

Slide 1

Administrivia

- If you have not turned in all of Homeworks 1 through 7, please do so as soon as you can. I will give some points for anything turned in through 5pm Monday, but not later.
- If you have not submitted a project proposal, please do that as soon as possible too, but no later than 5pm Friday. Talk to me if you're having trouble choosing a topic.

Slide 2

A Few (More) Words About "Cron Jobs"

- "Cron jobs" are the traditional-UNIX way of scheduling periodic tasks. A background process (`crond` on our system) checks every minute, and can run things at specified intervals. Typically these are shell scripts.
- What to run is determined by contents of some system directories (`/etc/cron.*` on our systems), and per-user "crontabs".
`crontab` command can be used to list and edit.
`man 5 crontab` gives details of syntax.
- Output is mailed to user. So there must be a mail transport agent running (e.g., `sendmail`). Also, if not forwarded, mail goes to local mail spool. Use a `.forward` file to forward, or a mailreader than can check the local mail spool.

Miscellaneous Useful Tips

Slide 3

- If you know a lot of editor tricks, but only a few shell tricks, consider using editor to build temporary scripts. (Example(s).)
- Remember that you can customize many things about your environment:
 - Define aliases, functions, and environment variables in `.bashrc` (for `bash`).
 - Write shell scripts and put in any (writeable-by-you) directory in your search path (`echo $PATH`).

A Very Little about CGI

Slide 4

- What's CGI? "Common Gateway Interface" — protocol for communicating between Web forms and (your) programs.
- Programs can be in any language that can read from `stdin`, write to `stdout`, and read environment variables.
- May require some setup/configuration in Web server. More information in tutorials linked from "Useful links" page ([here](#)).
- Notice that there are security implications — you're letting anyone with a Web browser execute these programs on your computer. No problem if they're harmless, but "be careful"?

Slide 5

CGI “Scripts”

- Input is one long string of name-value pairs, either from an environment variable or from stdin. (More about this in next slide.)
- Output begins with a single line giving content type, e.g.:
`Content-type: text/plain` or
`Content-type: text/html`
followed by a blank line.
After that, it can be anything appropriate for the content type.
- See `print*` examples from class, linked from “Sample programs” page, ([here](#)).
Also notice that Web “dynamic content” can be provided in other ways than CGI — e.g., PHP example supplied by a former student (sample program `notcgi.php`).

Slide 6

Input to CGI “Scripts”

- Missing link: CGI programs get input via environment variable or standard input, but where does it come from?
- Usual answer — HTML “form” (FORM tag). See examples linked from “Sample programs” page.
- Input, as examples show, is in the form of variable=value pairs. Parsing this could be a pain, so might best to do this in a language that provides nice library functions for it (e.g., Perl), and/or search for appropriate already-written functions in your programming language of choice.

Other Ways to Generate “Dynamic Content” for the Web

- CGI scripts (in your programming language of choice) are one way.
- Other ways include Javascript, PHP, Java applets, etc., etc. Probably more suitable if you want more-two-way interaction.

Slide 7

Minute Essay

- Have you written “dynamic content” Web pages? using what?

Slide 8