Remote file transfer

Adapted from materials by Dr. Carrier

Alex Perz - Unsplash

scp

scp

scp

scp

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

- The host we are copying to/from

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

- The host we are copying to/from
- The username to use on that host

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

- The host we are copying to/from
- The username to use on that host

Thus we get:

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

- The host we are copying to/from
- The username to use on that host

Thus we get:

```
scp [[user@]src_host:]src_path [[user@]dest_host:]dest_path
```

scp

We've discussed `ssh` - ***S***ecure ***SH***ell

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

- The host we are copying to/from
- The username to use on that host

Thus we get:

```
scp [[user@]src_host:]src_path [[user@]dest_host:]dest_path
```

Note that items in [ ] can sometimes be omitted

scp

We've discussed `ssh` - ***S**ecure **SH**ell*

Now, we're talking about `scp` - Secure CoPy

Reminder, we locally copy files like this:

```
cp <source> <destination>
```

`scp` is the same, but works remotely!

However, we need more info!

- The host we are copying to/from
- The username to use on that host

Thus we get:

```
scp [[user@]src_host:]src_path [[user@]dest_host:]dest_path
```

Note that items in [ ] can sometimes be omitted

Example, transfer local file to EOS

```
scp local_file username@eos01.cis.gvsu.edu:~/my_dir
```

Directedness in scp

scp doesn’t care which systems we are moving files to/from

The last example copied a local file to the server

```
scp local_file username@eos01.cis.gvsu.edu:~/my_dir
```

What if we want to copy a file from server to laptop?

```
scp username@eos01.cis.gvsu.edu:~/my_dir/file .
```

Why stop there? We could also copy files directly from one server to another, all from out local machine!

rsync

rsync

`rsync` is like a fancier `scp`

rsync

`rsync` is like a fancier `scp`

It checks to make sure there are differences between the files

rsync

`rsync` is like a fancier `scp`

It checks to make sure there are differences between the files

If there are differences, it only copies them, not the whole file

rsync

`rsync` is like a fancier `scp`

It checks to make sure there are differences between the files

If there are differences, it only copies them, not the whole file

Command format is very similar to `scp`:

rsync

`rsync` is like a fancier `scp`

It checks to make sure there are differences between the files

If there are differences, it only copies them, not the whole file

Command format is very similar to `scp`:

```
rsync [[user@]src_host:]src_path
[[user@]dest_host:]dest_path
```

rsync

`rsync` is like a fancier `scp`

It checks to make sure there are differences between the files

If there are differences, it only copies them, not the whole file

Command format is very similar to `scp`:

```
rsync [[user@]src_host:]src_path
[[user@]dest_host:]dest_path
```

Note that rsync has a ton of options!