۱) پروتکل TelNet

یکی از قابلیتهای اولیه یونیکس آن بود که امکان "ورود به سیستم"[۱] را از راه دور برای کاربران فراهم می‌کرد؛ یعنی یک کاربر می‌توانست با در اختیار داشتن یک "ترمینال"[۲] از هر مکانی و با استفاده از یک خط ارتباطی همانند خط تلفن با سیستم ارتباط برقرار کرده و به سیستم وارد شود و از آن سرویس بگیرد. یک کاربر مجاز در این سیستم ، ابتدا ارتباط فیزیکی ترمینال خود را با کامپیوتر مرکزی برقرار کرده و پس از وارد کردن شمارهٔ شناسایی و کلمهٔ عبور ، توسط سیستم یونیکس احراز هویت می‌شود؛ در صورت مجاز شناخته شدن ، یک پروسهٔ خاص برای سرویس‌دهی به او ایجاد خواهد شد. در حقیقت با این قابلیت ، یک کاربر در هر کجای دنیا ، در صورتی که بتواند ارتباط فیزیکی خود را با مرکز کامپیوتر برقرار کند ، قادر خواهد بود از سیستم سرویس بگیرد؛ با این قابلیت ، تفاوتی بین یک کاربر که در مقرّ کامپیوتر مرکزی نشسته و یک کاربر راه دور وجود ندارد.[۳]

با ارزان و سریع شدن سخت‌افزار و توسعهٔ خدمات اینترنت ، کامپیوترهای شخصی به خانه‌ها راه یافتند ولی هنوز کاربرانی وجود دارند که نیازمند آن هستند تا بجای استفاده از ترمینال ، از طریق کامپیوتر شخصی خود به یک سیستم راه دور وارد شوند. به عنوان مثال فرض کنید که شما یک کامپیوتر شخصی پنتیوم با سیستم عامل MS-Windows 9x در اختیار دارید ولی دانشگاه شما دارای یک سیستم مینی‌کامپیوتر SUN با سیستم عامل یونیکس است. شما برنامه‌های کاربردی خود را در محیط یونیکس نوشته‌اید و همانجا ذخیره کرده‌اید. حال به فرض اگر خواستید در منزل خود همانند کسی که در مرکز کامپیوتر نشسته است به محیط یونیکس وارد شده و برنامه‌های خود را ویرایش یا اجرا نمایید ، نیازمند یک ترمینال سازگار با یونیکس هستید ولی تنها چیزی که در اختیار شماست یک کامپیوتر شخصی است. در اینجا برنامهٔ Telnet راهگشاست.

برنامهٔ TelNet یک ترمینال مجازی و سازگار با ترمینالهای حقیقی از سیستم سرویس‌دهنده ، بر روی کامپیوتر شما شبیه‌سازی می‌کند و اجازه می‌دهد به سیستم یونیکس وارد شده و با آن محاوره نمایید. برنامهٔ TelNet فرامینی را که صادر می‌کنید به نحو مناسبی به سمت کامپیوتر راه

---

[۱] Remote Login

[۲] ترمینال ابزاری است که شامل یک صفحه نمایش ، یک صفحه کلید و یک ابزار ارتباطی -همانند یک مودم- جهت مخابرهٔ داده‌ها می‌باشد. ترمینال را نمی‌توان یک کامپیوتر مستقل به حساب آورد.

[۳] یک کاربر در محیط یونیکس ، به روش "اشتراک زمانی" از سیستم سرویس می‌گیرد و باید محدودیتهایی که سیستم عامل بر او تحمیل می‌کند را بپذیرد ، چراکه ترمینال ، یک کامپیوتر مستقل نیست و هیچ امکاناتی به غیر از محاوره -Conversation- با سیستم عامل در اختیار کاربر نمی‌گذارد.

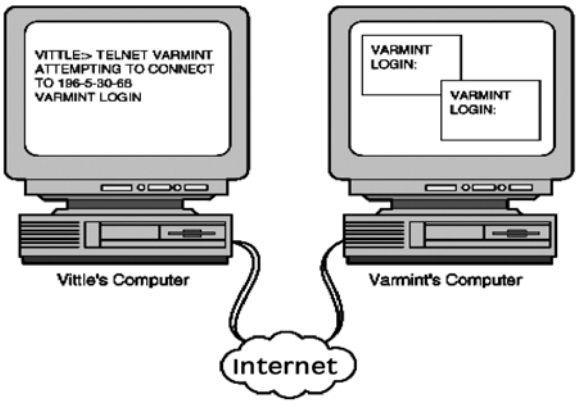
دور هدایت می‌کند و پس از تفسیر و اجرای فرمان صادره بر روی آن کامپیوتر ، نتیجه به برنامهٔ TelNet بر روی کامپیوتر شما باز خواهد گشت. بنابراین در یک تعریف ساده ، برنامهٔ TelNet موظف است بر روی ماشین کاربر ، مشخصه‌های ترمینال حقیقی سرویس‌دهنده را شبیه‌سازی نماید. به این ترمینال شبیه‌سازی شده اختصاراً NVT[1] گفته می‌شود.

در یک نگاه ساده برنامهٔ TelNet ، برنامهٔ ساده‌ای به نظر می‌رسد ، چرا که موظف است پس از برقراری یک "نشست"[2] فرمانهای کاربر را به سمت ماشین سرویس‌دهنده ارسال کرده و نتایج خروجی را نشان بدهد؛ ولی در مجموع برنامهٔ TelNet پیچیده‌تر از آنست که نشان می‌دهد ، چراکه موظف است با ترمینالهای متفاوت خود را تطبیق بدهد. به عنوان مثال فرض کنید یک کاربر از کامپیوتری با کدهای ASCII استفاده می‌کند در حالی که تمایل دارد به سیستمی وارد شود که استاندارد آن کدهای EBCDIC است؛ آگاهی از این موضوع و تبدیل این کدها به عهدهٔ برنامهٔ TelNet است. یا اگر کاربر یک برنامه به زبان C را بر روی سرویس‌دهنده اجرا کند ، نتایج خروجی منطبق با کنسول خروجی ماشین سرویس‌دهنده است نه ماشین خودش ، بنابراین تطبیق صفحه نمایش شبیه‌سازی شده ، به عهدهٔ برنامهٔ TelNet است.

مقصود از یک "نشست TelNet" برقراری موفق یک ارتباط TCP با پورت ۲۳ (یا یکی از پورتهای شناخته شده) از ماشین سرویس‌دهنده است به‌گونه‌ای که ماشین سرویس‌دهنده ضمن پذیرش این ارتباط و احراز هویت کاربر (در صورت لزوم) ، آمادهٔ پذیرش فرمانهای صادره از کاربر و اجرای آنها شود. در شکل (۱–۸) مراحل یک "نشست TelNet" به تصویر کشیده شده است. این نشست با اجرای برنامهٔ TelNet در خط فرمان آغاز می‌شود. در مثال زیر حروف پررنگ توسط کاربر نوشته شده و بقیه ، پیغامهای عمومی برنامهٔ TelNet هستند. در این مثال نام ماشین سرویس‌دهنده varmint و دارای سیستم عامل یونیکس و سخت‌افزار SUN است.

```
 telnet varmint
Trying 194.5.30.68 ...
Connected to varmint.
Escape character is '^]'.
SunOS UNIX (varmint)
login: anna
Password:******
varmint%
```

[1] Network Virtual Terminal
[2] TelNet Session



شکل (۱–۸) برقراری نشست Telnet

برنامهٔ TelNet در دو قسمت سازماندهی می‌شود:

- **پروسهٔ سرویس‌دهندهٔ TelNet** : این برنامه که بر روی کامپیوتر سرویس‌دهنده نصب و اجرا می‌شود ، موظف است تقاضاهای ورودی برای برقراری یک نشست TelNet را بپذیرد و پس از هماهنگی‌های لازم با برنامهٔ مشتری ، به او سرویس بدهد. این برنامه در محیط یونیکس به نام telnetd[۱] شناخته می‌شود.
- **پروسهٔ مشتری TelNet** : این برنامه که بر روی کامپیوتر کاربران نصب می‌شود و منطبق بر سخت‌افزار و سیستم عامل ماشین کاربر است وظیفه دارد تا مراحل برقراری یک نشست TelNet را برقرار کرده و یک ترمینال مجازی را به گونه‌ای شبیه‌سازی نماید که فرامین صادره از کاربر ، منطبق و سازگار با ماشین سرویس‌دهنده باشد. بطور عام این برنامه telnet نامیده شده است.[۲]

وقتی یک نشست TelNet برقرار شد کاربر می‌تواند یک فرمان را به سمت سرویس‌دهنده ارسال نماید. حال هر کلید یا فرمانی که بر روی کامپیوتر او فشار داده می‌شود ، باید از پروسه‌های گوناگونی عبور نماید تا تحویل برنامهٔ کاربردی شود. این روال بصورت زیر است:

---

[۱] Telnet Daemon

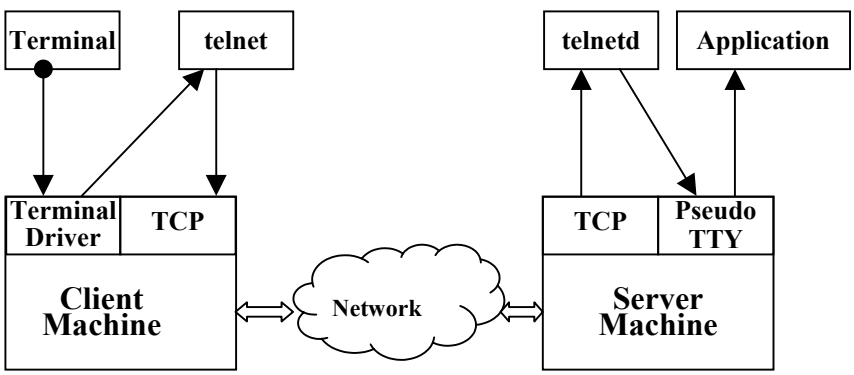
[۲] در محیط یونیکس برنامه **rlogin** نیز شرایط ورود به سیستم را از راه دور ، فراهم می‌نماید.

الف : ابتدا برنامهٔ TelNet بـر روی کـامپیوتر کـاربر ، آن کـاراکتر را تحویـل گرفتـه و پـس از پردازش لازم (از لحاظ تغییر استاندارد کد) از آن یک بستهٔ TCP ساخته و از طریـق لایـه‌های زیرین آنرا به سمت ماشین مقصد به جریان می‌اندازد و نهایتاً در مقصد تحویل سـرویس‌دهندهٔ TelNet می‌شود.

ب : برنامهٔ TelNet بر روی کامپیوتر سرویس‌دهنده آنرا تحویل گرفته و در صورت لزوم پس از تبدیل کد و با کمک سیستم عامل ، در اختیار برنامهٔ کاربردی قرار می‌دهد.

ج : برنامهٔ کاربردی ضمن پردازش آن ، خروجی مناسب را تولید و به سمت برنامـهٔ TelNet در سمت کاربر هدایت می‌نماید.

در شکل (۲-۸) جریان عبور داده‌ها از بین پروسه‌های مختلف بیـن برنامـهٔ مشـتری تـا برنامـهٔ کاربردی در ماشین سرویس‌دهنده ، نشان داده شده است.



شکل (۲-۸) جریان عبور داده‌ها بین پروسه‌های مبدأ و مقصد در TelNet

حال فرض کنید که سیستم عامل در کامپیوتر سرویس‌دهنده ، مبتنی بر "پنجره"[۱] باشد بدین معنا که بجای صدور فرامین در خط فرمان ، کاربر در یک محیط گرافیکی (شامل پنجره ، منـو ، آیکون و ...) از سیستم سـرویس بگـیرد. در اینجـا اتصـال بـه سیسـتم از راه دور ، شـرایطِ پیچیده‌تری دارد. زیرا این محیط گرافیکی باید بر روی کامپیوتر کاربر شبیه‌سازی شود؛ ولی هر

[۱] Window Based OS

عملی که کاربر بر روی این محیط انجام می‌دهد باید توسط سرویس‌دهندهٔ راه دور پردازش شود. در چنین شرایطی ، حرکت موْس ، هرگونه کلیک یا فشار کلیدهای صفحه کلید باید سریعاً به سمت سرویس‌دهنده ارسال و پاسخ آن از سرویس‌دهنده برگردد.[۱] در این محیط حجم اطلاعاتی که بین ماشین محلی و ماشین راه دور مبادله می‌شود بسیار زیادتر از برنامه‌های مبتنی بر "خط فرمان" است. چرا که هر حرکت مکان‌نمای موْس باید به سرویس‌دهنده گزارش شود. (برنامه‌های TelNet برای اتصال به سرویس‌دهنده‌های مبتنی بر پنجره وجود دارند ، همانند Motif یا X Windows .)

برای آنکه برنامهٔ TelNet را در خط فرمان اجرا نمایید ، ابتدا اسم برنامه و نام ماشین سرویس‌دهنده نوشته می‌شود. مثال:

telnet **leo.nmc.edu**

یا telnet **205.150.89.1**

telnet **varmint**

اگر از نامهای نمادین بجای آدرس IP استفاده می‌کنید باید این آدرسها قابل تحلیل و ترجمه به آدرس IP باشد.[۲]

```
merlin> telnet tpci_hpws4
Trying...Connected to tpci_hpws4.
Escape character is '^]'.
HP-UX tpci_hpws4 A.09.01 A 9000/720 (ttys2)
login: tparker
password: ********
tpci_hpws4-1 $  pwd
/u1/tparker
tpci_hpws4-2 $ cd docs
tpci_hpws4-3 $ pwd
/u1/tparker/docs
```

[۱] به عنوان مثال وقتی شما در این محیط بر روی یک منوی **Pull Down** کلیک می‌کنید هیچ پنجره‌ای ظاهر نمی‌شود مگر آنکه سرویس‌دهندهٔ راه دور اجازه بدهد.

[۲] اگر برنامهٔ **Telnet** را بدون مشخص کردن نام ماشین سرویس دهنده اجرا کنید ، این برنامه اجرا می‌شود ولی منتظر می‌ماند تا فرامین داخلی **Telnet** را اجرا نمایید. فرامین داخلی فرامینی هستند که به برنامهٔ **Telnet** مربوط می‌شوند و بر روی شبکه ارسال نخواهند شد.

```
tpci_hpws4-4 $ <Ctrl+d>
Connection closed by foreign host.
merlin>
```

وقتی برنامهٔ Telnet را برای ورود به یک سیستم راه دور بکار می‌گیرید ، پس از برقراری ارتباط ، از شما یک کد کاربری و یک کلمهٔ عبور خواسته می‌شود و در صورت تصدیق هویت ، علامت خط فرمان یونیکس[۱] روی خروجی ظاهر می‌شود. با دیدن این علامت هر فرمانی که روی خط فرمان صادر می‌نمایید به سمت سرویس‌دهنده ارسال خواهد شد. در مثال بالا پس از برقراری ارتباط ، پیغام زیر روی خروجی نشان داده شده است:

```
Escape character is '^]'.
HP-UX tpci_hpws4 A.09.01 A 9000/720 (ttys2)
```

این پیغام به این معناست که شما پس از برقراری نشست TelNet ، هر فرمانی که صادر کنید به سمت سرویس‌دهنده ارسال خواهد شد ولی اگر خواستید که فرامین کاربری Telnet را اجرا کنید می‌توانید با فشار دادن کلیدهای [+Ctrl به "حالت فرامین کاربری" بروید. در این حالت هر فرمانی که صادر می‌شود بجای ارسال به سرویس‌دهنده بصورت محلی و توسط برنامهٔ TelNet تحلیل خواهد شد. برای ختم یک نشست TelNet از کلیدهای Ctrl+D استفاده می‌شود.[۲]

در خط دوم از پیغام ، ضمن معرفی ماشین سرویس‌دهنده ، پارامترهای استاندارد ترمینال که بر روی آن توافق شده اعلان گردیده است. (ttys2)

بگونه‌ای که دیده می‌شود پس از برقراری نشست TelNet فرامین سیستم عامل یونیکس بر روی ماشین راه دور ، صادر و اجرا شده است.

برای برقراری ارتباط با یک سیستم عامل مبتنی بر پنجره :

اولاً باید سرویس‌دهندهٔ مورد نظر کاربر ، چنین سرویسی را فراهم آورده باشد.

ثانیاً برنامهٔ TelNet بر روی ماشین کاربر باید چنین قابلیتی را داشته باشد.

ثالثاً طرفین بر روی پارامترهای استاندارد توافق کنند تا بتوانند فرامین مربوط با پنجره را تحلیل کرده و خروجی لازم را نشان بدهد.

---

[۱] UNIX Prompt

[۲] **ختم یک نشست TelNet ، با استفاده از Ctrl+D ، در حقیقت معادل با عمل خروج از سیستم –logout– است.**

از TelNet می‌توان برای برقراری ارتباط با سرویس‌دهنده‌های دیگری مثل سرویس‌دهندهٔ HTTP (پورت TCP شمارهٔ ۸۰) بهره برد. در این حالت پس از برقراری نشست می‌توان فرامین پروتکل HTTP را ارسال کرد. مثال:

```
telnet www.csc.com 80
Trying 18.23.0.23 …
Connected to www.csc.com.
Escape character is '^]'.
GET /client-server-computing/toc.html HTTP/1.0
```

در مثال فوق شمارهٔ پورت پروسهٔ مقصد که باید نشست با آن برقرار شود ، ۸۰ معرفی شده است ولی اگر در خط فرمان شمارهٔ پورت تعیین نشود بصورت پیش‌فرض پورت TCP شمارهٔ ۲۳ که متعلق به پروسهٔ سرویس‌دهندهٔ TelNet است در نظر گرفته خواهد شد. سطر آخر در مثال بالا توسط کاربر نوشته شده و یک فرمان معتبر برای سرویس‌دهندهٔ HTTP محسوب می‌شود.

۲) فرامین TelNet

فرامین در TelNet بر دو نوعند:

الف) فرامین داخلی : این فرامین دارای قالب استاندارد و جهانی هستند و بین سرویس‌دهندهٔ TelNet در ماشین راه دور و برنامهٔ مشتری مبادله می‌شوند و کاربر دخالتی در مبادلهٔ این فرامین نخواهد داشت بلکه فقط در صورت تمایل می‌تواند مبادلهٔ آنها را ببیند.

ب) فرامین کاربری : این فرامین یکسری از دستورات کاربری در محیط TelNet هستند و با صدور آنها کاربر می‌تواند با برنامهٔ TelNet در ماشین خود "محاوره"[۱] داشته باشد. در مثال زیر فرامینی که در زمینهٔ خاکستری نشان داده شده‌اند فرامین کاربری در برنامهٔ TelNet هستند.

```
tpci_server-1> telnet
telnet> toggle options
Will show option processing.
telnet> open tpci_hpws4
Trying...
```

[۱] Converse

```
Connected to tpci_hpws4.
Escape character is '^]'.
SENT do SUPPRESS GO AHEAD
SENT will TERMINAL TYPE (don't reply)
SEND will NAWS (don't reply)
RCVD do 36 (reply)
sent won't 36 (don't reply)
RECD do TERMINAL TYPE (don't reply)
RCVD will SUPPRESS GO AHEAD (don't reply)
RCVD do NAWS (don't reply)
Sent suboption NAWS 0 80 (80) 0 37 (37)
Received suboption Terminal type - request to send.
RCVD will ECHO (reply)
SEND do ECHO (reply)
RCVD do ECHO (reply)
SENT won't ECHO (don't reply)
HP-UX tpci_hpws4 A.09.01 A 9000/720 (ttys2)
login: tparker
password: ********
tpci_hpws4-1 $
```

در این مثال ، فرمان کاربریِ toggle options باعث شده است که جریان مبادلهٔ فرامین داخلی بین سرویس‌دهندهٔ TelNet و برنامهٔ محلی ، به کاربر نشان داده شود. فرمان کاربری open نیز از برنامهٔ TelNet خواسته است که با یک ماشین خاص ارتباط برقرار کند.

با توجه به آنکه فرامین و داده‌ها در TelNet همگی روی یک کانال و بصورت رشته‌ای از کاراکترها ارسال می‌شوند لذا برای تمایز بین داده‌ها و فرامین ، یکسری کدهای فرمان تعریف

شده است. هر فرمان ابتدا با "کاراکترِ کد ۲۵۵"[1] شروع می‌شود و سپس کد فرمان بعد از آن قرار می‌گیرد و نهایتاً در صورت لزوم یکی از کدهای عمل اختیاری قرار می‌گیرد. پس بطور کلی فرامین داخلی در TelNet قالبی بصورت زیر دارند:

IAC[2]	Command Code	Option Code

مثال :

255 , 243 , 1

دقّت کنید در مثال بالا فرمان جمعاً سه بایت است که مقادیر عددی هر یک از کدها نوشته شده است؛ در ضمن کد عمل اختیاری می‌تواند حذف شود.

در جدول (٤-٨) برخی از کدهای فرمان که پس از کد ۲۵۵ قرار می‌گیرد ، تعریف شده‌اند. در جدول (٥-٨) نیز کدهای عمل اختیاری ارائه شده است.

شرح بعضی از کدهای فرمان در زیر آمده است:

- Will , Won't , Do , Don't : این چهار کد ، اصلیترین کدهای فرمان هستند که برای توافق و هماهنگی سرویس‌دهنده و برنامهٔ TelNet روی ماشین محلی کاربرد دارند. با این کدها یکی از طرفین ، گزینه‌ای را پیشنهاد می‌دهد و طرف دیگر می‌تواند آنرا پذیرفته یا رد کند.

Will با کد ۲۵۱ : با این کد یکی از طرفین می‌تواند عملی را به طرف دیگر پیشنهاد بدهد. مثال :

Character Code 255	Character Code 251	Character Code 1
IAC	Will	Echo

معادل نمادین :

با ارسال این سه کد از سرویس‌دهنده خواسته شده است که پس از دریافت هر کاراکتر آنرا برگشت بدهد.[3]

Won't با کد ۲۵۲ : با این کد یکی از طرفین تقاضای عدم انجام یا لغو یک عمل را می‌نماید. مثال:

Character Code 255	Character Code 252	Character Code 1
IAC	Won't	Echo

[1] Escape Code
[2] Interpret As Command
[3] معادل **Echo On**

با ارسال این سه کد از سرویس‌دهنده خواسته شده است که پس از دریافت هر کاراکتر آنرا برگشت ندهد.[۱]

Do : به معنای پذیرش تقاضای داده شده یا اعلام یک عمل انجام شده می‌باشد.[۲]

Don't : به معنای عدم پذیرش تقاضای داده شده یا اعلام لغو یک عمل می‌باشد.[۳]

- Interrupt Process (IP) : بسیاری از سیستمها به کاربر اجازه می‌دهند که کاربر بتواند برنامهٔ در حال اجرای خود را متوقف نماید. با ارسال این کد ، سرویس‌دهنده می‌تواند پروسهٔ در حال اجرای کاربر را متوقف نماید.
- Abort Output (AO) : فرض کنید کاربر برنامه‌ای را اجرا کرده که ضمن اجرا ، تولید خروجی می‌نماید. به عنوان مثال برنامه‌ای ضمن مرتب کردن رکوردهای درون یک فایل ، آنها را روی صفحه نمایش نشان می‌دهد. در این حالت کاربر می‌تواند با ارسال این کدها از سرویس‌دهنده بخواهد ، در حالی که اجرای برنامه را ادامه می‌دهد ، نتیجهٔ خروجی صفحه نمایش را برای او ارسال ننماید.
- Break (BRK) : ارسال این کد دقیقاً معادل آن است که شما در هنگام اجرای یک پروسه کلیدهای Ctrl+Break را فشاربدهید.
- Are You There (AYT) : فرض کنید که یک سرویس‌دهنده به دلایلی نظیر بار زیاد ، برای مدت طولانی سرویس‌دهی به کاربر را به تعویق بیاندازد. برنامهٔ TelNet با ارسال این کد می‌تواند مطمئن شود که آیا عدم پاسخگویی ناشی از بار زیاد است یا آنکه یک مشکل حادّ نظیر قطع ارتباط یا خرابی سرویس‌دهنده وجود دارد.
- Erase Line (EL) : با ارسال این کد از سرویس‌دهنده می‌خواهد که خط ارسال شدهٔ قبلی را از بافر ورودی خود حذف نماید.
- Erase Character (EC) : با ارسال این کد از سرویس‌دهنده می‌خواهد که کاراکتر ارسال شدهٔ قبلی را از بافر ورودی خود حذف نماید. (همانند عملی که کد Backspace انجام می‌دهد.

اگر با دستور toggle option ، حالت مشاهدهٔ فرامین را فعال کنید ممکن است پیغامهایی نظیر مثال زیر را ببینید :

RCVD will ECHO

---

۱ معادل **Echo Off**
۲ معادل **Positive Acknowledgement**
۳ معادل **Negative Acknowledgement**

یعنی تقاضای فعال شدن حالت ECHO از طرف مقابل دریافت شد.

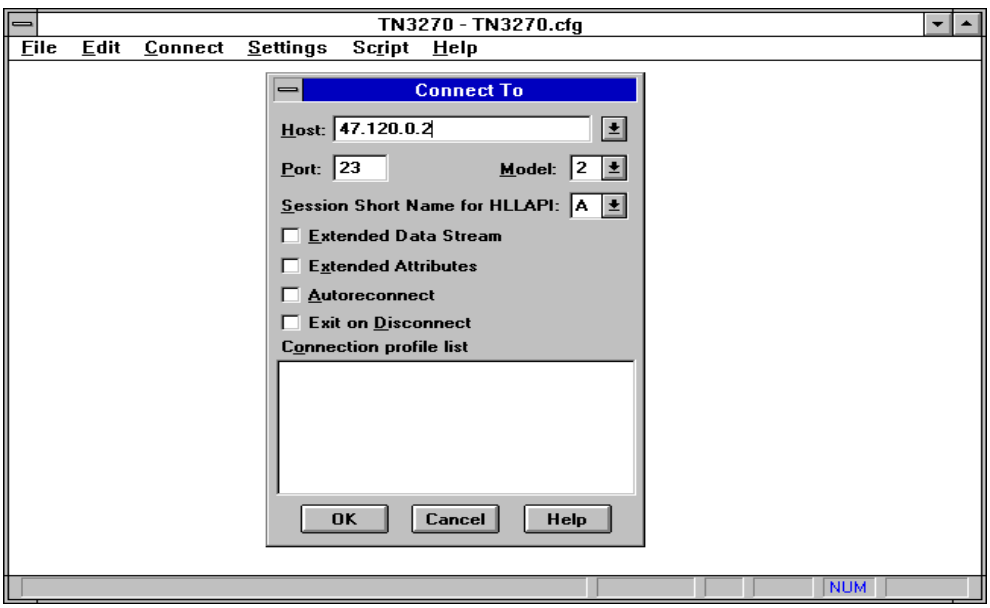
SEND do ECHO

یعنی تقاضای فعال شدن حالت ECHO ، پذیرفته و اعمال شد.

برای توضیح بیشتر در مورد عملکرد این کدها به مراجع معرفی شده در آخر این فصل مراجعه کنید.

TN3270 (۳–۱

اغلب کامپیوترهای چندکاربرهٔ بزرگ[۱] از کدهای EBCDIC استفاده می‌نمایند در حالی که ریزکامپیوترها معمولاً از استاندارد ASCII بهره برده‌اند. بنابراین برای ورود به یک سیستم راه دور باید تبدیل کد انجام شود به همین دلیل برنامهٔ TelNet با قابلیت تغییر مجموعهٔ کد با نام TN3270 ارائه شد. در شکل (۳–۸) برنامهٔ TN3270 از مجموعهٔ نرم‌افزار NetManage ChameleonNFS نشان داده شده است که از طریق آن سعی شده است با یک ماشین با آدرس 47.120.0.2 و استاندارد EBCDIC ارتباط برقرار شود.



شکل (۳–۸) برنامهٔ TN3270 از مجموعهٔ نرم‌افزار NetManage ChameleonNFS

[۱] Multiuser Mainframe Computer

نام کد	مقدار	توضیح
Abort Output (AO)	245	Runs process to completion but does not send the output
Are you there (AYT)	246	Queries the other end to ensure that an application is functioning
Break (BRK)	243	Sends a break instruction
Data Mark	242	Data portion of a Sync
Do	253	Asks for the other end to perform or an acknowledgment that the other end is to perform
Don't	254	Demands that the other end stop performing or confirms that the other end is no longer performing
Erase Character (EC)	247	Erases a character in the output stream
Erase Line (EL)	248	Erases a line in the output stream
Go Ahead (GA)	249	Indicates permission to proceed when using half-duplex (no echo) communications
Interpret as Command (IAC)	255	Interprets the following as a command
Interrupt Process (IP)	244	Interrupts, suspends, aborts, or terminates the process
NOP	241	No operation
SB	250	Subnegotiation of an option
SE	240	End of the subnegotiation
Will	251	Instructs the other end to begin performing or confirms that this end is now performing
Won't	252	Refuses to perform or rejects the other end performing

جدول (٤-٨) برخی از کدهای فرمان در TelNet

نام کد	توضیح
0	Binary transmission
1	Echo
2	Reconnection
3	Suppress Go Ahead (GA)
4	Approximate message size negotiation
5	Status
6	Timing mark
7	Remote controlled transmission and echo
8	Output line width
9	Output page length
10	Output carriage-return action
11	Output horizontal tab stop setting
12	Output horizontal tab stop action
13	Output form feed action
14	Output vertical tab stop setting
15	Output vertical tab stop action
16	Output line feed action
17	Extended ASCII characters
18	Logout
19	Bytes macro
20	Data entry terminal
21	SUPDUP
22	SUPDUP output
23	Send location
24	Terminal type
25	End of Record
26	TACACS user identification
27	Output marking
28	Terminal location number
29	3270 regime
30	X.3 PAD (Packet assembly and disassembly)
31	Window size

جدول (۵–۸) کدهای اعمال اختیاری

۳) پروتکل انتقال فایل (FTP)

پروتکل انتقال فایل که از این به بعد آن را FTP می‌نامیم ابزاریست مطمئن برای انتقال فایل بین کامپیوترهائی که به شبکهٔ اینترنت متصل هستند. خدماتی که این پروتکل ارائه می‌کند عبارتند از:

- تهیهٔ لیستی از فایلهای موجود از سیستم فایل کامپیوتر راه دور
- حذف، تغییر نام و جابجا کردن فایلهای کامپیوتر راه دور
- جستجو در شاخه‌های (دایرکتوریهای) کامپیوتر راه دور
- ایجاد یاحذف شاخه روی کامپیوتر راه دور
- انتقال فایل از کامپیوتر راه دور به کامپیوتر میزبان[۱]
- انتقال فایل و ذخیرهٔ آن از کامپیوتر میزبان به کامپیوتر راه دور[۲]

قابلیتهائی که پروتکل FTP عرضه می‌کند می‌تواند برای سیستم سرویس دهنده بسیار خطرناک باشد چرا که بسادگی می‌توان فایلهای یک کامپیوتر راه دور را آلوده یا نابود کرد. فلذا در این پروتکل کاربران باید قبل از تقاضای هر سرویسی کلمهٔ عبور خود را وارد نمایند و سرویس دهنده پس از شناسائی کاربر، سطح دسترسی و عملیات مجاز برای کاربر را تعیین می‌کند و یک نشست[۳] FTP آغاز می‌شود.

FTP این قابلیت را ندارد که بتوان همانند پروتکل Telnet برنامه‌ای را بر روی ماشین راه دور اجرا کرد بلکه فقط روشی سریع، ساده و مطمئن برای خدمات فایل به کاربران راه دور محسوب می‌شود. حال باید ارتباط بین سرویس دهنده و سرویس گیرندهٔ FTP را تشریح نماییم:

در پروتکل FTP برای شروع یک "نشست" بین برنامه سرویس دهنده و برنامهٔ سرویس گیرنده باید دو ارتباط همزمان از نوع TCP برقرار شود. به هر یک از این ارتباطات در ادبیات پروتکل FTP، "کانا ل" گفته می‌شود. این دو کانال عبارتند از :

- کانا ل داده: یک ارتباط TCP با پورت شماره ۲۰ از سرویس دهنده که روی آن داده‌ها (مثلاً بلوکهای یک فایل) مبادله می‌شوند.
- کانال فرمان: یک ارتباط TCP با پورت شماره ۲۱ که روی آن فرامین لازم برای مدیریت فایلها رد و بدل می‌شوند.

---

۱ Downloading
۲ Uploading
۳ Session

دلیل لزوم برقراری دو کانال مجزا بین سرویس‌دهنده و سرویس‌گیرنده آن است که بتوان بدون قطع جریان داده‌ها فرامین را بطور همزمان مبادله کرد. بعنوان مثال در حین انتقال یک فایل می‌توان روی کانال فرمان، دستور لغو عمل انتقال یا تغییر مود انتقال را صادر کرد.
ذکر این نکته ضروری است که در پروتکل FTP همهٔ عملیات انتقال فایل در "پیش زمینه"[۱] انجام می‌شود بدین معنا که پروتکل FTP از سیستم Spooler یا صف برای انتقال فایلها استفاده نمی‌کند بلکه عملیات انتقال به صورت بلادرنگ انجام می‌گیرد. (سیستمهائی مثل مدیریت چاپ در "پس زمینه"[۲] عمل می‌کند یعنی وقتی پروسه‌ای تقاضای چاپ یک سند را می‌دهد سیستم عامل آنرا به صف می‌کند تا در موقع مناسب آن را چاپ نماید فلذا مشخص نیست از زمان صدور فرمان چاپ چه مدت طول بکشد تا سند چاپ شود چرا که اولویت با پروسه هائی است که در پیش زمینه اجرا میشوند.)

بگونه‌ای که اشاره شد سرویس دهندهٔ FTP بایستی دو پروسه همزمان ایجاد نماید که یکی وظیفهٔ مدیریت ارتباط روی کانال فرمان را به عهده داشته و اصطلاحا "مفسر پروتکل" یا پروسه PI[۳] نامیده می‌شود. وظیفهٔ پروسهٔ دیگر مدیریت انتقال داده‌ها است و به DTP[٤] یا "پروسه انتقال داده" معروف است. پروسهٔ PI همیشه به پورت شماره۲۱ گوش می‌دهد و پروسه DTP به پورت شماره ۲۰ مقید شده است.

۱-۳) روشهای برقراری یک نشست FTP

برقراری ارتباط بین سرویس‌دهنده و سرویس گیرندهٔ FTP با دو روش امکان‌پذیر است:

- روش معمولی یا Normal Mode
- روش غیرفعال یا Passive Mode

در روش معمولی برای برقراری یک نشست FTP مراحل زیر انجام می‌شود:

الف) در برنامه سمت سرویس گیرنده (برنامه سمت مشتری) ابتدا دو سوکت نوع TCP با شماره پورت تصادفی بالای ۱۰۲٤ ایجاد می‌شود.

ب) در مرحله دوم برنامهٔ سمت مشتری سعی می‌کند با استفاده از دستور ()connect ارتباط یکی از سوکتهای ایجاد شده را با پورت شماره ۲۱ از سرویس‌دهنده برقرار نماید. اگر این

---

[۱] Foreground
[۲] Background
[۳] Protocol Interpreter
[٤] Data Transfer Protocol

ارتباط برقرار شود در حقیقت کانال فرمان باز شده و پروسه PI آمادهٔ تفسیر فرامین صادره از سمت مشتری می‌باشد.

ج) برنامهٔ سمت مشتری با فرمان "PORT" به برنامهٔ سمت سرویس دهنده شمارهٔ پورت سوکت دوم را اعلام می‌نماید و منتظر می‌ماند. (در حقیقت برنامه مشتری روی سوکت دوم عمل ()listen انجام می‌دهد)

د) در ادامه برنامهٔ سرویس دهنده سعی می‌کند یک ارتباط با TCP با شماره پورت اعلام شده برقرار نماید. یکی از نکات عجیب در این پروتکل آنست که سرویس دهنده FTP موظف است اقدام به برقراری یک ارتباط TCP از طریق دستور ()connect با برنامه مشتری نماید در صورتی که معمولاً سرویس دهنده پذیرندهٔ ارتباط است نه شروع کنندهٔ ارتباط.

ه) برنامهٔ سمت مشتری ارتباط TCP شروع شده از سرویس دهنده را تصدیق کرده و یک نشست FTP آغاز می‌شود.

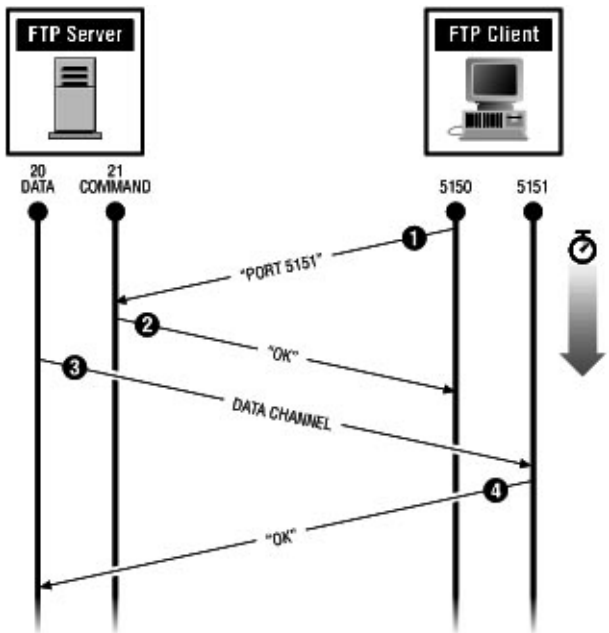
در شکل (٦–٨) با یک مثال مراحل فوق به تصویر کشیده شده است. ابتدا برنامهٔ سمت مشتری دو سوکت باز کرده و شمارهٔ پورتهای ۵۱۵۰ و ۵۱۵۱ را به آنها نسبت داده است. (با دستور ()bind) سپس از طریق سوکت اول یک ارتباط TCP با پورت ۲۱ از سرویس‌دهنده برقرار کرده و پس از برقراری ارتباط با ارسال فرمان "PORT 5151" شمارهٔ پورت سوکت دوم خود را اعلام کرده است. سمت سرویس دهنده ضمن تصدیق پذیرش یک نشست بلافاصله اقدام به برقراری یک ارتباط TCP بین پورت شمارهٔ ۲۰ خودش و پورت شمارهٔ ۵۱۵۱ از سرویس گیرنده نموده است. با تصدیق این ارتباط نشست FTP آغاز می‌شود.

حال باید روش "غیرفعال" را در برقراری یک نشست FTP بررسی نمائیم:

الف) در برنامهٔ سمت مشتری ابتدا دو سوکت نوع TCP با شماره پورت تصادفی بالای ١٠٢٤ ایجاد می‌شود.

ب) برنامهٔ سمت مشتری سعی می‌کند ارتباط TCP یکی از سوکتهای ایجاد شده را با پورت شمارهٔ ۲۱ از سرویس دهنده برقرار نماید. با برقراری این ارتباط کانال فرمان باز شده و پروسه PI آمادهٔ تفسیر فرامین صادره از سمت مشتری خواهد شد.

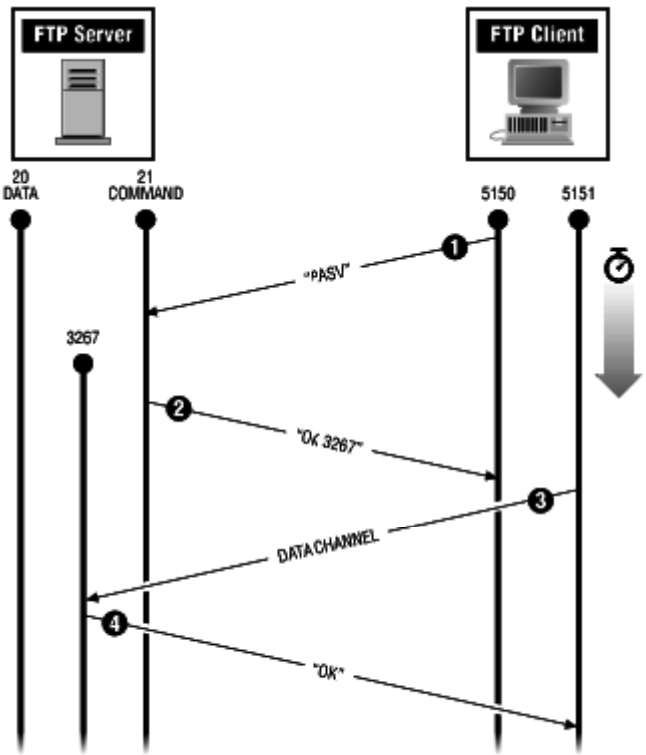
ج) برنامهٔ سمت مشتری با فرمان PASV به برنامهٔ سمت سرویس دهنده اعلام می‌کند که خواستار یک نشست از نوع غیر فعال است.



شکل (٦–٨) مثالی از یک نشست FTP به روش معمولی

د) برنامهٔ سمت سرویس دهنده یک سوکت با شماره پورت تصادفی (بالای ١٠٢٤) ایجاد کرده و شمارهٔ آنرا به برنامهٔ مشتری اعلام می‌نماید.
ه) برنامهٔ سمت مشتری ارتباط سوکت دوم خود را با شماره پورت اعلام شده برقرار کرده پس از تصدیق ارتباط نشست FTP آغاز می‌شود.

در شکل (٧–٨) مثالی از برقراری نشست غیرفعال تصویر شده است: ابتدا برنامهٔ سمت مشتری دو سوکت باز کرده و شماره پورتهای ٥١٥٠ و ٥١٥١ را به آنها نسبت داده است. سپس از طریق سوکت اول یک ارتباط TCP با پورت ٢١ از سرویس دهنده برقرار شده و با ارسال فرمان "PASV" منتظر تصدیق و اعلام شمارهٔ پورت از سرویس دهنده می‌ماند. در این مثال سرویس دهنده ضمن اعلام شماره پورت ٣٢٦٧ گشوده شدن کانال فرمان را تصدیق می‌کند.



شکل (۷–۸) مثالی از یک نشست FTP به روش غیرفعال

مجدداً برنامهٔ سمت مشتری ارتباط سوکت دوم خود را با شماره پورت ۳۲٦۷ از سرویس دهنده برقرار نموده و در صورت تصدیق ارتباط ، نشست FTP آغاز می‌شود.

در دنیای یونیکس برنامه سمت سرویس دهنده بنام ftpd[1] و برنامه سمت مشتری بنام ftp مشهور است. دقت کنید که برخی از سرویس دهنده‌ها از روش غیر فعال حمایت نمی‌کنند و فقط روش معمولی را می‌پذیرند.

در برنامهٔ ftp دو نوع فرمان تعریف شده که گاهاً با هم اشتباه می‌شوند. نوع اول فرامینی هستند که روی کانال فرمان یعنی بین سرویس دهنده و سرویس گیرنده رد و بدل می‌شود. این

[1] FTP Daemon

فرامین که در بخش بعدی معرفی می‌شوند فرامین داخلی نام دارند. نوع دوم فرامینی هستند که بین کاربر و برنامه ftp تعریف شده و کاربر برای بکارگیری برنامه ftp باید آنها را بداند. (همانند فرامین DOS)

۴) فرامین داخلی[۱] FTP

اگر به مراحل برقراری نشست FTP دقت کنید متوجه خواهید شد که در هر دو روش ابتدا کانال فرمان برقرار می‌شود و تا پایان نشست این کانال برقرار خواهد ماند. در این بخش بررسی می‌کنیم که روی این کانال چه فرامینی مبادله می‌شود.

فرامین داخلی ftp تماماً حالت متنی داشته و حداکثر چهار حرفی هستند. برخی از فرامین دارای پارامتر هستند که این پارامترها نیز بصورت متنی پس از یک فاصلهٔ خالی در ادامهٔ فرمان قرار می‌گیرند. خاتمهٔ فرمان با کدهای \r[۲] و \n[۳] مشخص شده است. مثلاً:

"PASV\r\n"

اولین مزیت بهره‌گیری از فرامین متنی آنست که کاربر می‌تواند سلسلهٔ فرامین را دیده و به آسانی بفهمد و این مزیت به یک کاربر حرفه‌ای تا حد زیادی به درک روند عملیات و اشکالزدایی کمک خواهد کرد. دومین مزیت آنست که یک متخصص می‌تواند براحتی (بدون نیاز به برنامهٔ ftp و با ابزاری مثل Telnet یا برنامه نویسی سوکت) مستقیماً با سرویس دهنده ارتباط برقرار کند.

مجموعه فرامین ftp با توضیح مختصر در جدول (۸-۸) آمده است. بگونه‌ای که مشاهده می‌شود این فرامین برای سه دسته عملیات در نظر گرفته شده‌اند:

- فرآیند برقراری و ختم یک نشست ftp (مثل فرمان PASV ، PORT ، QUIT)
- ارسال کلمهٔ عبور جهت تعیین جواز و سطح سرویس‌دهی
- فرامین مربوط به انتقال فایل، فهرست شاخه‌ها و عملیات مدیریت فایلها

در پاسخ به هر فرمان صادره یک کد سه رقمی برمی‌گردد تا وضعیت اجرای فرمان را مشخص نماید. برنامهٔ سمت مشتری با پردازش این کد می‌تواند در مورد عمل بعدی خود تصمیم بگیرید. (تصمیماتی مثل لغو فرمان، تلاش مجدد، صدور فرمان جایگزین یا ختم نشست)

---

۱ FTP Internal Commands
۲ Carriage Return
۳ New Line

فرمان	عملکرد فرمان (بسیاری از فرامین با پارامتری که در جلو فرمان درج شده معنا می‌یابند)
ABOR	تقاضای لغو و ناتمام رها کردن فرمان قبلی
ACCT	اعلام مشخصه کاربری
ALLO	تقاضای تخصیص حافظه و فضا برای عمل بعدی
APPE	تقاضای ضمیمه شدن داده های ارسالی به یک فایل موجود
CDUP	تقاضای تغییر زیرشاخه جاری به شاخه پدری آن (معادل دستور cd.. در dos)
CWD	تقاضای تغییر شاخه جاری به شاخه ای که نامش در جلو فرمان درج شده است
DELE	تقاضای حذف فایلی که نامش در جلو فرمان درج شده است
HELP	تقاضای راهنمایی و اخذ اطلاعات مفید در مورد فرامین
LIST	تقاضای انتقال فهرست شاخه ها
MKD	تقاضای ایجاد یک زیرشاخه در شاخه فعلی
MODE	تنظیم روش انتقال (متنی و دودویی)
NLST	تقاضای فهرست گیری از شاخه فعلی
NOOP	هیچ عملی انجام نمیدهد
PASS	ارسال کلمه عبور کاربر
PASV	تقاضا برای برقراری یک نشست غیر فعال
PORT	اعلام شماره پورت TCP برای برقراری کانال داده
PWD	تقاضای اعلام فهرست فایلهای شاخه جاری
QUIT	تقاضای ختم نشست FTP
REIN	تقاضای ختم نشست فعلی و برقراری یک نشست جدید
REST	تقاضای انتقال مجدد داده های فایل از ابتدای آن
RETR	تقاضای دریافت نسخه ای از یک فایل
RMD	تقاضای حذف یک شاخه
RNFR	نام قدیم یک فایل یا مسیری که باید تغییر نام بدهد

فرمان	عملکرد فرمان (بسیاری از فرامین با پارامتری که در جلو فرمان درج شده معنا می‌یابند)
RNTO	نام جدید یک فایل یا مسیری که باید تغییر نام بدهد
SITE	تقاضای مشخصات سرویسی که سرویس دهنده ارائه میکند
STAT	تقاضای وضعیت فعلی سرویس دهنده
STOR	تقاضای پذیرش و ذخیره داده های یک فایل از سرویس دهنده
STOU	تقاضای پذیرش و ذخیره داده های یک فایل از سرویس دهنده تحت نامی متفاوت
STRU	تقاضای تعیین ساختار یک فایل
SYST	تقاضای تعیین نوع سیستم عامل سرویس دهنده
TYPE	تعیین نوع داده ها
USER	اعلام کد کاربری به سرویس دهنده

جدول (۸-۸) فرامین داخلی پروتکل FTP

هریک از ارقام صدگان، دهگان و یکان کد بازگشتی معنای خاصی دارد. بعنوان مثال اگر رقم صدگان کد ۱ یا ۲ یا ۳ باشد فرمان ارسالی موفقیت‌آمیز بوده ولی اگر ٤ یا ٥ باشد بمعنای بروز خطا تلقی می‌شود.

در جدول (۹-۸) و (۱۰-۸) معانی متفاوت رقم صدگان و دهگان کد سه رقمی بازگشتی توسط سرویس دهنده را معرفی شده است. معانی رقم سوم (یکان) از کد بازگشتی در اینجا نیامده زیرا تعداد آنها زیاد و معانی آنها در سیستمهای متفاوت با هم فرق می‌کنند.

مقدار رقم	معنا
۱	فرمان درخواستی شروع به اجرا شد. منتظر کدهای برگشتی بعدی باشید.
۲	فرمان کاملا اجرا شد. ارسال فرمان بعدی مجاز است.
۳	فرمان شناخته شد ولی فعلا به دلیل کمبود اطلاعات لازم برای اجرا معلق مانده است
٤	فرمان پذیرفته نشد ولی صدور مجدد آن بلامانع می‌باشد.
٥	فرمان پذیرفته نشد وصدور مجدد آن بی فایده خواهد بود.

جدول (۹-۸) معانی رقم اول از کد بازگشتی در پروتکل FTP

مقدار رقم	معنای رقم (معنای این رقم با در نظر گرفتن معنای رقم صدگان کد بازگشتی تکمیل میشود)
۰	فرمان صادره بی معنی بوده و تشخیص داده نشده است
۱	به معنای آنکه کد بازگشتی کدی است در پاسخ به تقاضای اطلاعات
۲	به معنای آنکه کد بازگشتی کدی است در ارتباط با مدیریت نشست
۳	به معنای آنکه کد بازگشتی کدی است در ارتباط با احراز هویت کاربر
٤	بلا استفاده
٥	به معنای آنکه کد بازگشتی کدی است در ارتباط با وضعیت سرویس دهنده

جدول (۱۰-۸) معانی رقم دوم از کد بازگشتی در پروتکل FTP

در پروتکل FTP ، انتقال فایل می‌تواند در چند حالت متفاوت انجام شود. اکثر سیستمهای عامل (مثل یونیکس) فقط دارای دو روش انتقال هستند: روش متنی[۱] و روش دودوئی[۲]
هر نوع فایل اعم از فایلهای اجرائی، صدا، تصویر و حتی فایلهای متنی را می‌توان به حالت دودوئی انتقال داد و سرعت انتقال در این روش رضایت‌بخش است. با حالت متنی فقط فایلهای متن (با کدها ASCII) قابل مبادله هستند. فایلهای متنی مجموعه‌ای از کاراکترهای ASCII بهمراه برخی از کدهای کنترلی همانند "\n" ، "\t" هستند که می‌توان با دستورات معمولی از این نوع فایلها خواند و در آن نوشت. معمولاً تمام سرویس دهنده‌ها حالت پیش‌فرض یک نشست را حالت دودوئی فرض می‌کنند. کاربر باید قبل از انتقال فایل اطمینان حاصل کند که از یک مود انتقال صحیح استفاده می‌نماید چرا که مثلاً انتقال یک فایل تصویر در مود متنی نتیجهٔ مخرب خواهد داشت.

۵) فرامین کاربری برنامه FTP

بغیراز فرامین داخلی پروتکل FTP که روی کانال فرمان مبادله می‌شوند یکسری فرمان کاربری وجود دارد که کاربر برای استفاده از ftp باید از آنها اطلاع داشته باشد. برنامه‌های ftp در سمت کاربر معمولاً بر دو نوع هستند:
الف) برنامه‌هائی که در خط فرمان اجرا می‌شوند و کاربر مجبوراست همانند محیط DOS یا یونیکس فرامین استفاده از برنامه ftp را از حفظ باشد.

---

[۱] Text Mode
[۲] Binary Mode

ب) برنامه‌هائی که دارای ظاهر گرافیکی هستند (برنامه های GUI) و کاربر از طریق آیکون یا منوها فرمان مورد نظر خود را اجرا می‌نماید.

برای بکارگیری برنامه ftp در خط فرمان لازم است ضمن اجرای برنامه، نام ماشینی که سرویس ftp می‌دهد، بعنوان پارامتر ورودی برنامه مشخص شود:

ftp rtfm.mit.edu

یا مثلاً **ftp** tpci_hpws4

نامی که در جلوی اسم برنامهٔ ftp نوشته می‌شود باید قابل تحلیل[۱] و ترجمه به یک آدرس IP باشد. اگر آدرس IP ماشین سرویس دهنده را می‌دانید می‌توانید مستقیماً آدرس IP را در جلوی نام برنامهٔ ftp وارد نمائید.

مثال: **ftp** 205.150.89.5

وقتیکه ارتباط بین برنامهٔ کاربر و سرویس‌دهندهٔ ftp برقرار شد باید اعتبار کاربر برای سرویس دهنده مشخص شود تا امکانات لازم را در اختیار او قرار بدهد. در حقیقت وقتی بعنوان یک کاربر معتبر شناخته شدید به شما اجازهٔ ورود به سیستم ftp داده خواهد شد؛ به این عمل اصطلاحاً Login گفته می‌شود. پروسه‌ای که اعتبار کاربران را از لحاظ مجوز ورود و سطح دسترسی، کنترل می‌نماید "پروسه ورود"[۲] نامیده می‌شود. برای اجازه ورود به سیستم کاربر به یک کلمهٔ شناسائی و رمز عبور احتیاج دارد که این دو را مدیر شبکه در اختیار او می‌گذارد. البته بسیاری از سرویس دهنده‌های ftp در دنیا موجودند که اجازه دسترسی آزادانه به فایلها را (فقط برای خواندن و دریافت فایلها) را به کاربران می‌دهند. به اینگونه سرویس دهنده‌ها "ftp بی‌نام"[۳] گفته می‌شود و شما می‌توانید بجای کلمه شناسائی کلمهٔ anonymous یا کلمه guest را تایپ کنید.

در قطعه کد زیر مثالی از مراحل یک نشست ftp برای برقراری ارتباط با سرویس دهنده‌ای بنام tpci_hpws4 ارائه شده است: (خطوط پر رنگ از طرف سرویس دهنده صادر شده و در ابتدای خط شماره کد بازگشتی مشخص می‌باشد)

```
ftp tpci_hpws4
Connected to tpci_hpws4.
220 tpci_hpws4 FTP server
Name (tpci_hpws4:tparker):
```

---

[۱] Resolvable
[۲] Login Process
[۳] Anonymous FTP

331 Password required for tparker.

Password: *******

230 User tparker logged in.

Remote system type is UNIX.

Using binary mode to transfer files.

برخلاف برنامهٔ Telnet بعد از برقراری ارتباط با ماشین سرویس‌دهندهٔ FTP شما واقعاً روی ماشین راه دور نیستند و عملا با پردازنده و سیستم عامل خود کار می‌کنید. از دیدگاه کاربر، فرآیندی که برنامهٔ ftp برای برقراری یک نشست دنبال می‌کند به شرح ذیل است:

الف) عمل Login : شامل تعیین کلمهٔ شناسائی و رمز عبور و تائید آنها توسط سرویس دهنده

ب) Define Directory : تعیین شاخه‌ای که پس از برقراری ارتباط بصورت پیش فرض برای شروع عملیات کاربر در نظر گرفته می‌شود.

ج) Define file Transfer Mode : تعیین روش انتقال (حالت متنی یا دودوئی)

د) Start data transfer : امکان صدور فرمان را برای کاربر فراهم می‌آورد.

ه) Stop data transfer : ارتباط را خاتمه می‌دهد.

این مراحل چندگانه به ترتیب برای هر نشست انجام می‌شوند.

برای بهره‌گیری کاربر از خدمات سیستم فایل در پروتکل ftp فرامینی تعریف شده‌اند که در جدول (۱۱–۸) معرفی شده‌اند.

فرامین کاربری	معنای فرمان
Ascii	تنظیم حالت انتقال فایل به حالت متنی
Binary	تنظیم حالت انتقال فایل به حالت دودویی
Cd	تغییر شاخه جاری به شاخه جدید بر روی سرویس دهنده
Close	ختم نشست
Del	حذف یک فایل از روی سرویس دهنده
Dir	فهرست گیری از شاخه جاری سرویس دهنده
Get	تقاضای انتقال یک فایل از سرویس دهنده
Hash	هرگاه یک بلوک داده از یک فایل در حال انتقال سالم رسید علامت ویژه ای را نشان بدهد

Help	راهنمایی
Lcd	تقاضای تغییر شاخه جاری بر روی ماشین محلی کاربر
Mget	دریافت چندین فایل از روی سرویس دهنده
Mput	ارسال چندین فایل بر روی سرویس دهنده
Open	تقاضای برقراری یک نشست و وصل به یک سرویس دهنده
Put	ارسال یک فایل بر روی سرویس دهنده
Pwd	نمایش شاخه جاری از سرویس دهنده
Qoute	ارسال مستقیم یکی از فرامین داخلی
Quit	تقاضای ختم نشست

جدول (۱۱-۸) فرامین کاربری پروتکل FTP

یک مثال از نشست ftp در زیر آمده است: به کدهائی که در پاسخ به هر فرمان از طرف سرویس‌دهنده صادر شده دقت کرده با جدول (۹–۸) و (۱۰–۸) مقایسه نمایند.

tpci_hpws1-1> ftp tpci_hpws4

Connected to tpci_hpws4.

220 tpci_hpws4 FTP server (Version 1.7.109.2 Tue Jul 28 23:32:34 GMT 1992) ready.

Name (tpci_hpws4:tparker):

331 Password required for tparker.

Password: *******

230 User tparker logged in.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> pwd

257 "/u1/tparker" is current directory.

ftp> get mandelfile1.gif

remote: mandelfile1.gif local: mandelfi.gif

200 PORT command successful

150 Opening BINARY mode data connection for mandelfile1.gif

226 File transfer complete

1192834 bytes sent in 0.89 seconds

ftp> <Ctrl+d>

tpci_hpws1-2>

در مثال بالا فایلی بنام mandelfile1.gif از یک ماشین سرویس‌دهنده با سیستم عامل یونیکس به ماشین میزبان با سیستم عامل DOS منتقل شده است. دقت کنید که چون در سیستم عامل DOS نام فایل حداکثر هشت کاراکتر است به همین دلیل اسم فایل بصورت خودکار کوتاه شده است. در اینجا از حالت دودوئی برای انتقال فایل استفاده شده که پیش فرض سیستم بوده و احتیاجی به تنظیم ندارد.

با اضافه کردن گزینه d- در هنگام فراخوانی و اجرای برنامه می‌توان عمل اشکال زدائی انجام داد. با این گزینه تمام فرامین داخلی که بین برنامه FTP و سرویس دهنده مبادله میشوند قابل مشاهده خواهند بود.در مثال بالا پیغامهایی که از طرف سرویس دهنده ارسال شده همگی با یک کد سه رقمی ظاهر شده‌اند و بقیه پیغامها از طرف برنامهٔ ftp می‌باشد.

در مثال سادهٔ زیر عملیات نشست ftp با امکان اشکال‌زدایی آورده شده است:

tpci_hpws1-1> ftp -d

ftp> open tpci_hpws4

Connected to tpci_hpws4.

220 tpci_hpws4 FTP server Name (tpci_hpws4:tparker):

---> USER tparker

331 Password required for tparker.

Password:

---> PASS qwerty5

230 User tparker logged in.

---> SYST

215 UNIX Type: L8

Remote system type is UNIX.

---> Type I

200 Type set to I.

Using binary mode to transfer files.

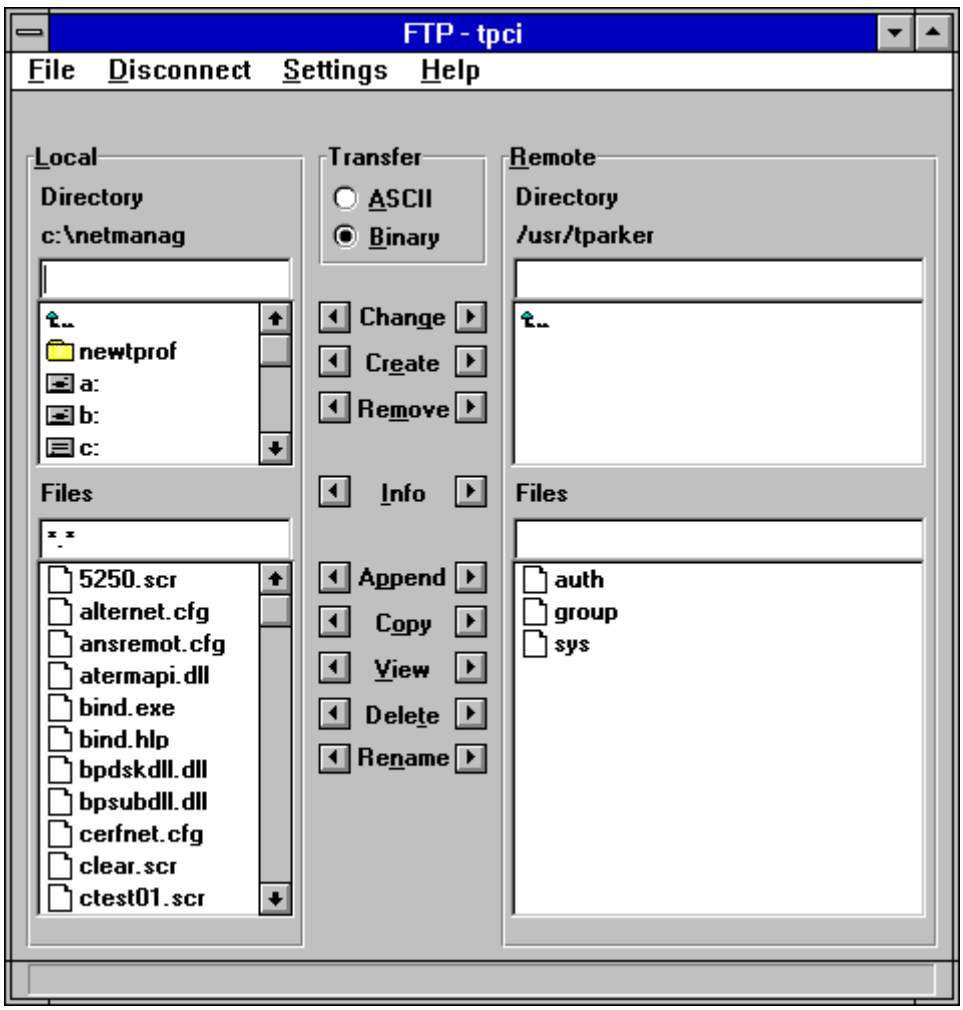
```
ftp> ls
---> PORT 47,80,10,28,4,175
200 PORT command successful.
---> TYPE A
200 Type set to A.
---> LIST
150 Opening ASCII mode data connection for /bin/ls.
total 4
-rw-r-----  1 tparker  tpci    2803  Apr 29 10:46 file1
-rw-rw-r--  1 tparker  tpci    1286  Apr 14 10:46 file5_draft
-rwxr-----  2 tparker  tpci   15635  Mar 14 23:23 test_comp_1
-rw-r-----  1 tparker  tpci      52  Apr 22 12:19 xyzzy
Transfer complete.
---> TYPE I
200 Type set to I.
ftp> <Ctrl+d>
tpci_hpws1-2>
```

دقت کنید که در مثال بالا هنگام دریافت لیست شاخه‌ها[1]، چگونه حالت باینری به متنی تبدیل شده و سپس دوباره به مود دودویی (مقدار پیش‌فرض سیستم) برمی‌گردد.

وقتیکه از یک سیستم عامل گرافیکی مثل MS-Windows استفاده می‌شود میتوان از یک ابزار مبتنی بر GUI[2] استفاده کرد. برای مثال نرم افزار NetManage's ChameleonNFS یا CuteFTP برنامه های کمکی FTP هستند. در شکل (۱۲–۸) نرم افزار FS client تحت Windows با یک سرویس دهنده FTP تحت یونیکس نشست برقرار کرده است. طرف Local (سمت چپ) پنجرهٔ مربوط به ماشین Windows را نشان میدهد و طرف Remote (سمت راست) پنجرهٔ مربوط به محتویات سیستم فایل جاری یونیکس را نشان می‌دهد. وقتی که از یک برنامه کمکی

[1] Directory listing
[2] Graphic User Interface

GUI مثل این برنامه استفاده می‌شود، می‌توان از ماوس و کلیدهای گوناگون برای انتقال فایل بین دو ماشین ، استفاده کرد.

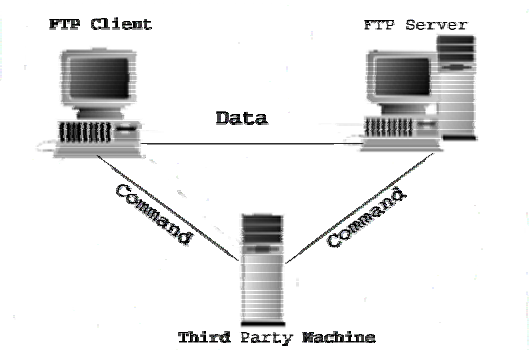


شکل (۱۲–۸) مثالی از یک نرم افزار مشتری FTP با ظاهر گرافیکی

۶) انتقال با واسطه در پروتکل FTP[۱]

FTP این امکان را فراهم می‌سازد که انتقال فایلها از طریق ماشین سومی که بین برنامه مشتری و سرویس دهنده قرار گرفته است، انجام شود. این رویه به عنوان "انتقال باواسطه" شناخته شده است و اجازه دسترسی به سیستم فایلِ ماشین سرویس دهنده و نظارت بر فرامین صادره از طریق این ماشین انجام می‌شود. استفاده از یک سیستم واسطه برای نظارت بر فرامین صادره از طرف کاربران و بررسی مجوزها و سطوح دسترسی هر کاربر باعث میشود که حجم پردازش روی سرویس دهنده FTP کاهش یافته ، سرعت انتقال افزایش داشته باشد. شکل (۱۳-۸) نمایش کلی یک انتقال باواسطه را همراه با کانالهای ایجاد شده در ماشین واسط، نشان می‌دهد.

در این روش هرگاه یک نشست برگزار شود، کانال فرمان حتما از طریق ماشین واسطه بین ماشین مشتری و ماشین سرویس دهنده برقرار میشود در حالیکه کانال داده مستقیماً بین دو ماشین برقرار می‌شود. وقتی یک فرمان صادر شود ، چون از کانال ماشین واسطه عبور میکند ، مجوزها بررسی می‌شود و در صورت مجاز بودن فرمان ، درخواست به سوی سرویس دهنده ارسال میشود. انتقال داده ها مستقیماً صورت می‌گیرد، چرا که مجوزها قبلا بررسی شده اند.



شکل (۱۳-۸) انتقال با واسطه در پروتکل FTP

[۱] Third Party Transfer

۱-۶) دسترسی بی‌نام به FTP

FTP برای توانا ساختن قابلیتهای انتقال فایل نیاز به کد کاربری و کلمه عبور کاربر دارد. روش بسیار راحت‌تری برای ایجاد دسترسی عمومی به یک مجموعه فایل یا دایرکتوری وجود دارد که FTP بی‌نام خوانده می‌شود. با FTP بی‌نام، دیگر نیازی به داشتن مجوز روی یک ماشین نیست. FTP این کار را معمولاً با فعال کردن حالت "ورود بی‌نام" توسط یک کلمه عبور به نام "مهمان یا guest" یا کلمه عبور پوچ (یک رشته خالی) انجام می‌دهد. نشست زیر استفاده از یک سرویس دهنده FTP بی‌نام را نشان می‌دهد:

tpci_hpws4-1> ftp uofo.edu

Connected to uofo.edu.

220 uofo.edu FTP server (Version 1.7.109.2 Tue Jul 28 23:32:34 GMT 1992) ready.

Name (uofo:username): anonymous

331 Guest login ok, send userID as password.

Password: tparker

230 Guest login ok, access restrictions apply.

ftp> <Ctrl+d>

tpci_hpws4-2>

اگر سرویس دهنده FTP روی تواناسازی ورود بی‌نام تنظیم شده باشد، از شما یک کلمه عبور می‌خواهد و چون آنرا وارد نمیکنید بعد از یک اخطار در مورد محدودیتهای دسترسی به شما مجوز ورود داده می‌شود. اگر شما به فایلی روی سرویس دهنده نیاز داشته باشید می‌توانید آنرا انتقال بدهید. با عمومیت یافتن اینترنت، سایتهای FTP بی‌نام رشد فزاینده ای پیدا کرده اند و اهدافی نظیر نشر رایگان دانش یا تبلیغات دارند.

۷) سرویس دهنده های FTP

بسیاری از ماشینهای یونیکس به صورت پیش‌فرض به عنوان سرویس دهنده FTP عمل می‌کنند. برای ارائه امکانات سرویس دهنده FTP ، باید هنگام بوت شدن سیستم عامل ، دایمون[۱] ftpd اجرا شده باشد. این دایمون در سیستم عامل یونیکس معمولاً بوسیلهٔ پروسهٔ inted مدیریت می‌شود. هنگامیکه سیستم با استفاده از پروسه inetd بوت می‌شود ، به پورت

[۱] Daemon

فرمان TCP (کانال ۲۱) گوش می‌دهد تا درخواستهای رسیده برای برقراری ارتباط را ببیند، سپس دایمون ftpd را برای سرویس دادن به درخواست ، احیا می‌کند. اگر می‌خواهید مطمئن شوید که سیستم یونیکس یا لینوکس شما می‌تواند درخواستهای ftp را مدیریت کند باید مطمئن شوید که آیا برنامه ftpd ، وقتی که inetd آنرا فراخوانی می‌کند ، شروع می‌شود یا نه. در ضمن این کار را میتوان با بررسی فایل "تنظیمات پیکربندی"[1] inetd انجام داد. باید دید که آیا خطی مثل خط زیر در فایل تنظیمات inetd وجود دارد. اگر این خط وجود نداشته باشد باید آن را اضافه کنید:

ftp stream tcp nowait root /usr/etc/ftpd ftpd -l

در بسیاری از سیستم‌های یونیکس این خط در فایل تنظیمات پیکربندی inetd وجود دارد، گرچه ممکن است که به صورت توضیح[2] علامت خورده باشد که در این صورت باید علامتهای توضیح را بردارید. تنظیمات پیکربندی inetd در یونیکس یا لینوکس معمولا در فایل زیر ذخیره می‌شود:

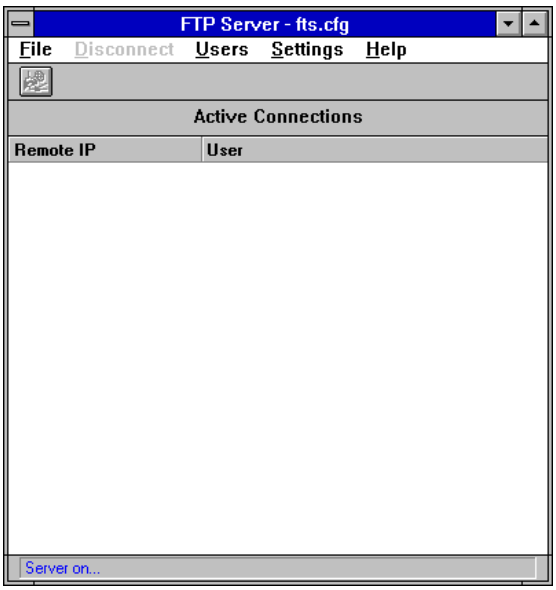
/etc/inetd.config یا /etc/inetd.conf

MS-Windows 95/98 فاقد یک برنامهٔ سرویس دهنده FTP به عنوان بخشی از نرم‌افزار توزیع شده خودشان هستند ، لذا در هنگام نیاز باید یک بستهٔ تجاری به آنها اضافه کرد. شکل (۸–۱٤) برنامهٔ Net Manage's ChameleonNFS را نشان می‌دهد که یک برنامهٔ سرویس دهنده FTP می‌باشد و می‌توان از آن، به عنوان ابزاری برای ارائه خدمات فایل در ماشینهای مبتنی بر سیستم عامل ویندوز 3.x استفاده کرد. برای فعال کردن نرم‌افزار Net Manage FTP Server باید ابتدا گزینه FTP Server Config در گروه برنامهٔ NetManage اجرا شود. در این صورت پنجره ای که در شکل (۸–۱۵) نشان داده شده، ظاهر می‌شود. پروسهٔ FTP Server اکنون فعال است و هرکس دیگری روی یک ماشین در شبکه ، اگر دارای مجوز دسترسی باشدمی‌تواند به ماشین شما متصل شود.

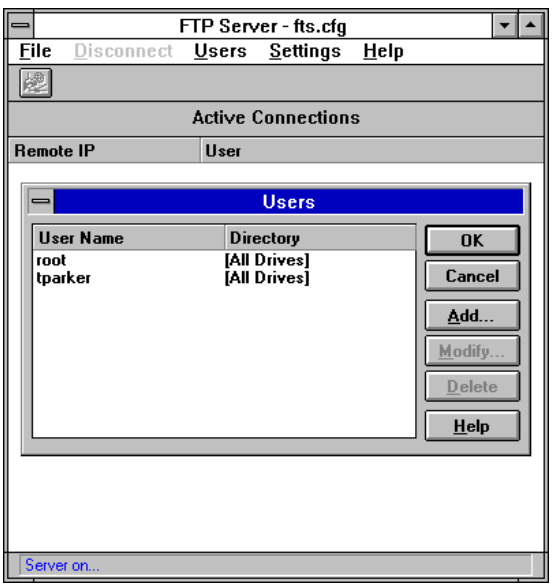
- دسترسی به سرویس FTP توسط تعریف لیست کاربران که در بستهٔ FTP Server قرار دارد کنترل می‌گردد و میتوان با انتخاب منوی User از Net Manage FTP Server کاربران را تعریف کرد. با این کار می‌توانید نام کاربرها را به سیستم خود اضافه کنید. اگر کاربری روی یک

[1] inetd Configuration File

[2] Comment



شکل (۱٤–۸) یک برنامهٔ سرویس دهنده FTP در محیط MS-Windows



شکل (۱۵–۸) یک برنامهٔ سرویس دهنده FTP در محیط MS-Windows

ماشین سعی کند که به نرم‌افزار FTP Server شما متصل شود، کد شناسائی و کلمه عبور آنها با نام و کلمهٔ عبوری که شما در این پنجره وارد کرده‌اید، تطبیق داده می‌شود. این به شما امکان می‌دهد که لیستی از کاربرهایی را که می‌توانند با سیستم شما مبادلهٔ فایل داشته باشند، تنظیم کنید. البته دسترسی به سیستم فایل تا زمانی امکان پذیر است که FTP Server در حال اجرا باشد.

اگر یک برنامهٔ FTP Server را اجرا می‌کنید بهترین کار آنست که برای هر کاربر یک شاخهٔ مجزا ایجاد کنید تا کاربر بتواند عملیات حذف ، اضافه یا تغییر فایلهایش را در شاخهٔ خودش انجام بدهد. از طرفی یک شاخهٔ عمومی[۱] ایجاد کنید بگونه ای که در اختیار همهٔ کاربران باشد وتمام فایلهایی که می‌خواهند وارد سیستم محلی شوند و یا از آن خارج گردند در آن قرار بگیرند. این کار موجب جلوگیری از دسترسی هر کاربر به فایلهای سایر کاربران و بالابردن ضریب امنیت سیستم خواهد شد. علاوه بر این امکان بررسی فایلهای وارده به شاخه عمومی از لحاظ آلوده بودن به ویروس وجود خواهد داشت.

اگر بخواهید یک سرویس بی‌نام یا مهمان برای کاربران روی شبکه تعریف کنید ، باید یک کاربر بدون کلمهٔ عبور یا یک کلمهٔ عبور ساده مثل مهمان (guest) تعریف شود. باید ناحیه‌ای که یک کاربر بی نام یا مهمان می‌تواند از آن استفاده کند به شدت محدود بوده و فقط اجازه دریافت فایل داشته باشد.

۸) پروتکل سادهٔ انتقال فایل : TFTP[۲]

پروتکل جزئی و ساده انتقال فایل که از این به بعد آنرا TFTP می‌نامیم یکی از ساده‌ترین پروتکلهای انتقال فایل است که امروزه از آن استفاده می‌شود. این پروتکل در دو زمینهٔ اصلی با FTP متفاوت است :

- این پروتکل نیاز به برقراری یک نشست و عملیات ورود[۳] به سیستم ندارد و بالطبع بدون برقراری یک نشست و انجام عملیات بررسی صلاحیت کاربر مشکلاتی نظیر دسترسی کاربران غیر مجاز محتمل خواهد بود.

TFTP از پروتکل UDP که یک پروتکل انتقال "بدون اتصال" است به جای TCP استفاده می‌کند و چون پروتکل UDP نظارتی بر ترتیب داده ها اعمال نمی‌کند بنابراین TFTP مجبور

---

[۱] Pubic
[۲] Trivial File Transfer Protocol
[۳] Login

است برای تضمین صحت و ترتیب داده ها الگوریتم‌هایی را به کـار بگـیرد.(TFTP شـمارهٔ پورت ٦٩ را به کار می‌برد.)

معمولاً برای انتقال فایل بین دو ماشین، در جایی که بشود از FTP استفاده کرد TFTP را بـه کار نمی‌برند چرا که در این پروتکل عملیاتی نظیر فهرست گیری از فایلها و شاخه هـا ، تغییـر شاخه جاری و احراز هویت کاربر امکان پذیر نیست ولی TFTP با تمام مشـکلاتش مزایـائی نسبت به FTP دارد. مثلا هنگام کار با ماشینهای بدون دیسک[۱] یا ایسـتگاههای کـاری[۲] TFTP کارآمدتر است. نوعاً TFTP برای بار کردن برنامه‌های کاربردی کوچک یا فونت روی ماشـینها به کار می‌رود. مهمترین کاربرد این پروتکل برای بوت کردن سیستمهائی است که بدون دیسک بوده و مجبورند از طریق ROM بوت شوند. در اینگونه موارد TFTP حتماً لازم است چـرا کـه ماشینهای بدون دیسک ، تا وقتی که سیستم عامل کاملا بار نشده باشـد، قـادر بـه اجـرای FTP نیستند. اندازه کوچک برنامه اجرایی TFTP و نیاز کم آن به حافظه باعث شده کـه بتـوان آن را در BOOTROM جا داد.

TFTP اجازه دسترسی به یک فایل را با در نظر گرفتن مشخصه[۳] آن فـایل اعمـال میکنـد. بـه عنوان مثال روی سیستم‌های یونیکس ، فـایلی کـه مشـخصه آن " خواندنـی/ نوشـتنی" اسـت می‌تواند توسط تمام کاربرها قابل دسترسی باشد. (اجازه خوانـدن و نوشـتن، هـر دو را داشـته باشد.) به خاطر این مقررات سهلگیرانه، بسیاری از مسـئولین شـبکه ، کنـترل بیشـتری را روی TFTP اعمال می‌کنند (یا کلاً استفاده از آن را منع می‌کنند.). در غـیر ایـن صـورت بـرای یـک کاربر آگاه بسیار ساده است که اقدام به دریافت فایلی کند که موجب خدشه دار شـدن امنیـت شود.

انتقال توسط TFTP به دلایل بسیاری می‌تواند با شکست مواجه شود و هر نوع خطـایی کـه هنگام عمل انتقال رخ بدهد، باعث شکست کامل انتقـال می‌شـود. TFTP برخـی از پیغام‌هـای اساسی خطا را پشتیبانی می‌کند ولی نمی‌تواند خطاهای ساده مثل کمبود منابعی نظیر حافظه یـا فضای ناکافی دیسک برای انتقال یک فـایل را رفـع و مدیریـت کنـد. در هنگـام بـروز چنیـن خطاهائی تمام مراحل انتقال باید از نو آغاز شود.

---

[۱] Diskless
[۲] Workstations
[۳] File Attribute

۸-۱) بسته‌های TFTP

بگونه ای که در بخش قبلی اشاره شد پروتکل TFTP از سوکتهای نوع دیتاگرام (مبتنی بر UDP) استفاده میکند و در ضمن کانالی مجزا برای ارسال فرمان وجود ندارد و چون فرامین و داده ها بطور همزمان ارسال می‌شوند بنابراین باید ساختاری برای بلوکهای داده تعریف شود تا طرفین ارتباط بتوانند داده ها را از فرامین و پیغامهای کنترلی تشخیص بدهند.

در پروتکل TFTP داده ها اعم از بلوکهای فایل ، فرامین یا پیغامهای کنترلی در قالب بلوکهائی از داده که ساختمان مشخصی دارند مبادله میشوند. هر بلوک داده که از ساختار تعریف شده تبعیت نکند بعنوان یک خطا تلقی شده و دور ریخته می‌شود.

پنج ساختار برای بلوکهای داده تعریف شده که در ادبیات این پروتکل به هر یک از آنها "بسته TFTP" گفته میشود. ساختار هر یک از این بسته ها به صورت زیر است:

بسته RRQ : تقاضای دریافت یک فایل

Opcode (2 Byte)	File Name (String)	0	Mode (String)	0

بسته WRQ : تقاضای ارسال یک فایل

Opcode (2 Byte)	File Name (String)	0	Mode (String)	0

بسته Data : ارسال داده های یک فایل

Opcode (2 Byte)	Block Number (2 Byte)	Data (0~512 Byte)

بسته Ack : پیغام تصدیق و پذیرش

Opcode (2 Byte)	Block Number (2 Byte)

بسته Error : پیغام خطا

Opcode (2 Byte)	Block Number (2 Byte)	Error Message (String)	0

همانگونه که از ساختار بسته ها مشخص است در تمام آنها دو بایت اول که Opcode نـام دارد نوع بسته را مشخص می‌کند. در این فیلد دو بایتی فقط یکی از مقادیر ۰ تا ۵ قرار می‌گیرد کـه معانی هر یک از آنها در جدول (۱۶-۸) مشخص شده است. حال باید ساختار برنامه سـرویس دهنده و مشتری و طریقه مبادله داده ها را بررسی نماییم:

توضیح	Opcode	نوع بسته
بسته Ack : پیغام تصدیق و پذیرش	٤	Ack
بسته Data : ارسال داده های یک فایل	۳	Data
بسته Error : پیغام خطا	۵	Error
بسته RRQ : تقاضای دریافت یک فایل	۱	RRQ
بسته WRQ : تقاضای ارسال یک فایل	۲	WRQ

جدول (۱۶-۸) انواع بسته های TFTP

الف) در برنامه سمت سرویس دهنده یک سوکت دیتاگرام باز شده و به آن شـماره پـورت ٦۹ نسبت داده می‌شود. (توسط تابع (bind) سپس برنامه به حالت دریافت (با تـابع (recvfrom) وارد شده و منتظر می‌ماند.

ب) در برنامه سمت مشتری سوکتی از نوع دیتاگرام باز میشود و سـپس یـک شـماره پـورت تصادفی به آن نسبت داده می‌شود. برنامه سمت مشتری در پروتکل TFTP از مواردی است کـه مجبور است به سوکت ایجاد شده آدرس پورت مقید کند چرا که در خـلال انتقـال یـک فـایل نباید شماره پورت عوض شود. پس از این کار تقاضای خود را در قـالب یکـی از بسـته هـای RRQ یا WRQ به سرویس دهنده ارسال مینماید.

ج) در صورتی که سرویس دهنده تقاضای رسیده را بپذیرد سـوکتی جدیـد بـاز کـرده و یـک شماره پورت تصادفی به آن نسبت می‌دهد. سپس از طریق سوکت جدیـد یـک بسـته Ack بـا مشخصات(Block No=0 و Opcode=4) به برنامه مشتری بر خواهد گرداند. سوکت جدیـد تـا پایان عملیات انتقال فایل باقی مانده و پس از آن بسته خواهد شد.

د) برنامه مشتری ، پس از دریافت بسته Ack ، اقدام به ارسال یا دریـافت بسـته هـای داده کـه دقیقا ۵۱۲ بایتی هستند می‌نماید. اگر بسته داده ای ارسال (یا دریافت) شود که اندازه کمـتر از ۵۱۲ بایت داشته باشد به عنوان آخرین بلوک فایل تلقی شده و خاتمه ارتباط را اعـلام خواهـد کرد.

در ساختار بسته های RRQ یا WRQ باید فیلدهای زیر مقدار دهی شوند:

- فیلد Opcode : مقدار ۱ برای بسته RRQ (تقاضای دریافت فایل)و مقدار ۲ برای بسته WRQ (تقاضای ارسال فایل)
- فیلد File Name : نام فایلی که باید دریافت یا ارسال شود.
- فیلد Mode : در این فیلد یکی از سه رشته زیر می‌تواند قرار بگیرد.
  - 'NetASCII' : یعنی فایل متنی است و از کدهای ASCII استفاده کرده است.
  - 'Byte' : یعنی فایل بصورت دودوئی و در قالب رشته ای از بایتهای تفسیر نشده ارسال یا دریافت می‌شود.
  - 'Mail' : مشخص می‌کند که مقصد یک کاربر است نه یک فایل . در چنین حالتی به جای نام فایل آدرس پست الکترونیکی یک شخص (به فرم user@xyz.com) یا مشخصه کاربری او قرار می‌گیرد و قالب اطلاعات قطعاً متنی خواهد بود.

فرآیند ارتباط در TFTP توسط برنامه مشتری با فرستادن یک درخواست RRQ یا WRQ آغاز می‌شود. به عنوان بخشی از درخواست ، نام فایل و حالت انتقال مشخص می‌شود و شماره اولین بلوک داده ۱ خواهد بود. وقتی یکی از طرفین یک بلوک ۵۱۲ بایتی داده ارسال کرد یک زمان سنج را تنظیم کرده و منتظر بسته Ack می‌ماند. اگر در مهلت مقرر (به طور معمول ۳ ثانیه) بسته Ack نرسید بلوک داده مجددا ارسال می‌شود. اگر عمل تکرار یک بلوک چندین بار متوالی انجام شود ولی پاسخ Ack بر نگردد (مثلا ۸ بار) خاتمه ارتباط تلقی و سوکت ایجاد شده بسته خواهد شد و تمام مراحل باید از نو انجام شود.

هر بلوک داده یک شماره ترتیب دارد که از ۱ شروع شده و به ازای انتقال موفقیت آمیز آن یکی اضافه خواهد شد. طرفین هیچگاه بسته ای را خارج از ترتیب قبول نخواهند کرد و با این موضوع امکان هر گونه خطایی در مورد ترتیب بسته ها منتفی خواهد بود. به دلیل اینکه فرستنده یک بسته داده ، قبل از فرستادن بسته بعدی منتظر پاسخ Ack می‌ماند به پروتکل TFTP پروتکل فلیپ فلاپ گفته می‌شود.

فرآیند انتقال وقتی پایان می‌یابد که یک بستهٔ داده با طول کمتر از ۵۱۲ بایت دریافت شود یا خطائی اتفاق بیفتد. فقط یک مورد خطا وجود دارد که قابل جبران است و آن خطا در شماره پورت است (خطای شماره ۵) بدین معنا که برنامه مشتری از شماره پورتی که در ابتدا روی آن توافق شده است استفاده نکرده و شماره پورت دیگری را به کار برده است. برای روشنتر شدن چگونگی توافق روی شمارهٔ پورت ارائه یک مثال ، مناسب خواهد بود:

برنامه A به عنوان مشتری شماره پورت تصادفی ۳۵٤۰ را انتخاب و یکی از بسته های RRQ یا WRQ را برای سرویس دهنده به شماره پورت ٦۹ ارسال میکند. سرویس دهنده سوکتی ایجاد و با انتخاب یک شماره پورت تصادفی مثل ۱٤۵۲۳ برای برنامهٔ مشتری بسته Ack را با این شماره پورت ارسال میکند. برنامه A تا پایان عملیات انتقال ، پورت مقصد را ۱٤۵۲۳ تلقی میکند. پس بدین نحو روی شماره های ۳۵٤۰ از مشتری و ۱٤۵۲۳ توافق شده است. هرگاه برنامه مشتری برای ارسال از شماره پورت دیگری استفاده کند خطای شماره ۵ اتفاق می‌افتد که قابل جبران است یعنی میتواند عملیات را با اصلاح شماره پورت ادامه بدهد. در جدول (۸–۱۷) انواع خطاهائی که توسط سرویس دهنده و با ارسال بسته Error گزارش خواهد شد تعریف شده اند.

شماره خطا	توضیح
۰	**نوع خطا تعریف نشده است. می‌توان به پیغام خطا در فیلد Error Message رجوع کرد.**
۱	**فایل درخواستی وجود ندارد**
۲	**اجازه دسترسی به فایل وجود ندارد**
۳	**ظرفیت دیسک پر شده یا آنکه در تخصیص فضا مشکل وجود دارد**
٤	**بسته ارسالی تعریف نشده و نا معتبر است**
۵	**شماره پورت استفاده شده توسط برنامه مشتری تعریف نشده است**
٦	**فایل از قبل وجود دارد و امکان نوشتن مجدد آن نیست.**
۷	**کاربر نام برده شده وجود خارجی ندارد**

جدول (۸–۱۷) شماره و نوع خطاهای گزارش شده توسط سرویس دهنده TFTP

۸–۲) دستورات TFTP

مهمترین دستورات کاربری TFTP در جدول (۸–۱۸) نشان داده شده است. مجموعهٔ فرامین TFTP شبیه FTP است ولی از چند جنبهٔ مهم به لحاظ طبیعت بدون اتصال این پروتکل متفاوت است. قابل توجه‌ترین تفاوت در فرمان connect (اتصال) است که به سادگی به جای اینکه یک نشست را برقرار کند، آدرس ماشین سرویس دهنده را معین می‌کند.

در سیستم عامل یونیکس نام برنامه سرویس دهندهٔ TFTP ، دایمون tftpd میباشد.

در زیر مثالی از برقراری ارتباط با یک سرویس دهنده TFTP ارائه شده است :

```
tpci_hpws1-1> tftp
tftp> connect tpci_hpws4
tftp> trace
Packet tracing on.
tftp> binary
Binary mode on.
tftp> verbose
Verbose mode on.
tftp> status
Connected to tpci_hpws4.
Mode: octet Verbose: on Tracing: on
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp> get /usr/rmaclean/docs/draft1
getting from tpci_hpws4:/usr/rmaclean/docs/draft1 to /tmp/draft1 [octet]
sent RRQ <file=/usr/rmaclean/docs/draft1, mode=octet>
received DATA <block1, 512 bytes>
send ACK <block=1>
received DATA <block2, 512 bytes>
send ACK <block=2>
received DATA <block3, 128 bytes>
send ACK <block=3>
Received 1152 bytes in 0.2 second 46080 bits/s]
tftp> quit
tpci_hpws1-2>
```

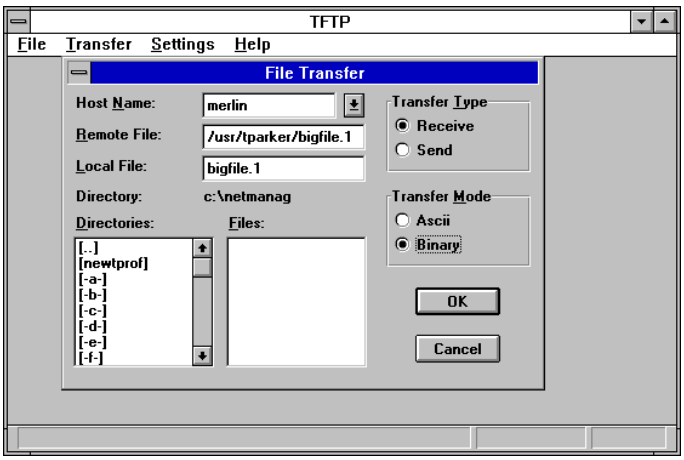
در فرآیند بالا می‌توانید ببینید که دستورات trace و verbose باعث میشود که در حین انتقال فایل تمامِ فعل و انفعالات بین سرویس دهنده و سرویس گیرنده و مبادله کدها بین دو ماشین روی خروجی نشان داده شود.

بگونه ای که در مثال مشخص شده است بعد از صدور دستور get برای تقاضای یک فایل هر بار که یک بلوک داده از فایل دریافت می‌شود در پاسخ به آن یک پیغام Ack ارسال میگردد تا سرویس دهنده ارسال بلوکها را ادامه بدهد.

سرویس دهنده TFTP روی تمام سیستم‌های یونیکس در دسترس میباشد ولی معمولا با احتیاط و اکراه نصب میشود چرا که این سرویس دهنده میتواند امنیت سیستم را به مخاطره بیندازد . شکل (۱۹-۸) برنامه کمکی TFTP از Chameleon NFS را نشان می‌دهد، که به شما اجازه می‌دهد نام یک یک ماشین راه دور ، نام فایل مورد نظر شما و نام فایلی که ارسال یا دریافت میشود و نوع انتقال را مشخص کرده تا انتقال فایل در "پس زمینه" و با استفاده از پروتکل UDP انجام شود.

فرامین TFTP	توضیح فرمان
Binary	تعیین حالت انتقال فایل به صورت دودوئی
Connect	تعیین آدرس ماشین سرویس دهنده
Get	تقاضای دریافت یک فایل از سرویس دهنده
Put	تقاضای ذخیره فایل روی ماشین سرویس دهنده
Trace	فرامین پروتکل را روی خروجی نشان میدهد
Verbose	تمامی اطلاعات لازم را به کاربر نشان میدهد

جدول (۱۸-۸) فرامین کاربری TFTP



شکل (۱۹-۸) نرم افزار TFTP در محیط MS_Windows

۹) مراجع این فصل

مجموعهٔ مراجع زیر می‌توانند برای دست آوردن جزییات دقیق و تحقیق جامع در مورد پروتکلهای معرفی شده در این فصل مفید واقع شوند.

مراجع مفید در مبحث TelNet

RFC1205	"Telnet 5250 Interface," Chmielewski, P.; 1991
RFC1198	"FYI on the X Window System," Scheifler, R.W.; 1991
RFC1184	"Telnet Linemode Option," Borman, D.A., ed.; 1990
RFC1091	"Telnet Terminal-Type Option," VanBokkelen, J.; 1989
RFC1080	"Telnet Remote Flow Control Option," Hedrick, C.L.; 1988
RFC1079	"Telnet Terminal Speed Option," Hedrick, C.L.; 1988
RFC1073	"Telnet Window Size Option," Waitzman, D.; 1988
RFC1053	"Telnet X.3 PAD Option," Levy, S.; Jacobson, T.; 1988
RFC1043	"Telnet Data Entry Terminal Option: DODIIS Implementation," Yasuda, A.; Thompson, T.; 1988
RFC1041	"Telnet 3270 Regime Option," Rekhter, Y.; 1988
RFC1013	"X Window System Protocol, version 11: Alpha Update," Scheifler, R.W.; 1987
RFC946	"Telnet Terminal Location Number Option," Nedved, R.; 1985
RFC933	"Output Marking Telnet Option," Silverman, S.; 1985
RFC885	"Telnet End of Record Option," Postel, J.B.; 1983
RFC861	"Telnet Extended Options: List Option," Postel, J.B; Reynolds, J.K.; 1983

RFC 860	"Telnet Timing Mark Option," Postel, J.B.; Reynolds, J.K.; 1983
RFC 859	"Telnet Status Option," Postel, J.B.; Reynolds, J.K.; 1983
RFC 858	"Telnet Suppress Go Ahead Option," Postel, J.B,; Reynolds, J.K.; 1983
RFC 857	"Telnet Echo Option," Postel, J.B.; Reynolds, J.K.; 1983
RFC 856	"Telnet Binary Transmission," Postel, J.B.; Reynolds, J.K.; 1983
RFC 855	"Telnet Option Specifications," Postel, J.B.; Reynolds, J.K.; 1983
RFC 854	"Telnet Protocol Specification," Postel, J.B.; Reynolds, J.K.; 1983
RFC 779	"Telnet Send-Location Option," Killian, E.; 1981
RFC 749	"Telnet SUPDUP-Output Option," Greenberg, B.; 1978
RFC 736	"Telnet SUPDUP Option," Crispin, M.R; 1977
RFC 732	"Telnet Data Entry Terminal Option," Day, J.D.; 1977
RFC 727	"Telnet Logout Option," Crispin, M.R.; 1977
RFC 726	"Remote Controlled Transmission and Echoing Telnet Option," Postel, J.B.; Crocker, D.; 1977
RFC 698	"Telnet Extended ASCII Option," Mock, T.; 1975

مراجع مفید در مبحث پروتکلهای انتقال فایل

RFC 1094	"NFS: Network File System Protocol Specification," Sun Microsystems, Inc.; 1989
RFC 1068	"Background File Transfer Program (BFTP)," DeSchon, A.L.; Braden, R. T.; 1988
RFC 959	"File Transfer Protocol," Postel, J.B.; Reynolds, J.K.; 1985
RFC 949	"FTP Unique-Named Store Command," Padlipsky, M.A.; 1985
RFC 783	"TFTP Protocol (Revision 2)," Sollins, K.R.; 1981
RFC 775	"Directory Oriented FTP Commands," Mankins, D.; Franklin, D.; Owen, A.D.; 1980