eller mellem visse former for computere eller gerne fil-systemer, og derfor fungerer dårligt (eller slet ikke) andre steder, og tilbyder få, hvis nogen metoder til at tilpasse sig til ikke-planlagt-for situationer. Kermit, på den anden side, giver dig mulighed for at opnå en succesfuld filoverførsel, og den højest mulige præstation på en given forbindelse.

I modsætning til FTP eller X-, Y-, og ZMODEM (de andre protokoller som Kermit er de fleste ofte sammenlignet) Kermit-protokollen ikke antage eller kræve, at:

- en fuld-duplex forbindelse;
- en forbindelse, der er gennemsigtig for at kontrollere tegn;
- en 8-bit forbindelse;
- en ren forbindelse;
- store buffere alle sammen meddelelse sti;
- fysisk-link-layer-flow-styring.

(selv om Kermit ikke brug for nogen af disse betingelser, kan det tage fordel af dem, når de er tilgængelige). En kronik om Kermit-protokollen af Tim Kientzle i februar 1996 spørgsmålet om [Dr. Dobb ' s Journal](#) bemærkes, at "Kermit' s windowing tilgang er hurtigere end protokoller såsom XModem og YModem . . . Hvad mange mennesker ikke indser, er, at der under mindre end ideelle forhold, Kermit ' s windowing tilgang er væsentligt hurtigere end ZModem, en protokol med et velfortjent ry for hurtigtoverførsler i løbet af god kvalitet linjer." Effektiviteten af Kermit protokollen er analyseret i dybden, [her](#) og [her](#).

Således Kermit overførsler arbejde "ud af boksen" næsten hver gang. Og på en højere niveau, Kermit command language gør det muligt for alle slags praktisk fil udvælgelseskriterierne i blive brugt i en kombination, for eksempel:

- Jokertegn og mønstre til at matche filnavne
- Udvalg af datointervaller
- Valg af størrelse varierer
- Kun tekst-filer
- Kun binære filer
- Kun filer, der ikke eksisterer på den anden ende, eller der er nyere
- Undtagelse lister og mønstre

Til at udføre næsten enhver gruppering du kan forestille dig. I transit, en fil kan har sin character-set konverteret, kan det være passeret gennem et filter, m.m., og efter en vellykket overdragelse, kilde-fil kan blive slettet eller omdøbt, destinationsfilen kan omdøbes eller mail, og så videre.

Den Kermit-Protokollen Specifikation



Specifikationen for Kermit-filoverførselsprotokol er givet i bogen [Kermit, En File Transfer Protocol](#) af Frank da Cruz, med et forord af [Donald Knuth](#) (nu tilgængelig online i [PDF - format](#)). Specifikationen er også tilgængelige online på den Sjette udgave af [Kermit Protokol Manual](#) (1986). Begge disse mangler nogle af de senere justeringer, men de omfatter server mode, lang pakker, skydevinduer, osv. Dokumentation for den senere protokol tilføjelser er indsamlet og offentligt tilgængelige [HER](#). En **formel specifikation og verifikation** af den Kermit-protokollen blev offentliggjort af James Huggins af University of Michigan i 1995, og du kan download det [HER](#).

KERMIT EKSEMPEL PA FILOVERFØRSEL

Lad os se den fælles sag, hvor du har en Windows med et forbindelse — enhver form for forbindelse (modem, seriel port, regelmæssige Telnet, secure Telnet, rlogin, sikker rlogin, SSH) — til en shell - session på en Unix-server ("Unix" = Linux, Mac OS X, FreeBSD, Solaris, AIX, HP-UX, [osv](#)), og du vil overføre en fil mellem din PC og Unix-server. Din terminal emulator på Windows er [Kermit 95](#) og Unix server, har [C-Kermit](#) eller [G-Kermit](#) installeret, som kan påberåbes ved bare at skrive "kermit" påshell prompt (eller måske "ckermi" eller "gkermit").

For at *downloade* en fil, sig, `message.txt`, du skrive følgende kommando i en shell prompt:

```
kermit -s message.txt
```

Filen er sendt til Kermit 95 aktuelle mappe på din PC (eller til sin DOWNLOAD-MAPPE hvis du har defineret en). Det gør ikke noget, hvis filen er tekst eller binært; Kermit tal det ud, og overfører det automatisk i den relevante tilstand.

Det samme gælder, hvis du ønsker at overføre en gruppe af filer, sig alle de filer, hvis navne starter med "daglige.":

```
kermit -s daglige.*
```

Kermit sender hver fil, der svarer, skifter automatisk mellem tekst og binær tilstand, som er passende for hver fil (daglige.jpg, daglige.xls, daglige.txt, ...)

At *uploade* en fil fra din PC til Unix er bare så nemt. Antag, at du har en fil der hedder "budget.xls" i Kermit 95 aktuelle mappe på din PC. At uploade det til UNIX, type dette på Unix-shell-prompten

```
kermit -g budget.xls
```

Der er det grundlæggende, og der er mange variationer og finesser; for eksempel:

- Kun overføre filer, der er nyere end den tilsvarende på den anden ende.
- Konverterer tegnsæt, som tekst-filer på passende vis (fx mellem ISO 8859-1 og UTF-8 Unicode -).
- Gendanne en delvis overførsel fra det punkt af manglende (binær-tilstand).

For at spare dig selv nogle at skrive, kan du angive et alias på Unix (i din shell profil):

```
alias s="kermit -Ys"  
alias g="kermit -Yg"
```

(**s** for at Sende, **g** for Få). Og derefter:

```
s message.txt  
g budget.xls
```

Det er værd at bemærke, at du overfører dine filer i samme forbindelse du allerede har; der er således ingen skal oprette en ny forbindelse, ægthedsbekræftelse af dig selv, eller lignende bureaukrati. Hvis forbindelsen er sikret ved SSH, Kerberos, SSL, TLS eller SRP, så den file transfer er også sikker, automatisk.

Dette markerer en hidtil uset grad af bekvemmelighed. Når du fortæller C-Kermit på Unix til at sende eller modtage en fil, sin første fil-overførsel pakke er anerkendt automatisk ved Kermit 95 terminal-emulator, og K95 popper ind i enten modtage mode eller server-tilstand, afhængigt af den retning, og når overførslen er færdig, K95 vender tilbage til sin terminalemulerings-skærm. Hvis der er en fejl (for eksempel, hvis du ikke har skriveadgang i det destinationsmappen) K95 forbliver i sin fil-overførsel skærmen, så du kan se, hvad problemet var.

Den samme fremgangsmåde virker også Unix-to-Unix, K95-til-FOS -, Unix-til-FOS FOS til at Unix-eller OS/2 til FOS-eller Unix, så længe du bruger K95 eller C-Kermit som din terminal-program.

KONTROVERSER

Se også: [Populære Misforståelser](#).

I løbet af de år, den Kermit-Projektet og softwaren, der er omfattet af forskellige kontroverser, især:

Licens

Fra starten ønskede vi Kermit softwaren til at være fri til at alle. Men starter i 1984, Columbia University tvunget os til at finde en måde at gøre det betale for sig selv, det er, at betale løn for fuld - og deltidsansatte, og for udstyr, forsyninger, telefon, etc. Ellers kan vi ville ikke have lov til at fortsætte med at udvikle, vedligeholde, distribuere, og understøtter softwaren, som på dette tidspunkt var blevet populære over hele verden.

Vores løsning var at holde softwaren gratis for den enkelte og organisation for hans/hendes/sin egen brug, men at kræve, at selskaberne til at licensere det hvis de skulle til at bundle det med et produkt, eller ellers give det til kunder eller klienter, der er de, hvis

de ønsker at tjene penge fra vores arbejdskraft. Denne måde, de *kan* tjene penge, men de ville have til at dele det med dem, der udførte arbejdet.

Som den Fri Software-bevægelsen tog rod, dens fortalere indsigelse strenously til denne fremgangsmåde, men det er tilladt Kermit projektet til at fortsætte med yderligere 10 år. Så i 1994, med den kommende udgave af Microsoft Windows 95, vi har besluttet at frigive en og kun Kermit program, der var 100% kommerciel: [Kermit 95](#). Dette produkt tilladt Kermit Projektet til at blomstre indtil omkring 2003, hvor den AMERIKANSKE økonomi og verdensøkonomien begyndte at gå ned, og til at fortsætte med at eksistere i stigende grad formindsket form indtil 2011, hvor den Kermit-Projektet på Columbia University blev endelig aflyst. På det tidspunkt, da ingen job afhang af det noget mere, alle Kermit softwaren, som vi havde fulde rettigheder til, var placeret under en [Open Source licens](#) og nu alle er glade undtagen for dem, der mistede deres job, og dem, der kaldte vores gratis tech-support antallet, når de havde brug for hjælp. Og dem, der spekulerer på, hvorfor der aldrig var en anden Kermit 95 udgivelse.

Kermit vs X/Y/ZMODEM

Den [XMODEM](#)- filoverførselsprotokollen blev udviklet andre steder i 1977 for at overføre filer over telefon forbindelser fra en mikrocomputer til en anden, og dermed fundet bred anvendelse blandt computer hobbyfolk, BYTE magasiner fans, brugere og admins af [BBS systemer](#) og lignende. Dens efterfølgere, som YMODEM og ZMODEM, voksede op i den samme kultur, der tjener cirka det samme bruger-base. I BBS verden, kommunikation links var altid 100% gennemsigtig, at alle 256 byte værdier, så disse protokoller til at være forholdsvis enkel og stadig fungerer godt *i dette miljø* således indbyggere BBS/ hobbykulturen havde ingen grund til at bruge eller lære om Kermit.

Den Kermit-protokollen, på den anden side, var designet til mikro-mainframe forbindelser, der var meget mindre tolerant og meget mere krævende, fordi de forbindelser, som sjældent blev gennemsigtigt, og de underliggende computere blevder er radikalt anderledes, for eksempel, de kan bruge forskellige fil-formater og tegnsæt for filen opbevaring. Kermit, så var rettet mere mod institutioner — universiteter, hospitaler, virksomheder, offentlige agenturer — der var inde værelser med store centrale delte computere eller en mangfoldighed af afdelingerne minicomputere plus de enkelte brugere med Pc ' er eller arbejdsstationer på deres skriveborde, snarere end hobbyfolk alle med relativt homogene personlige mikrocomputere.

XMODEM var en smertefuldt langsom protokol, så den drivkraft var at udvikle det til hurtigere og hurtigere protokoller, og derfor YMODEM og ZMODEM. Men de nyere MODEM protokoller, der stadig antages, at et (mere eller mindre) 100% gennemsikkelig forbindelse mellem to identiske eller meget ens computere.

Som YMODEM og ZMODEM dukkede op, begyndte folk at kritisere Kermit protokol for at være langsom, så det var faktisk i sin oprindelige form: korte pakker, fordi de fleste mainframes kunne ikke modstå lange byger af indgående data fra enterminal; halv-duplex-stop-and-wait, fordi IBM mainframes ikke støttede fuld-duplex kommunikation; indkodning i kontrol af tegn og 8-bit-tegn, fordi disse ikke kunne passere gennem mainframe terminale

driver. Således er den oprindelige Kermit-protokollen var et "mindst fællesnævner" blandt alle de platforme, hvor det var nødvendigt at køre (og mange flere, foruden, som det viste sig). Dens store styrke var, at det var tilpasset enhver platform eller kommunikation metode, herunder dem, hvor XMODEM familie ikke passer på alle, for eksempel i IBM mainframe verden.

I mellemtiden, nogle BBS softwaren-pakker, der udbydes, Kermit protokollen om deres Upload og Download menuer, men de Kermit implementeringer var altid minimal (dvs langsom), der ofte er fejlbehæftet, og til tider helt ufunktionelt (se [Misforståelser side](#) om tredje-part Kermit protokol implementeringer). Denne tendens til at styrke indtrykket af, inden hobbyist kultur, der Kermit-protokollen var langsom.

Til at løse de resultater klager, vi tog fordel af den iboende strækbarhed af Kermit protokol design (i, hvilke overførsler, der begynder med en feature-forhandlingsfasen) for at tilføje muligheder for længere pakker og for fuld-duplex-skydevinduer med selektiv retransmission, samt muligheder for komprimering og for at drage fordel af gennemsigtige og/eller fejl-gratis-forbindelser (for eksempel, netværksforbindelser), når de var til rådighed. Disse ændringer, der er foretaget Kermit-protokollen så hurtigt eller hurtigere end ZMODEM uden at gå på kompromis med dens universalitet, data konvertering funktioner, robusthed, og (vigtigst) bagud-kompatibilitet (som er grunden til, at du kan du ikke se separate protokoller: XKERMIT, YKERMIT, ZKERMIT). Udførelsen ændringer, der kan dateres tilbage til omkring 1993; [se benchmarks](#).

Ikke desto hver lejr havde sine tilhængere i høj grad baseret på sin egen kultur, og hver tendens til at afvise andre, en tendens, der fortsætter frem til i dag. De fleste kritikere af Kermit basere deres bemærkninger om Kermit softwaren fra begyndelsen 1980'erne, eller på 3.-parts Kermit protokol implementeringer, som har en tendens til at arbejde dårligt. For en mere detaljeret gennemgang, se de [Misforståelser](#) side.

I 2013 har jeg lagt mærke til [Slashdot diskussion af annullering af Kermit Projekt på Columbia University](#). Det viser den nuværende emnet ganske godt, som det diskussionen er domineret af hobbyfolk og BBS-brugere. Men et par kyndigeKermit brugere også bidraget; her er nogle eksempler:

- Wow, i min college og indlæg college dage har jeg brugt denne protokol i så mange steder og på så mange måder, jeg kan ikke engang begynde at tælle. Det var en meget konservative protokol, der var i stand til at gå gennem næsten hvad som helst. Én gang Jeg havde den til at gå fra en bærbar computer via en modem-forbindelse til en Equinox data skifte til et AT&T 3b5 Unix, at en cu tilbage til Equinox (for at ændre den hastighed fra 300 baud til 9600 baud) til IBM-7171 protokol konverter til en IBM 4361. Og det faktisk kunne overføre filer. En anden gang havde jeg stress test en DECNET terminale simulator på en Sun (den gamle version vil mislykkes i den midt på dagen på de travleste dage), Så jeg brugte kermit til at oprette forbindelse til host1, derefter til host 2, tilbage til host 1, tilbage til host 2, jeg tror, at noget som 40 gange. Så jeg gjorde en fil overførsel gennem alle forbindelser. Det arbejdet.

- Wow. I begyndelsen af 90'erne, var jeg ansvarlig for at tilslutte den første Rumænsk universiteter (Bukarest, i særdeleshed) til Internettet. Da vi ikke kunne få IP-gå til forskellige tekniske årsager, besluttede vi at få dem i e-mail den gennemsnitlige tid, mindst. Den første prøve var med uucp, men de kunne ikke håndtere sine operationer på Bukarest side. Telefonlinjer var ikke stabil nok, så. Så for 1. 6 måneder, e-mail blev sendt til Bukarest med Kermit filoverførsel, udløst af en hodge-podge af MDA-scripts, der påberåbes af sendmail. Kermit var langt mere robust end nogen anden file transfer protocol på dette tidspunkt, vi troede til sidst det kunne håndtere smule overførsler over det våde tøj linjer.
- Ja, det bliver brugt en masse i den integrerede verden. En af de få værktøjer til rådighed til at inddrive en muret RS232-kun baseret enhed. Bruges på ting som den gumstix, beagleboard, og masser af andre SBC som ARM-baseret integreret enheder. Hvis du laver/for tilpassede versioner eller din egen shipping produkt ikke indeholde alternativer såsom MMC - /SD-kort boot kapaciteter, c-kermit er en af de få ting der giver dig mulighed for at starte op, indlæse koden, og derefter gå til konsol alle fra ét redskab på sådanne enheder. Gemt på min (og mine arbejdsgivere) røv mange gange på muret eller buggy indlejrede enheder.

I den samme diskussion, at der er nogle klager over, at der ikke er tilstrækkelig forklaring til, hvorfor nogle moduler af Kermit 95 ikke kunne udgivet i Open Source. Forklaringen var, og er, [HER](#).