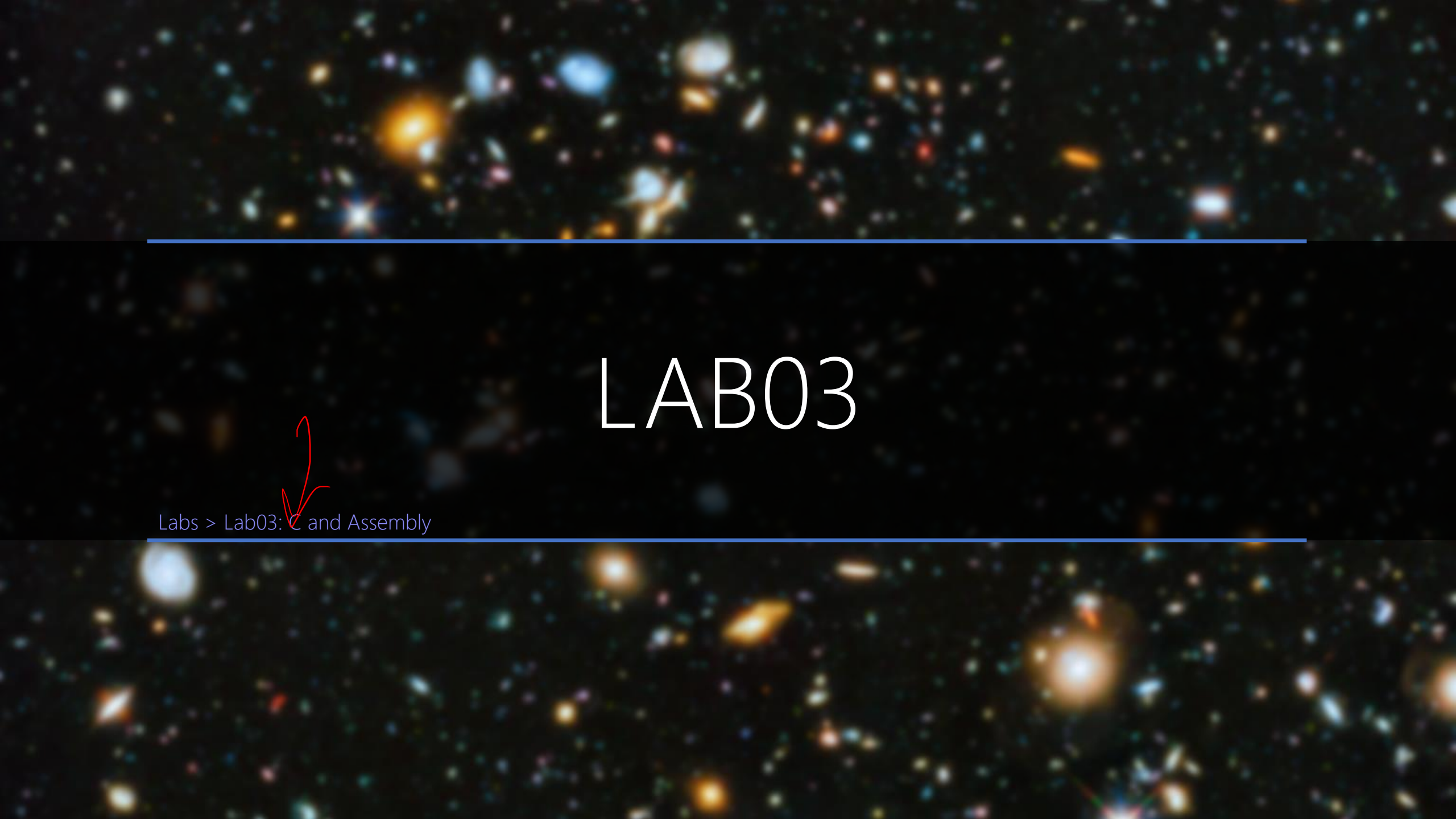
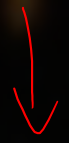

MATT DAMON
THE
BOURNE ULTIMATUM



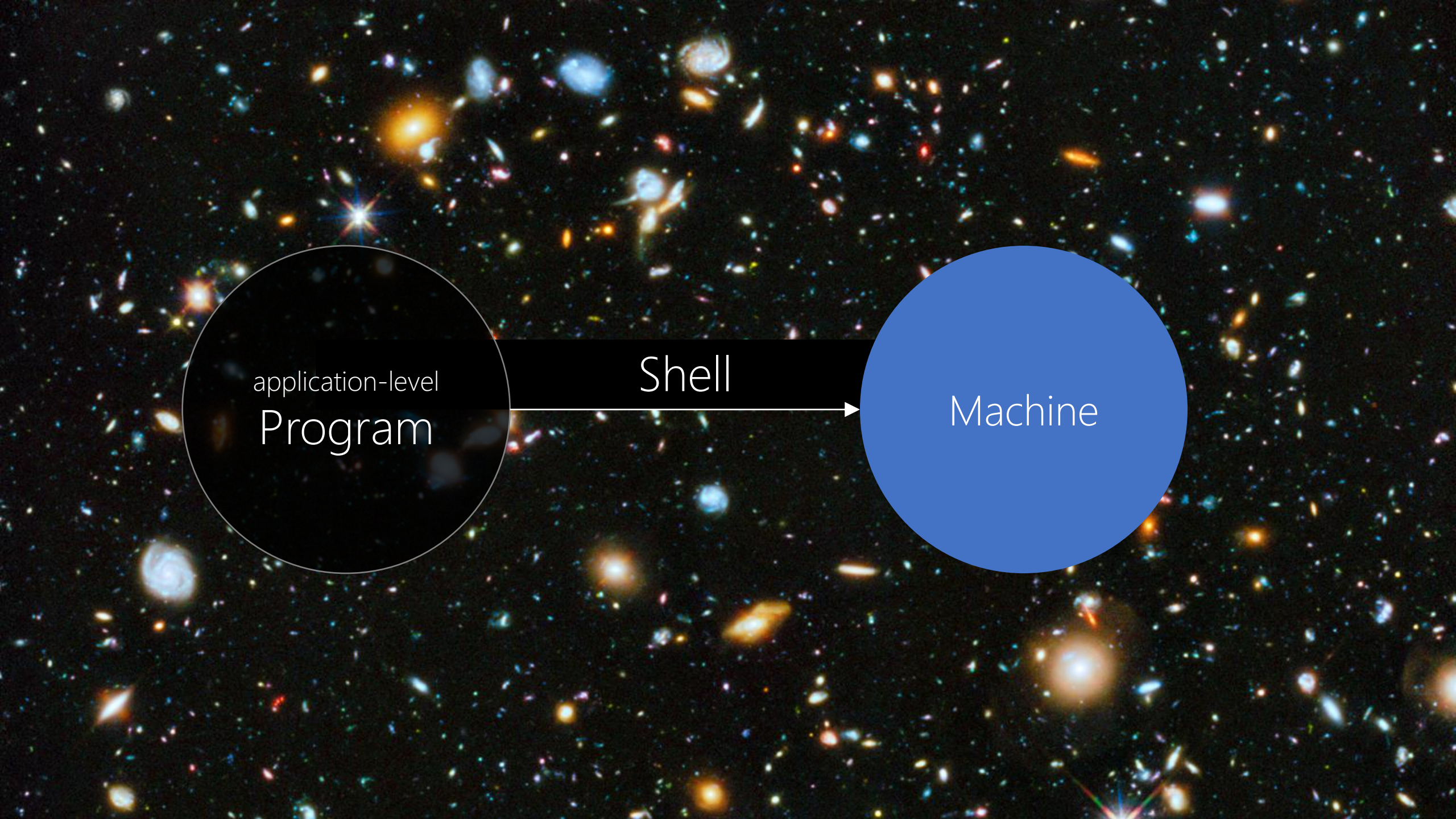
LAB03

Labs > Lab03: C and Assembly

LEC03



Lecs > Lec03: Shell


application-level
Program
Shell
Machine

Know the Computer System Status

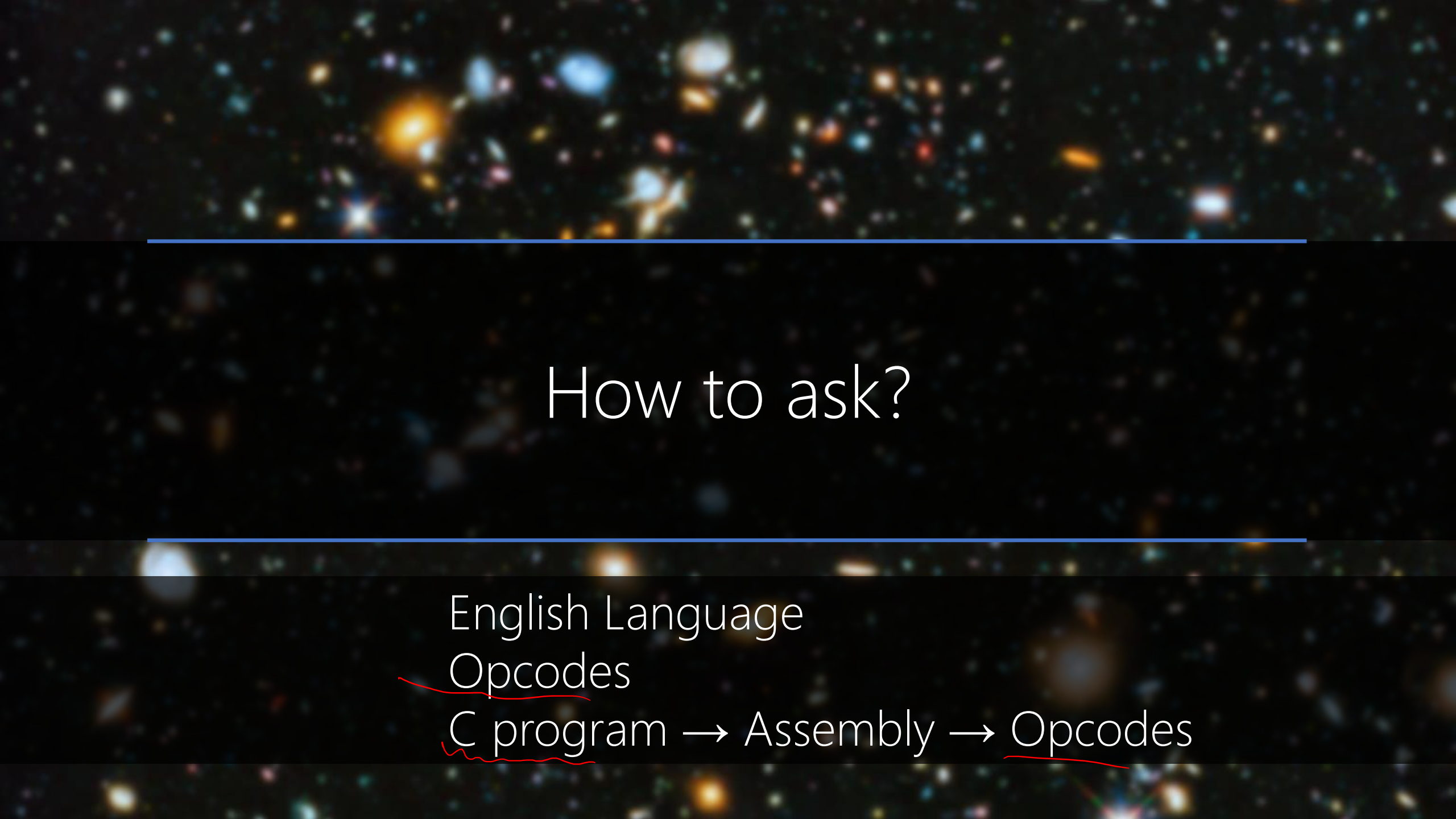
Is there any other programs?
Is there any other files?
What are the files?
What are the files' sizes?
Are there any hidden files?
Does memory/HDD have free space?
And many other questions.

Who knows the answers?

Oracle
Programmer
Shell
Kernel
Processor

Who knows the answers?

Oracle
Programmer
Shell
Kernel
Processor

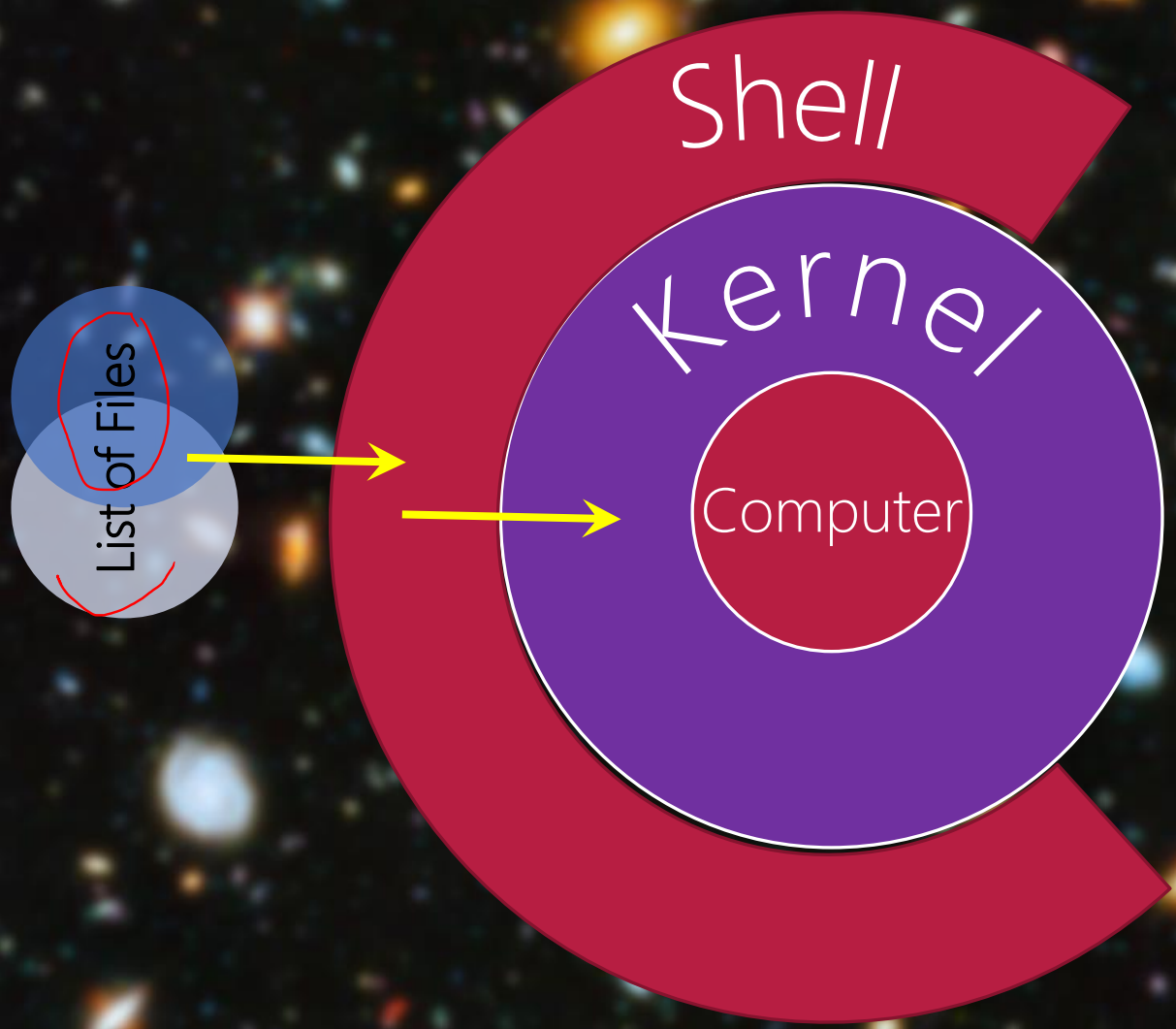


How to ask?

English Language
Opcodes
C program → Assembly → Opcodes

How to ask?

English Language
Opcodes
C program → Assembly → Opcodes



Computer

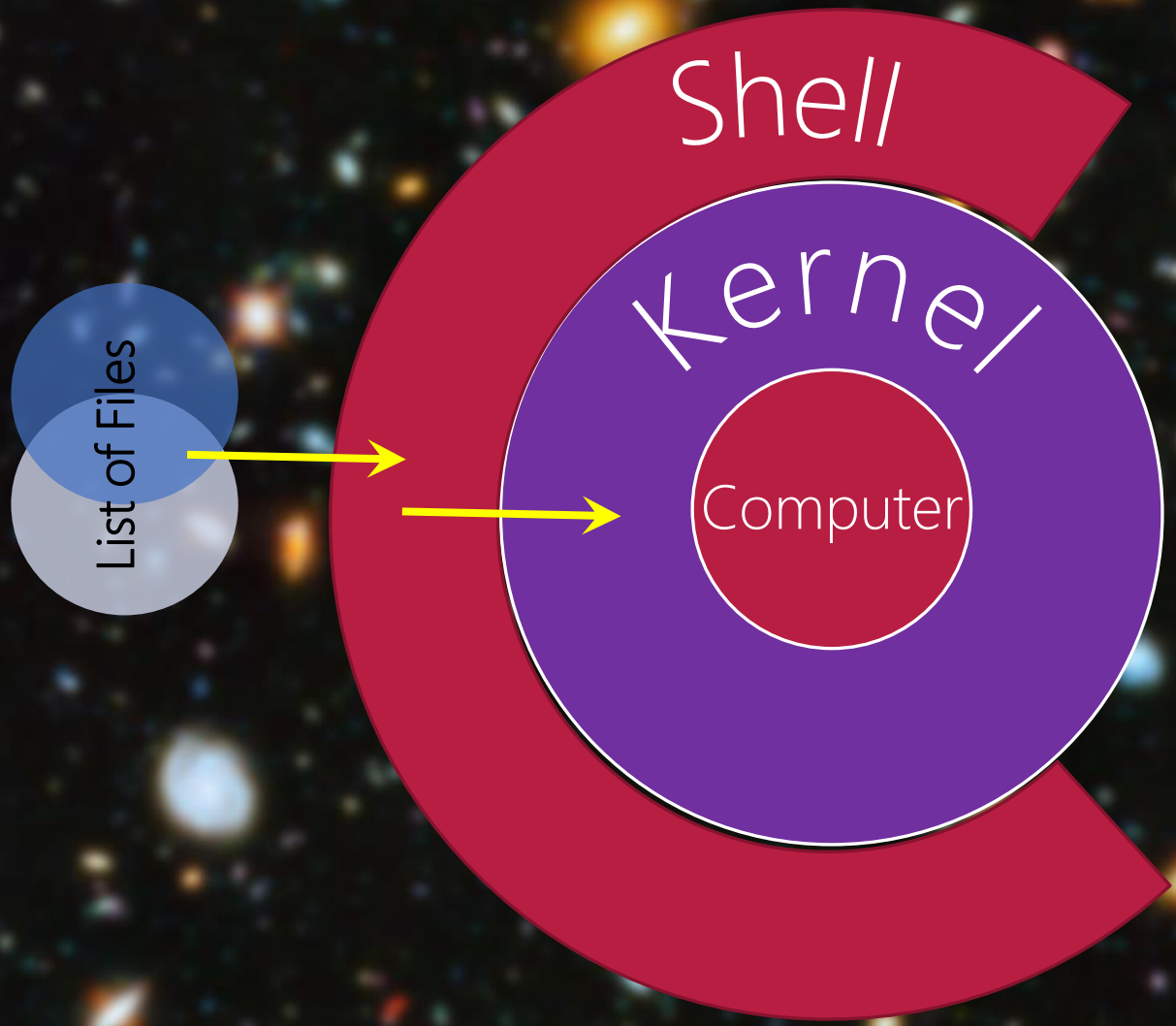
Memory

Kernel
File System

Shell

Bus

Processor



Computer

Memory

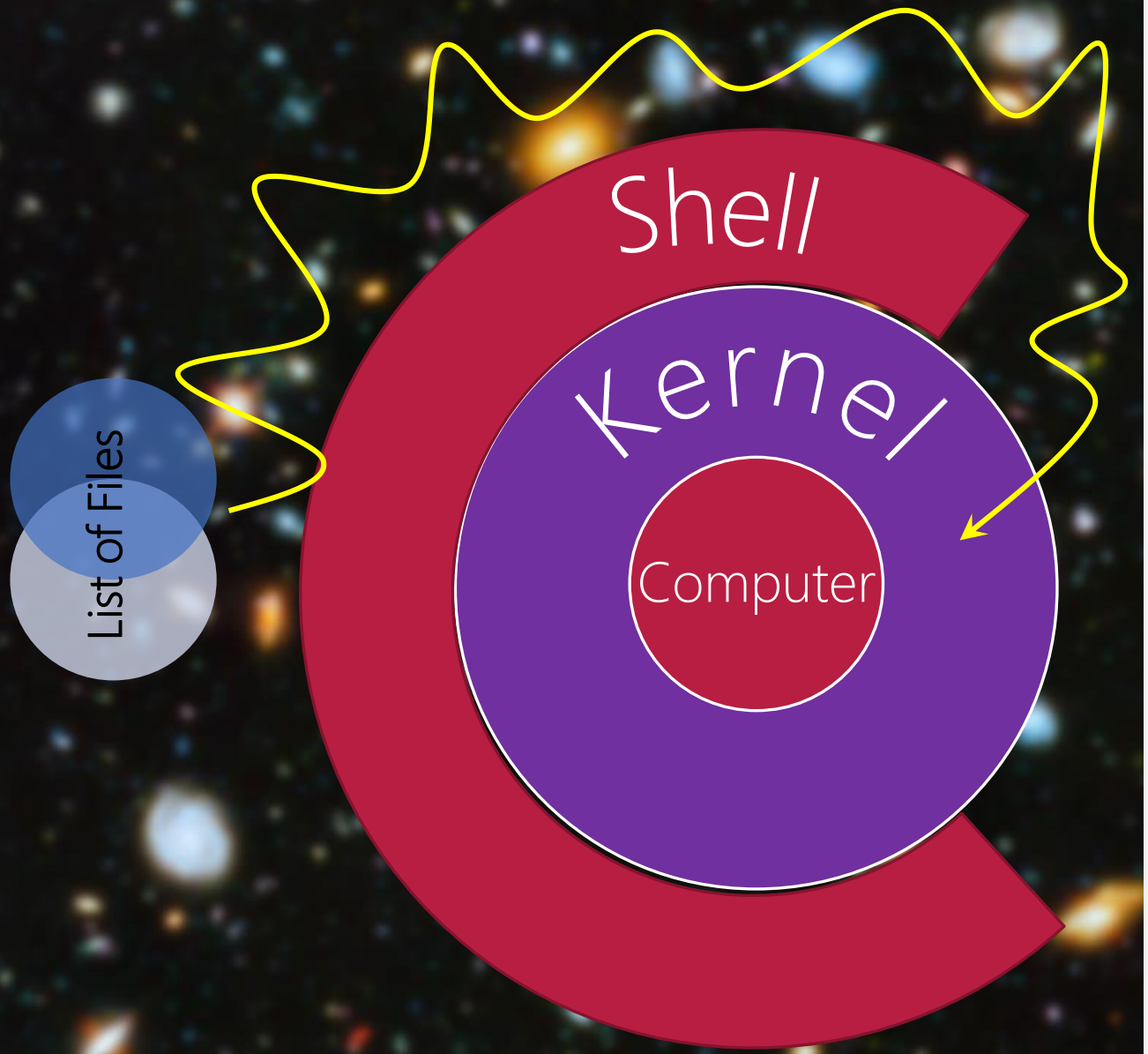
Kernel
File System

Shell

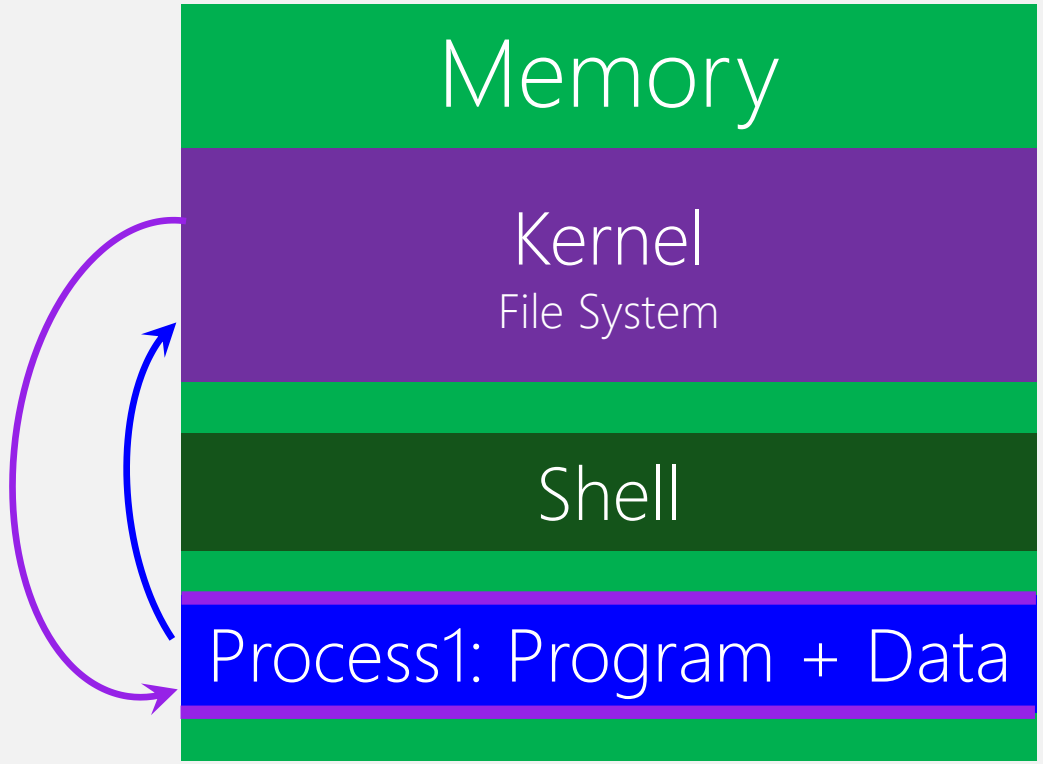
Process1: Program + Data

Bus

Processor

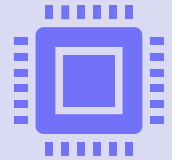


Computer

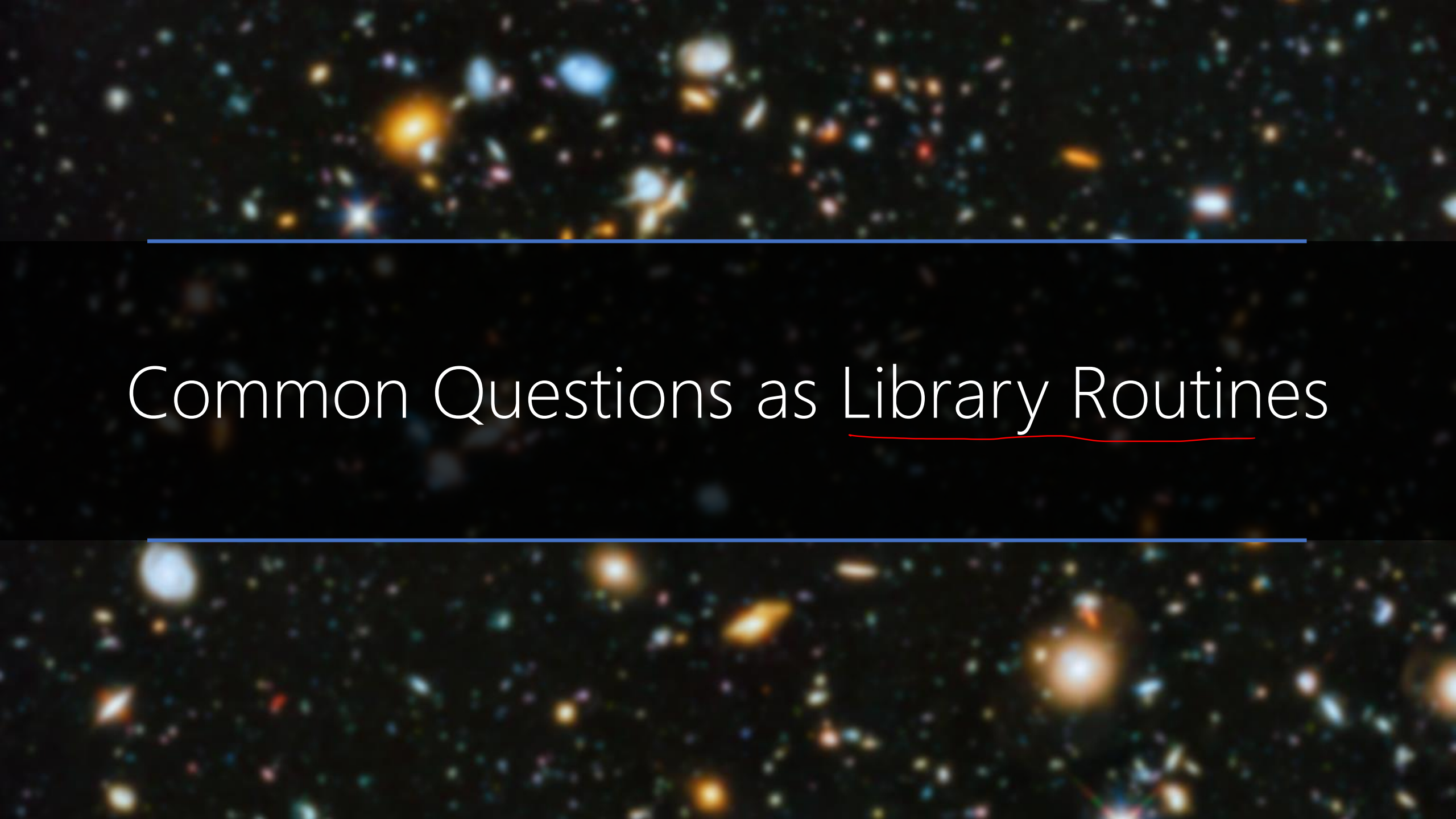


Bus

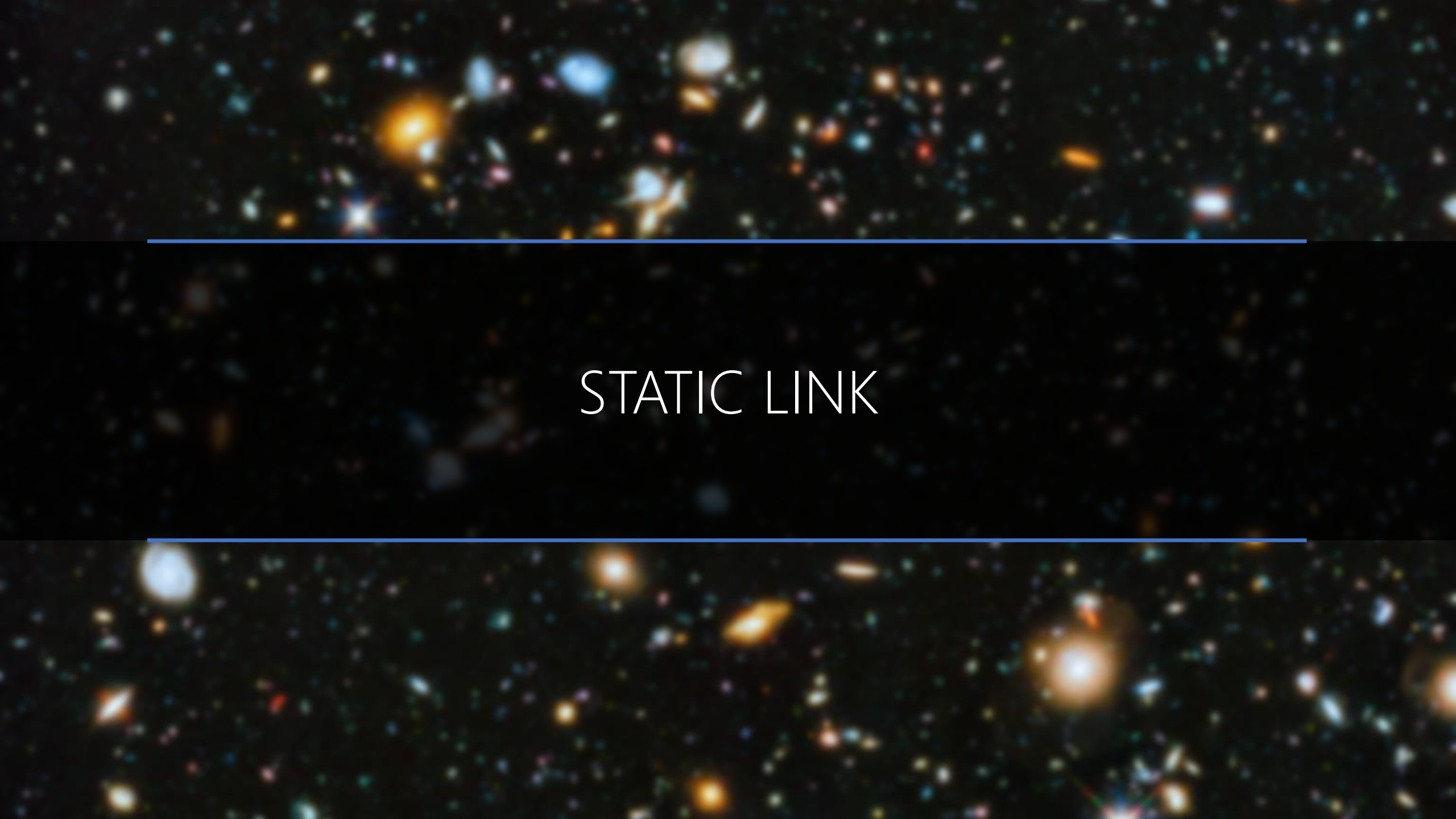
Processor



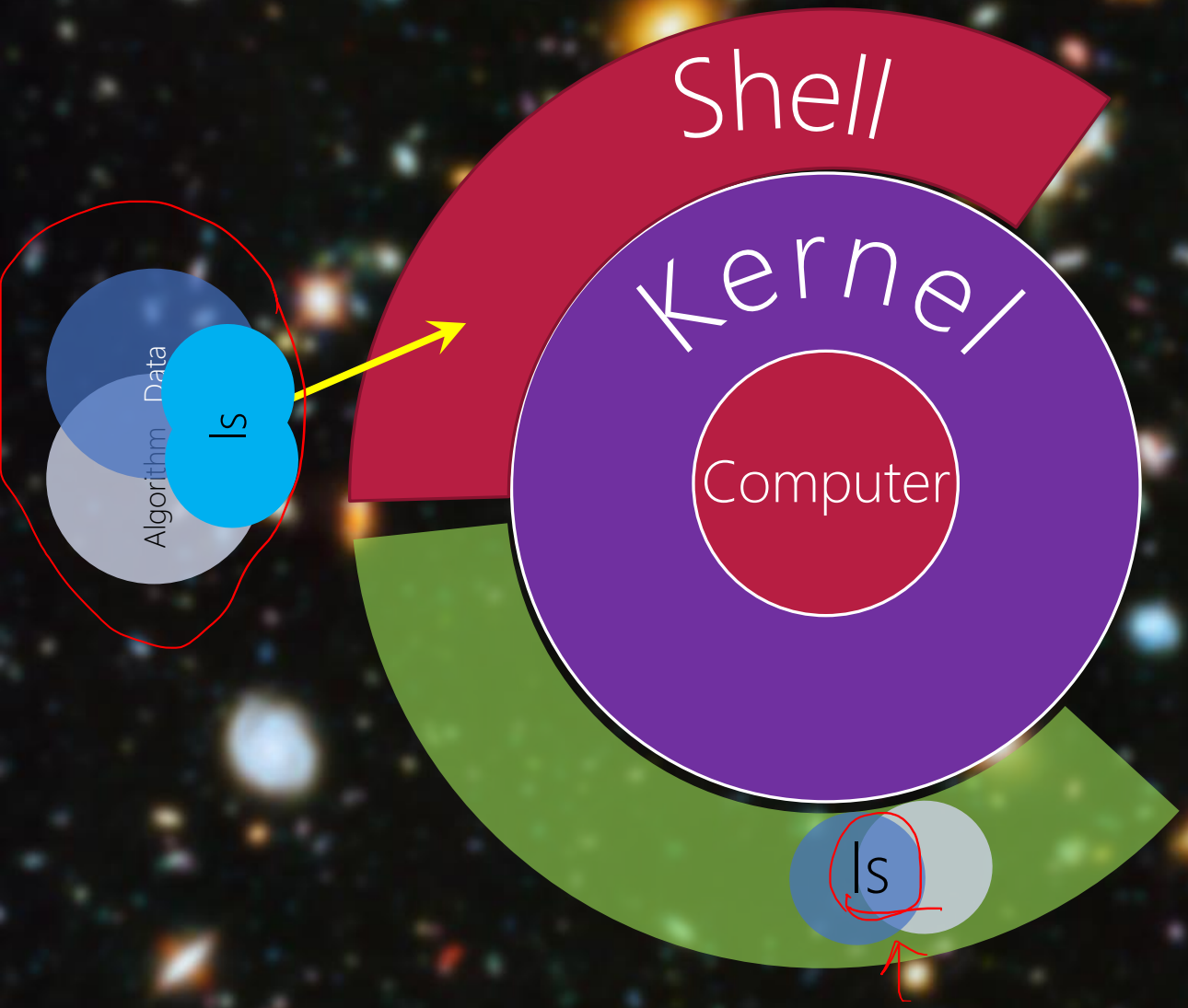
https://github.com/wertarbyte/coreutils/blob/master/src/ls.c



Common Questions as Library Routines



STATIC LINK



Computer

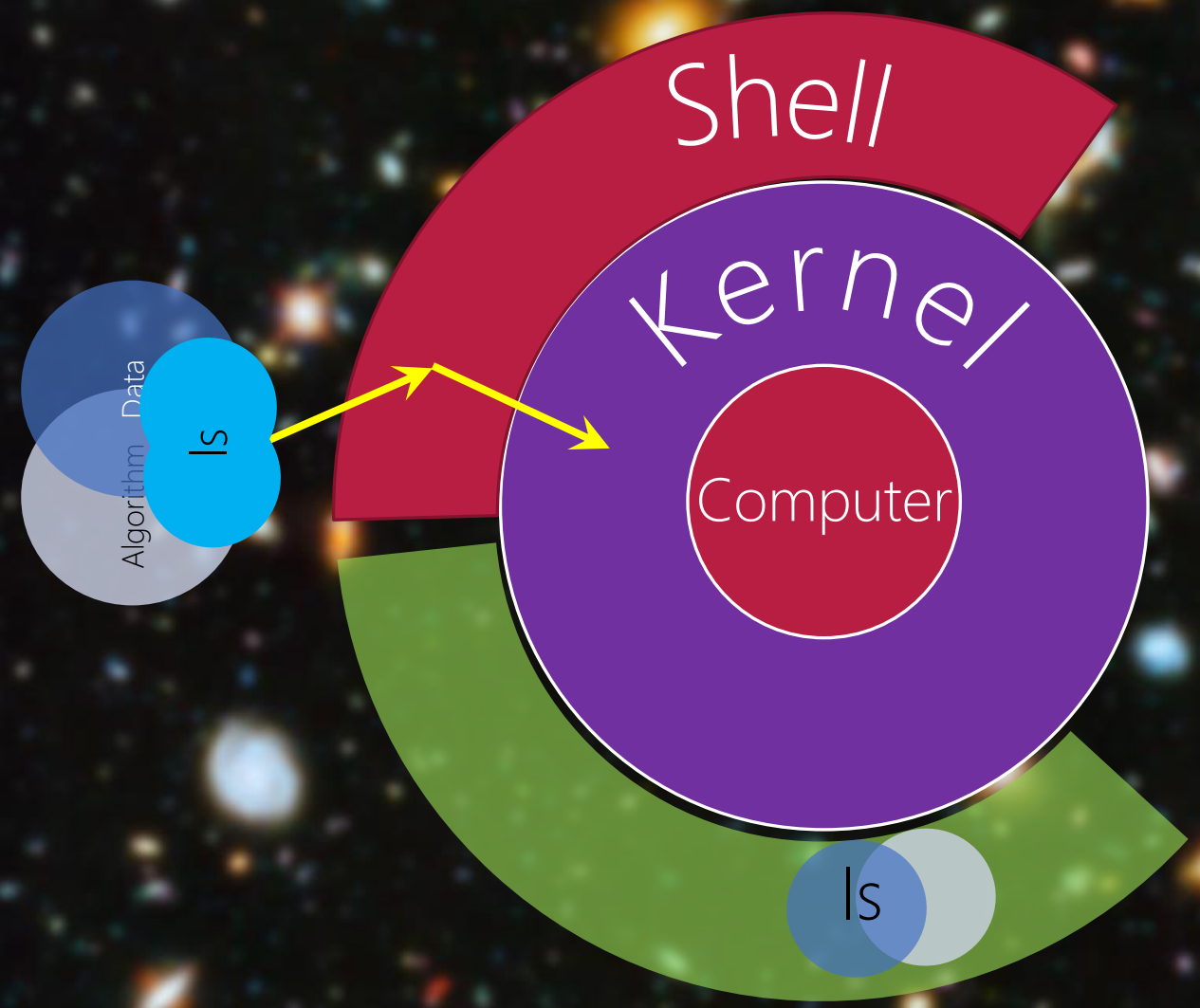
Memory

Kernel

Shell

Bus

Processor



Computer

Memory

Kernel

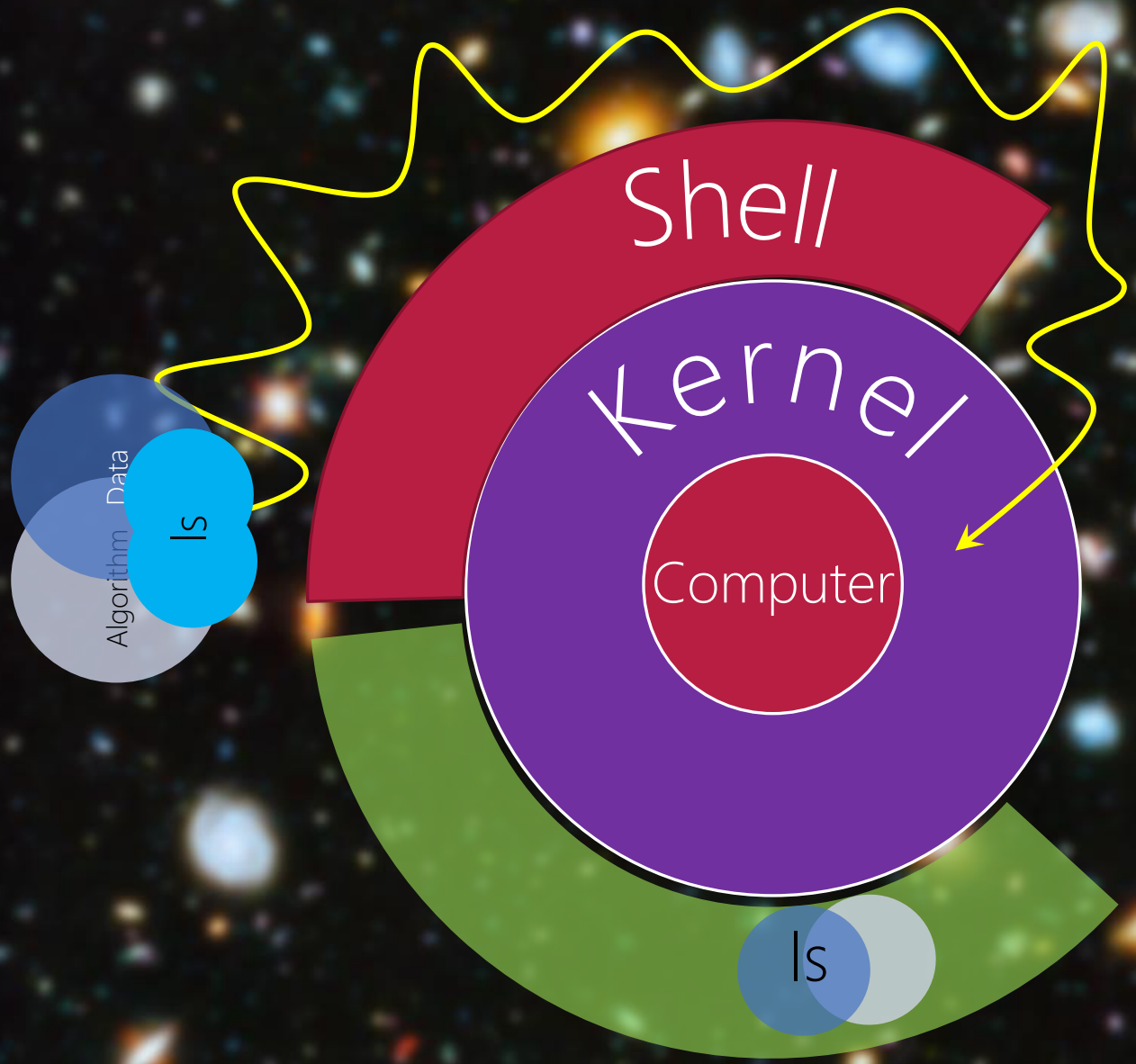
Shell

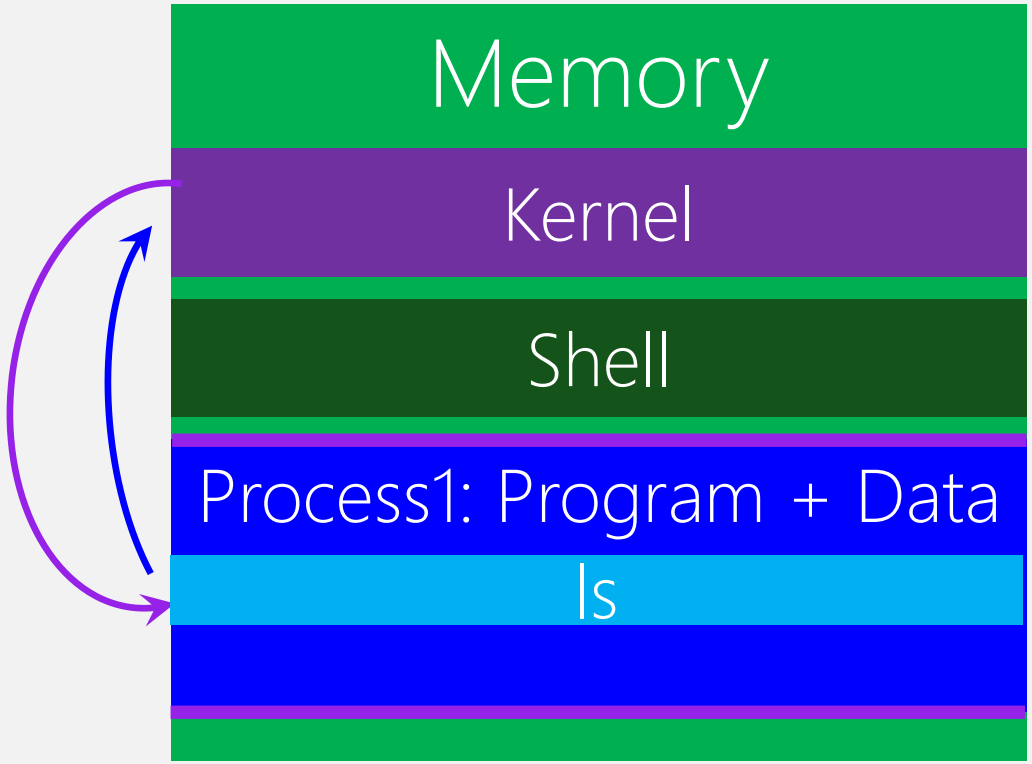
Process1: Program + Data

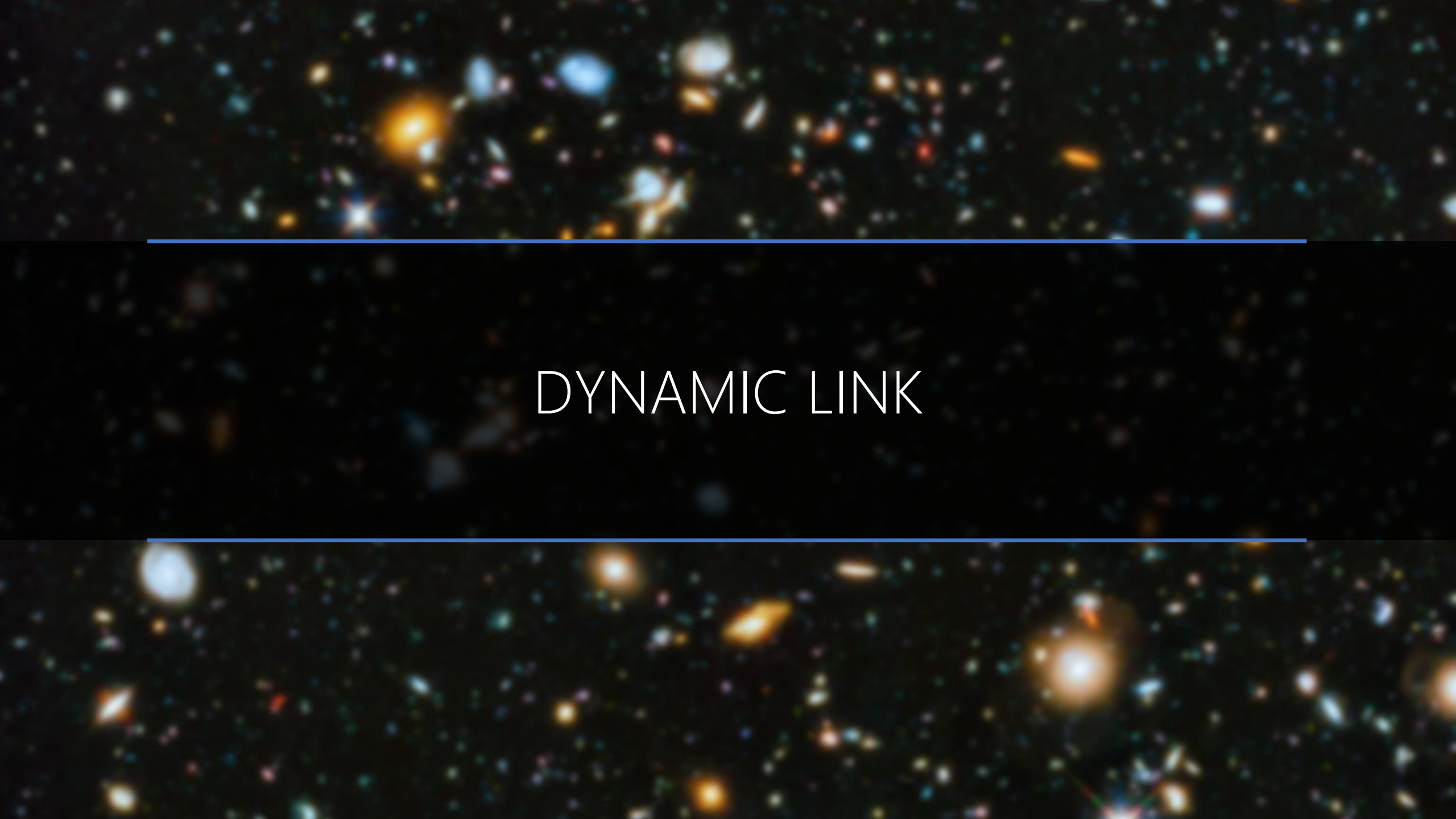
Is

Bus

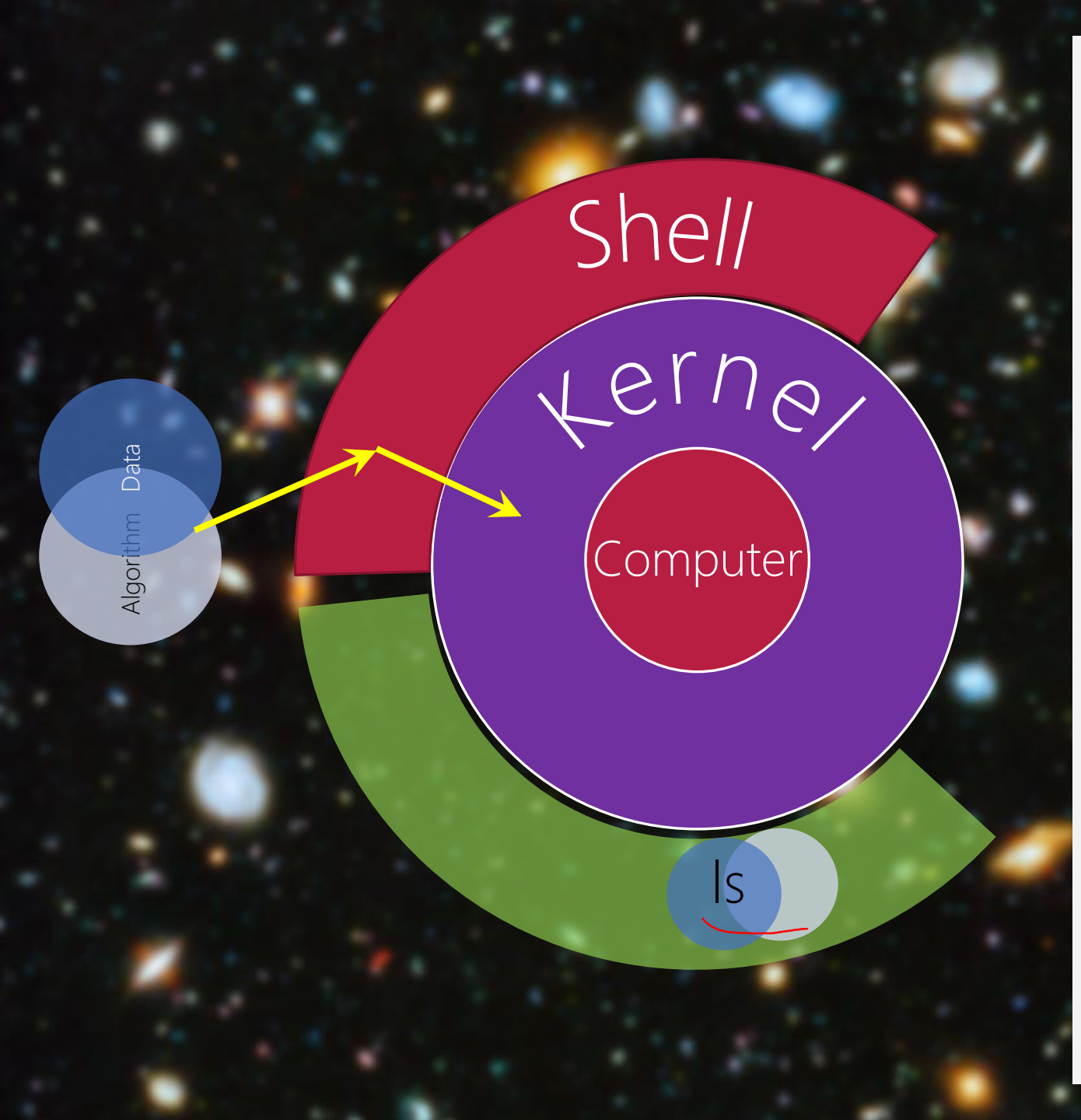
Processor


Shell
Kernel
Computer
Algorithm
Data
Is
Is


Computer
Memory
Kernel
Shell
Process1: Program + Data
Is
Bus
Processor



DYNAMIC LINK



Computer

Memory

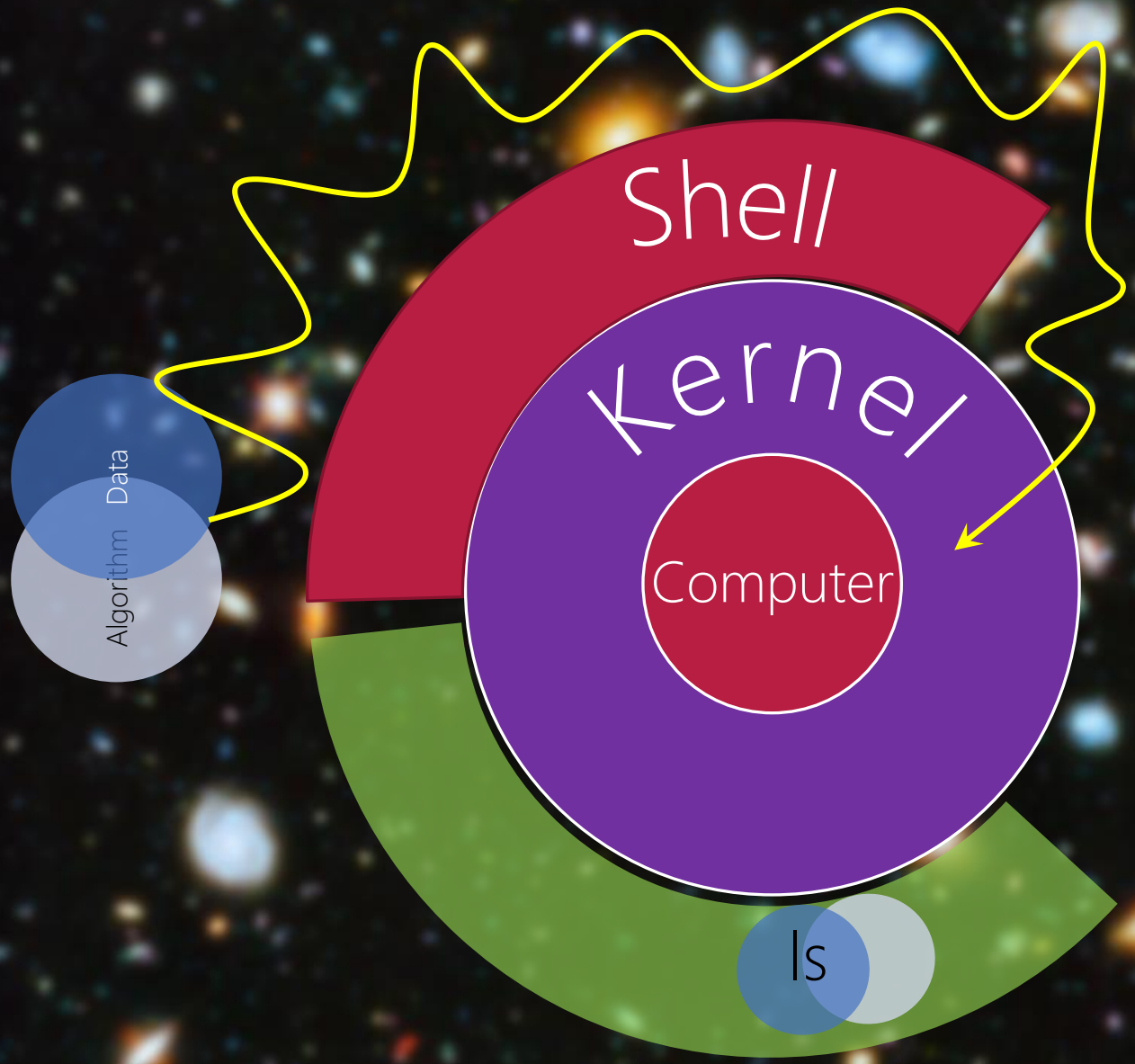
Kernel

Shell

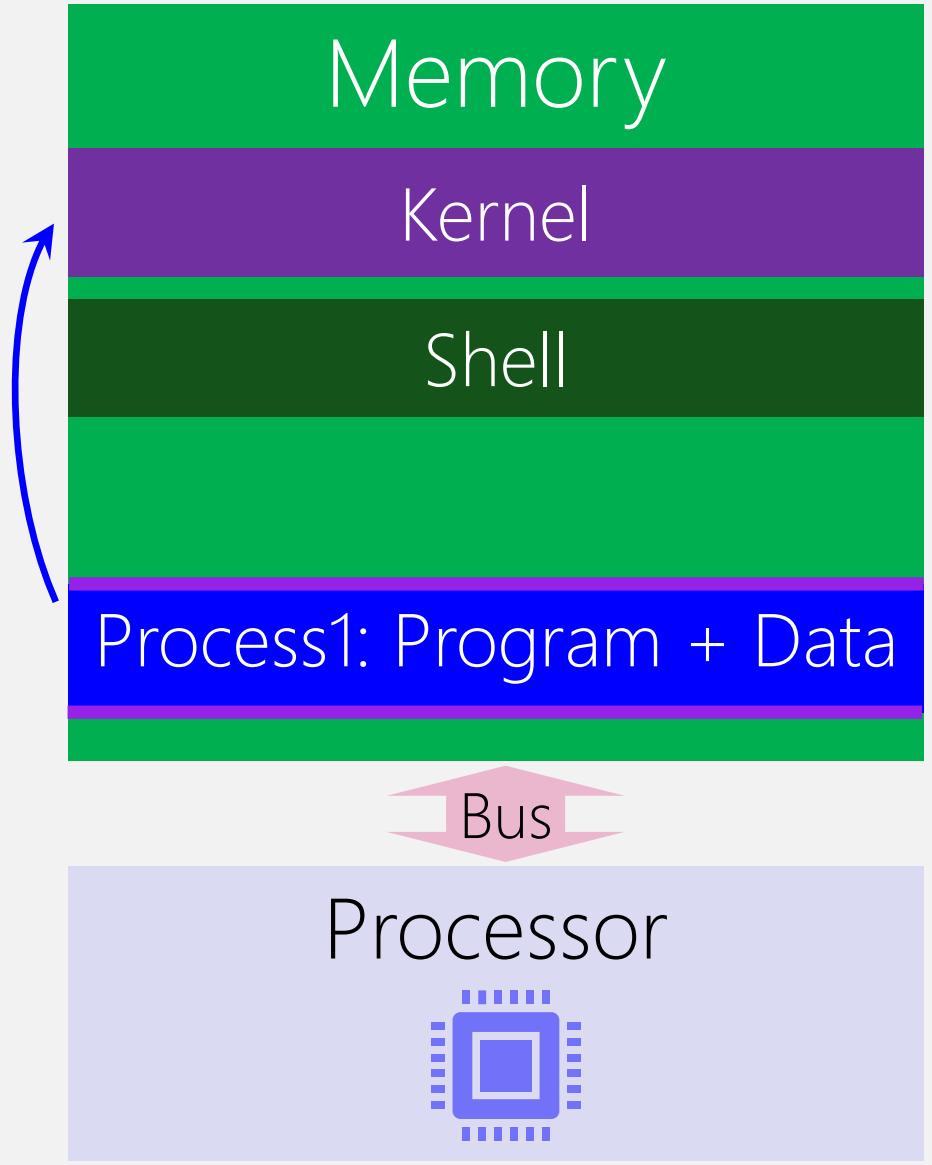
Process1: Program + Data

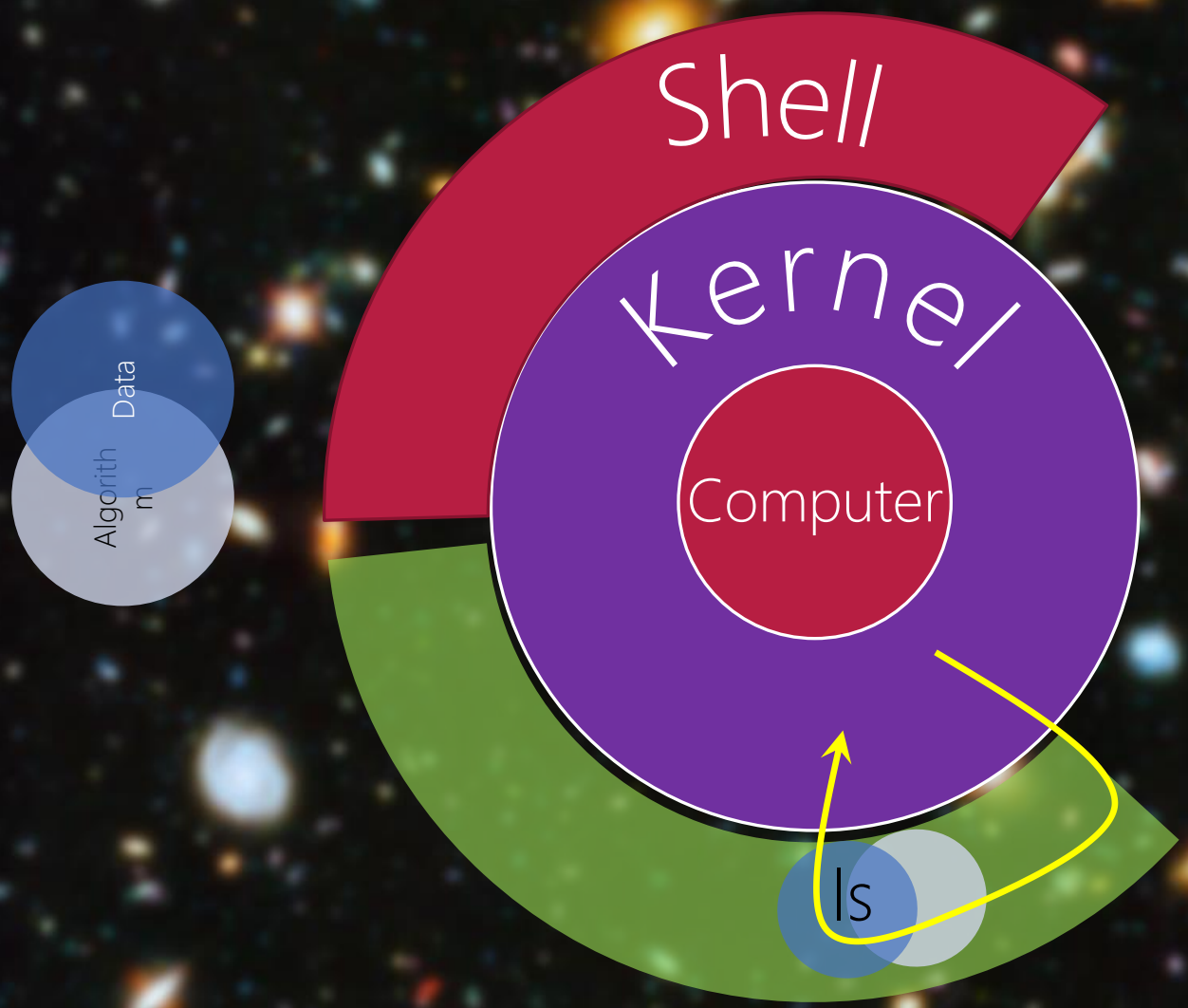
Bus

Processor

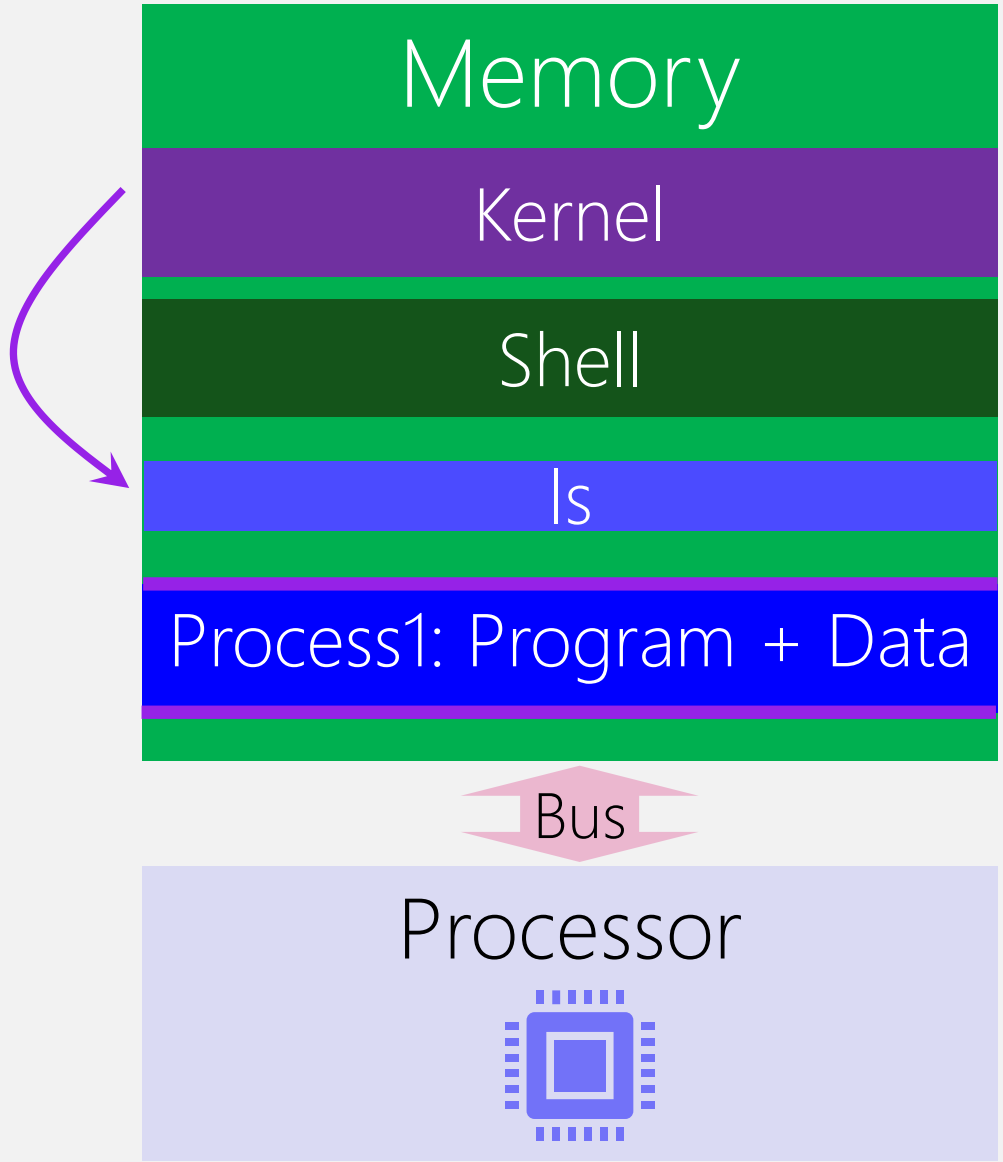

Shell
Kernel
Computer
Algorithm
Data
ls

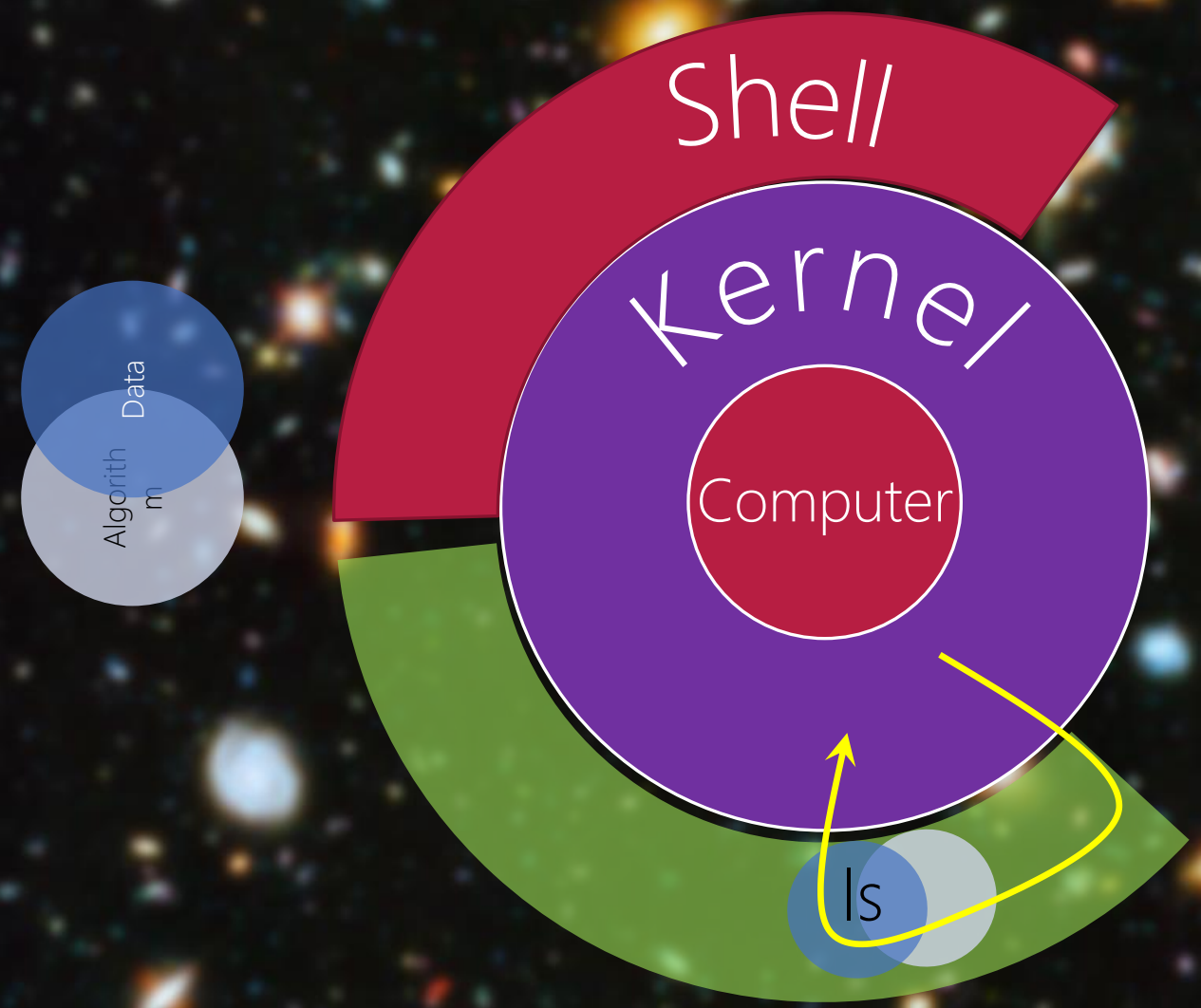
Computer


Memory
Kernel
Shell
Process1: Program + Data
Bus
Processor


Shell
Kernel
Computer
Is
Data
Algorithm

Computer


Memory
Kernel
Shell
Is
Process1: Program + Data
Bus
Processor



Computer

Memory

Kernel

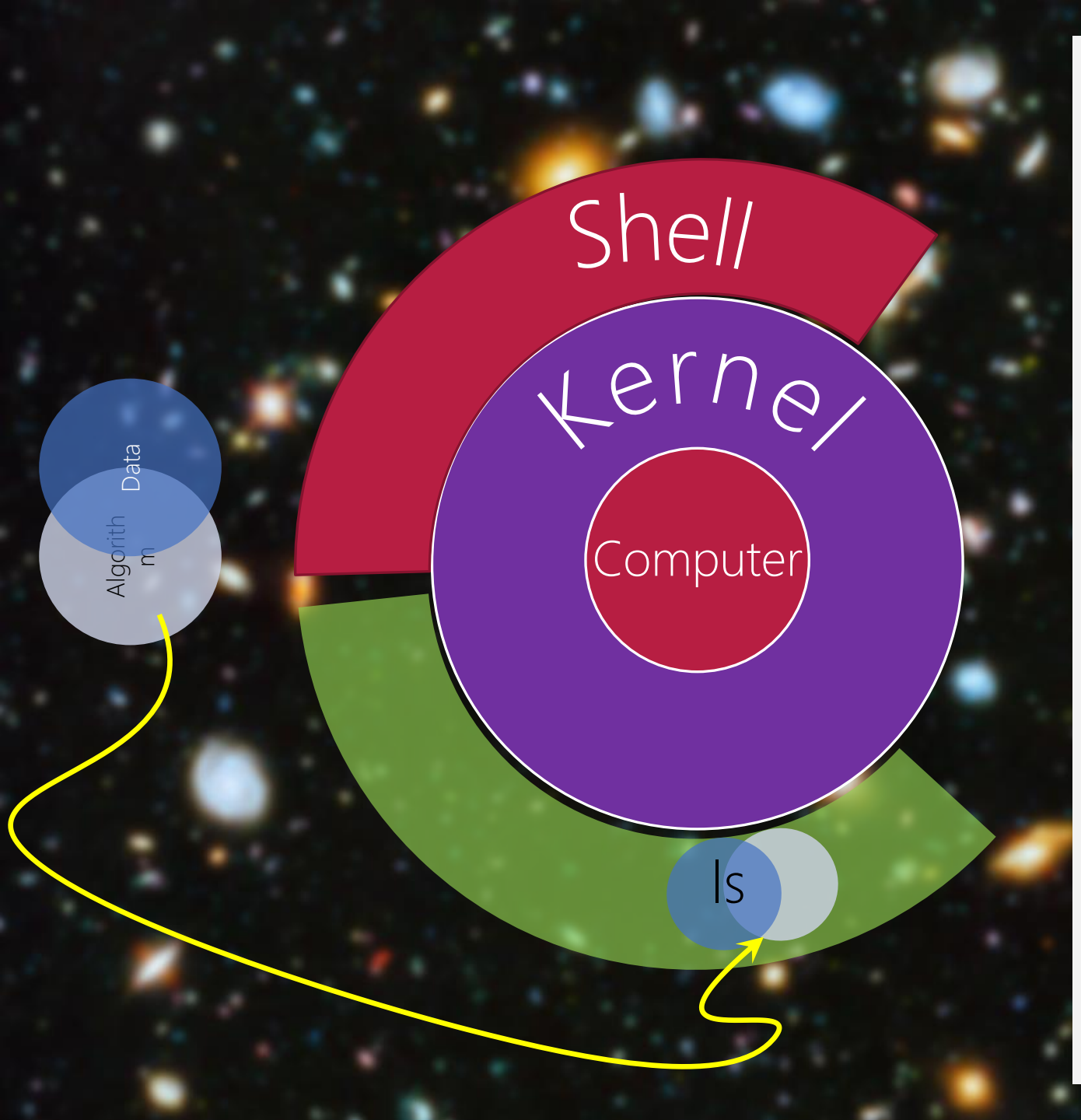
Shell

ls

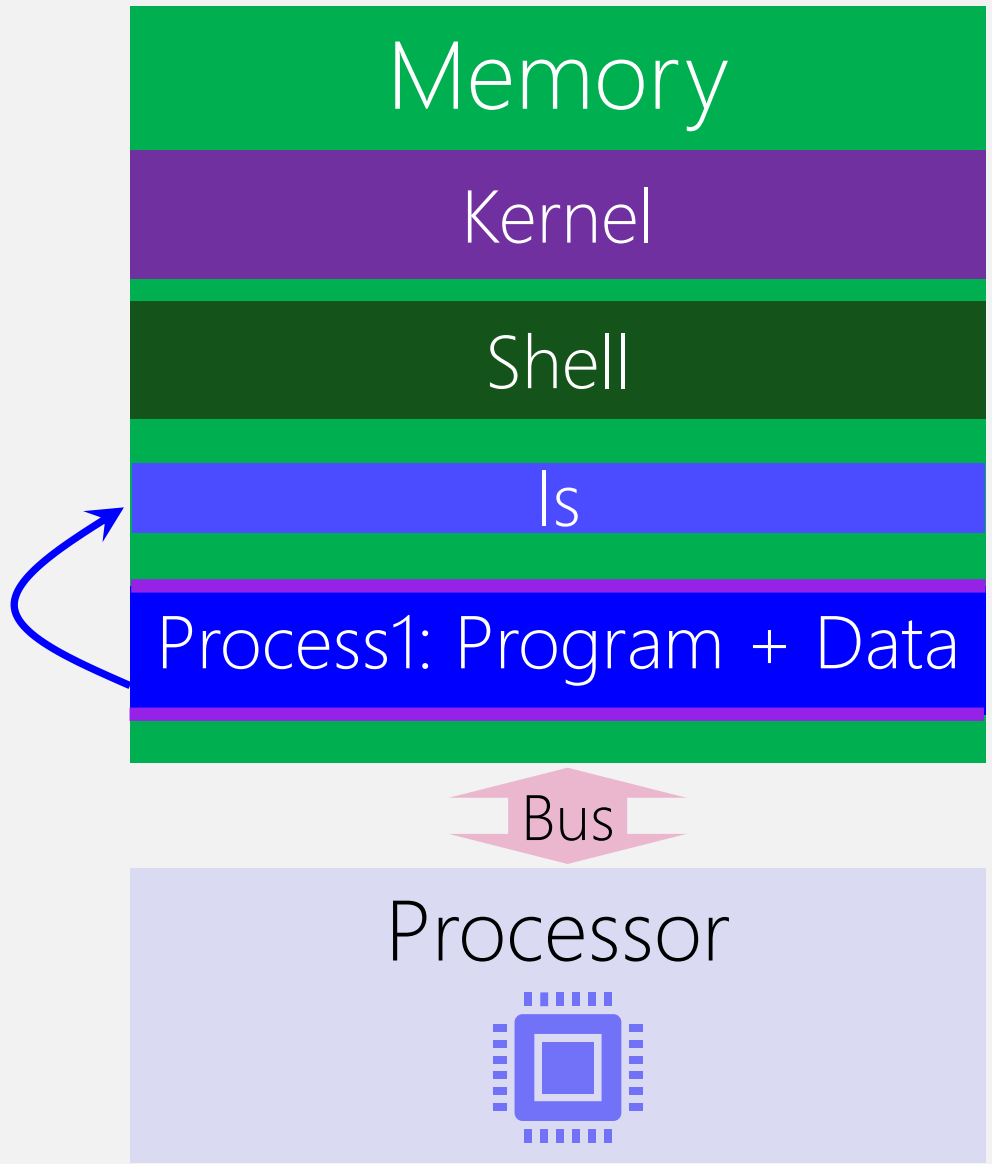
Process1: Program + Data

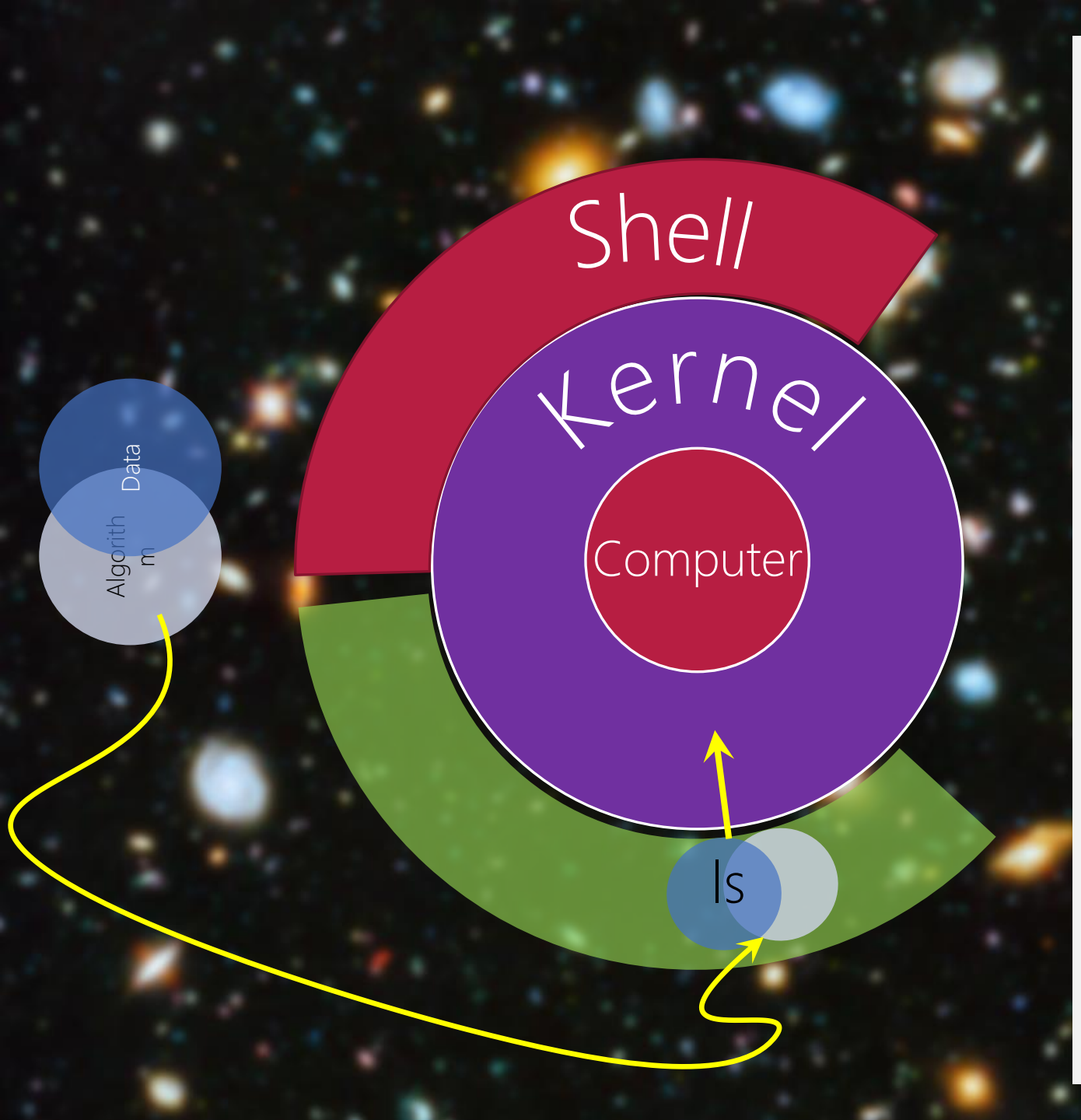
Bus

Processor

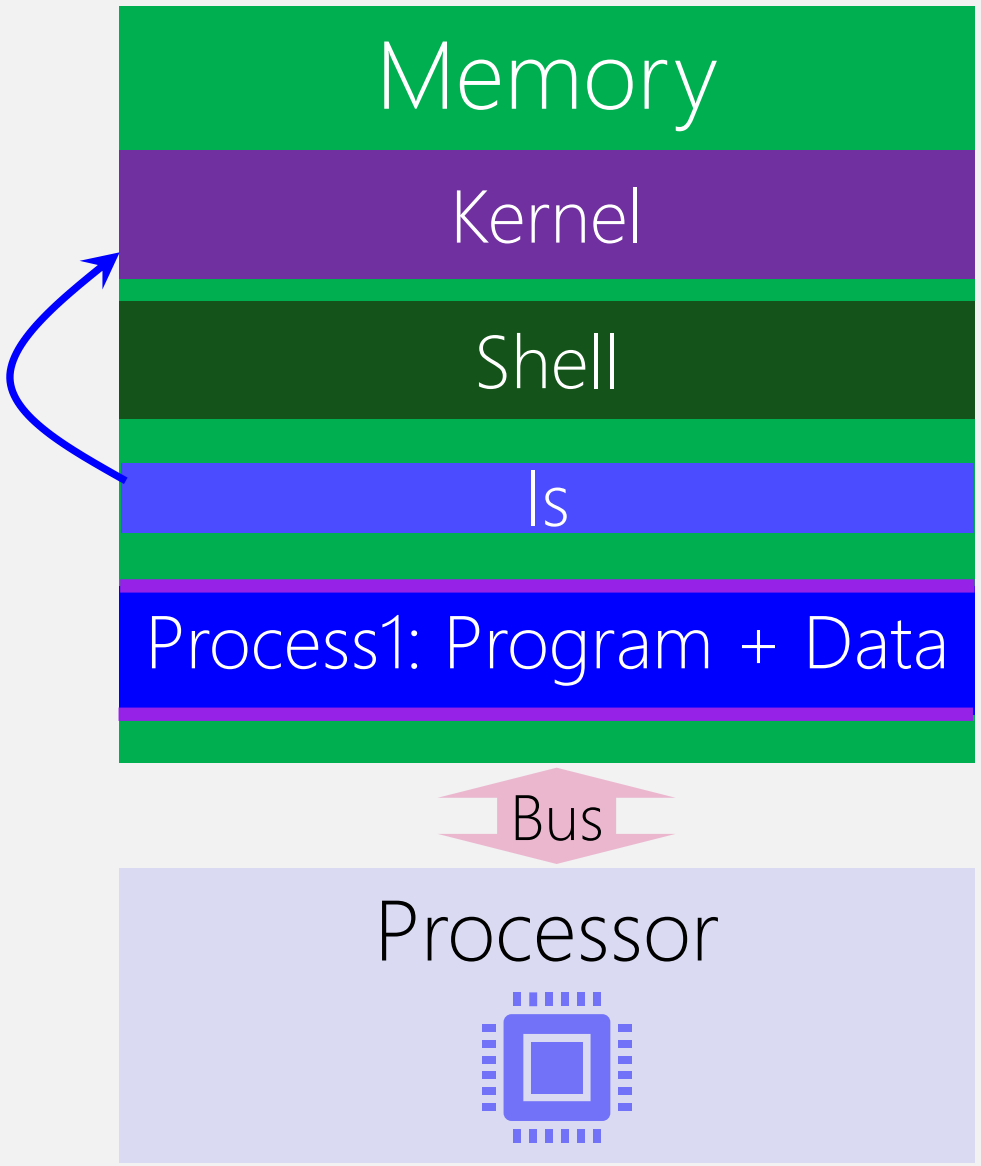

Shell
Kernel
Computer
Data
Algorithm
ls

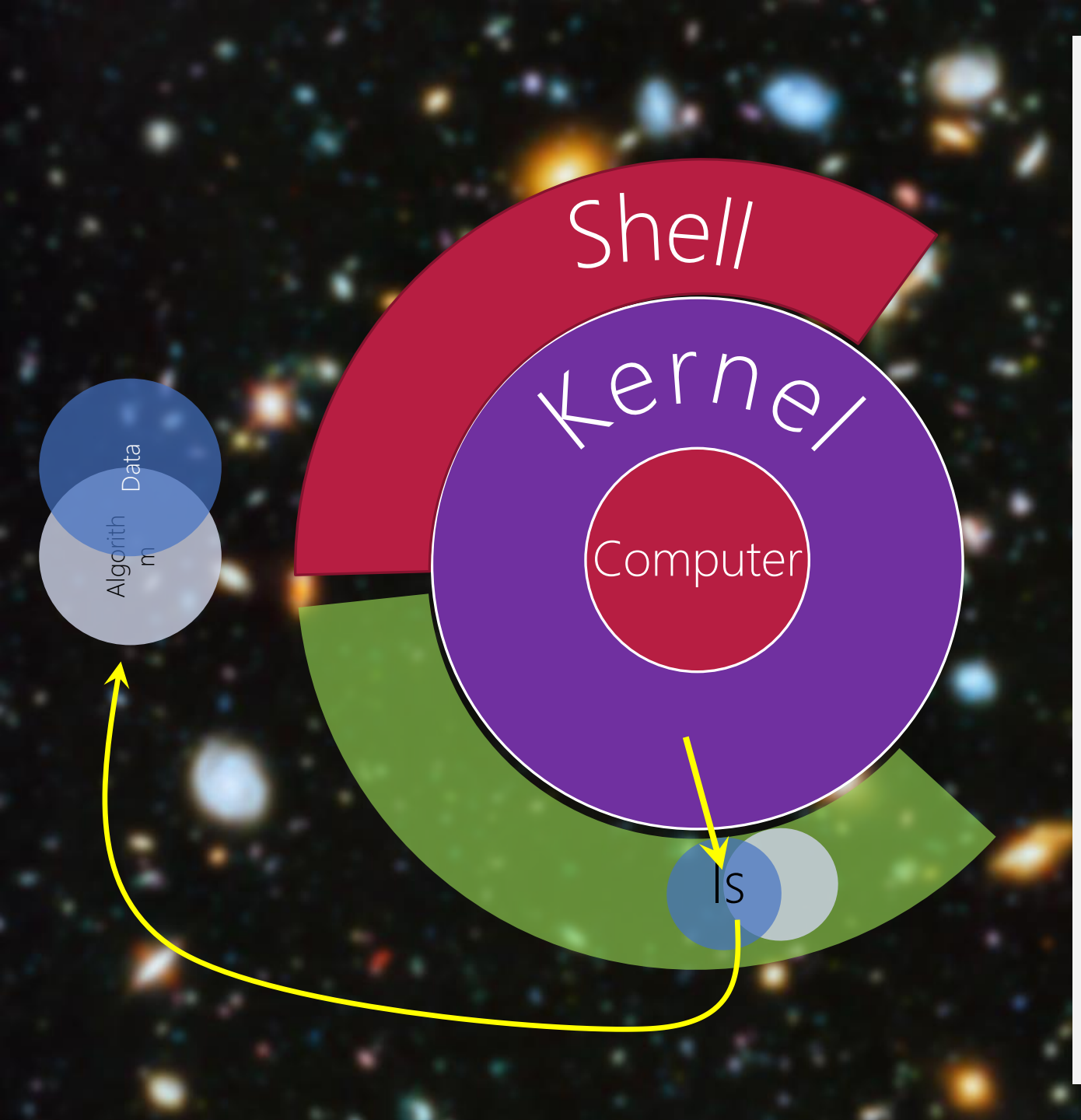
Computer


Memory
Kernel
Shell
ls
Process1: Program + Data
Bus
Processor

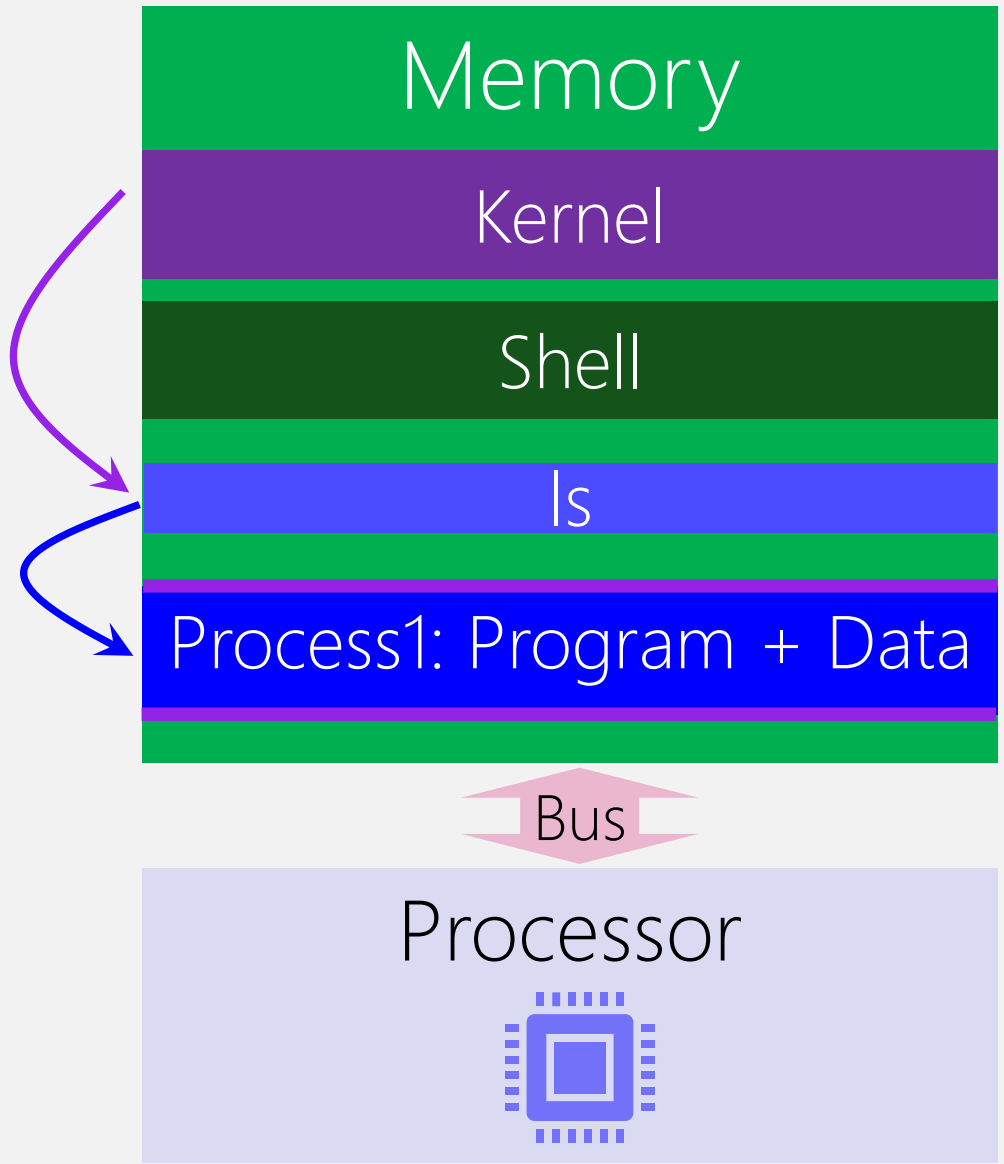

Shell
Kernel
Computer
Data
Algorithm
ls

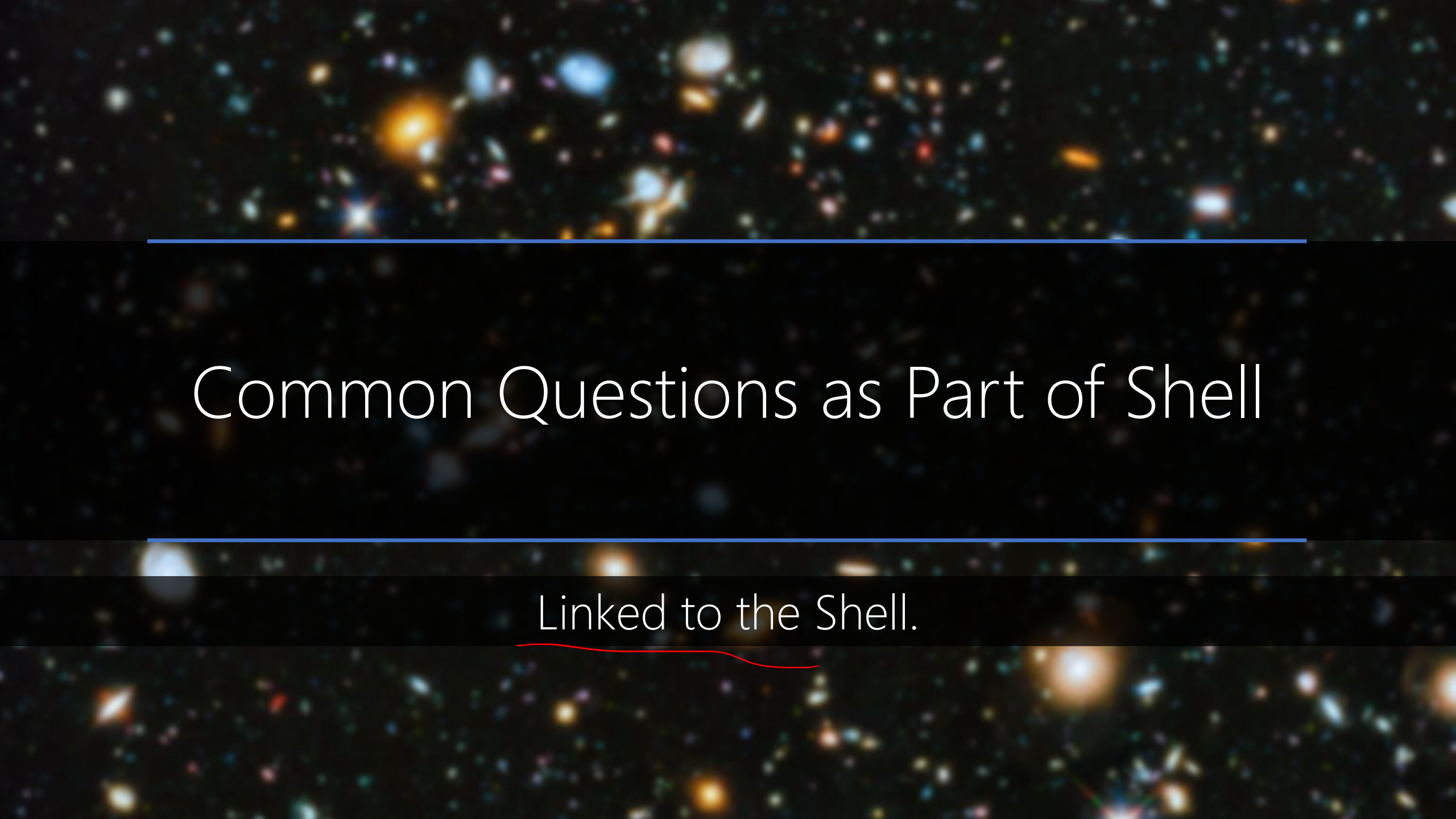
Computer


Memory
Kernel
Shell
ls
Process1: Program + Data
Bus
Processor


Shell
Kernel
Computer
Data
Algorithm
ls

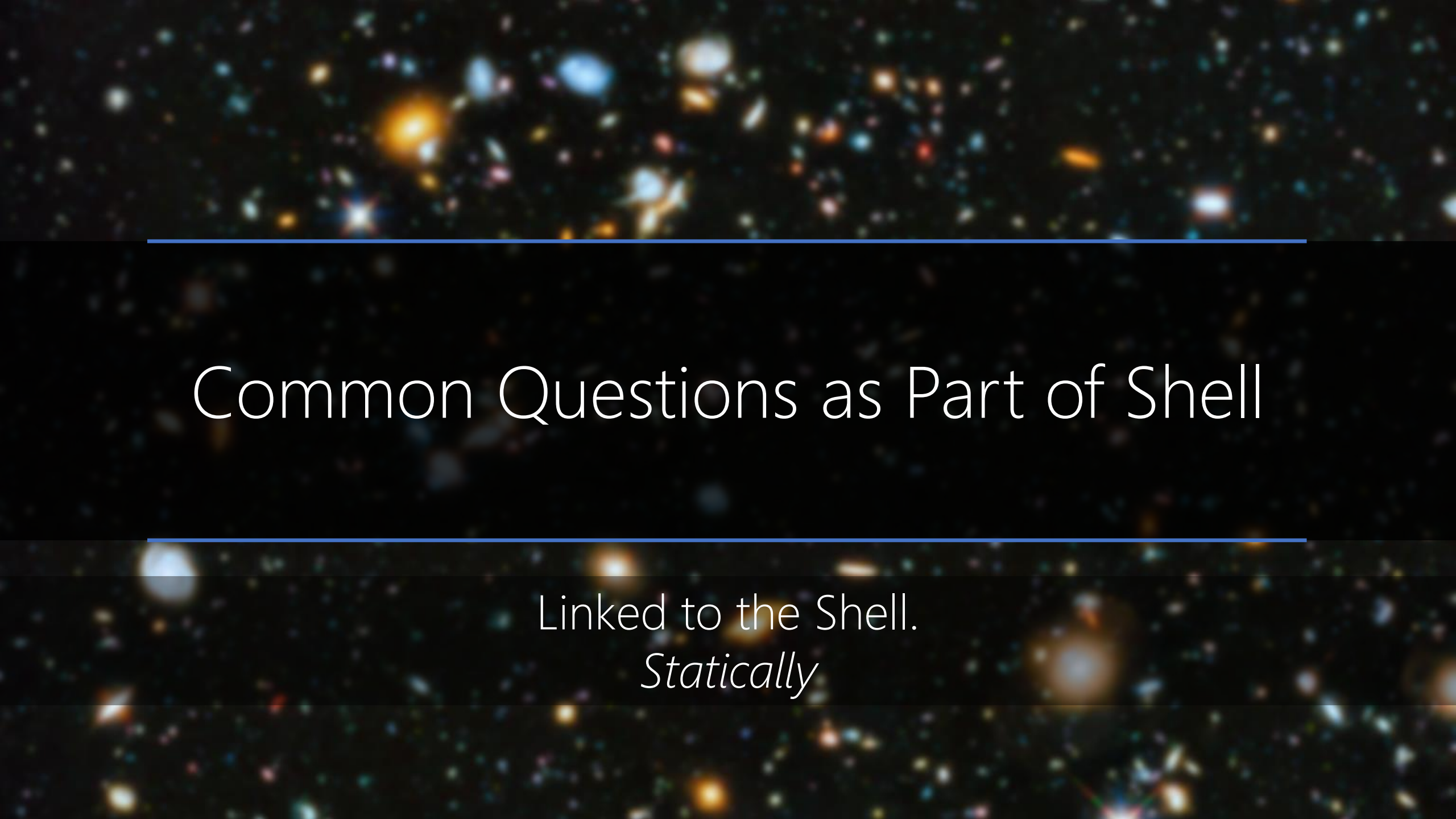
Computer


Memory
Kernel
Shell
ls
Process1: Program + Data
Bus
Processor



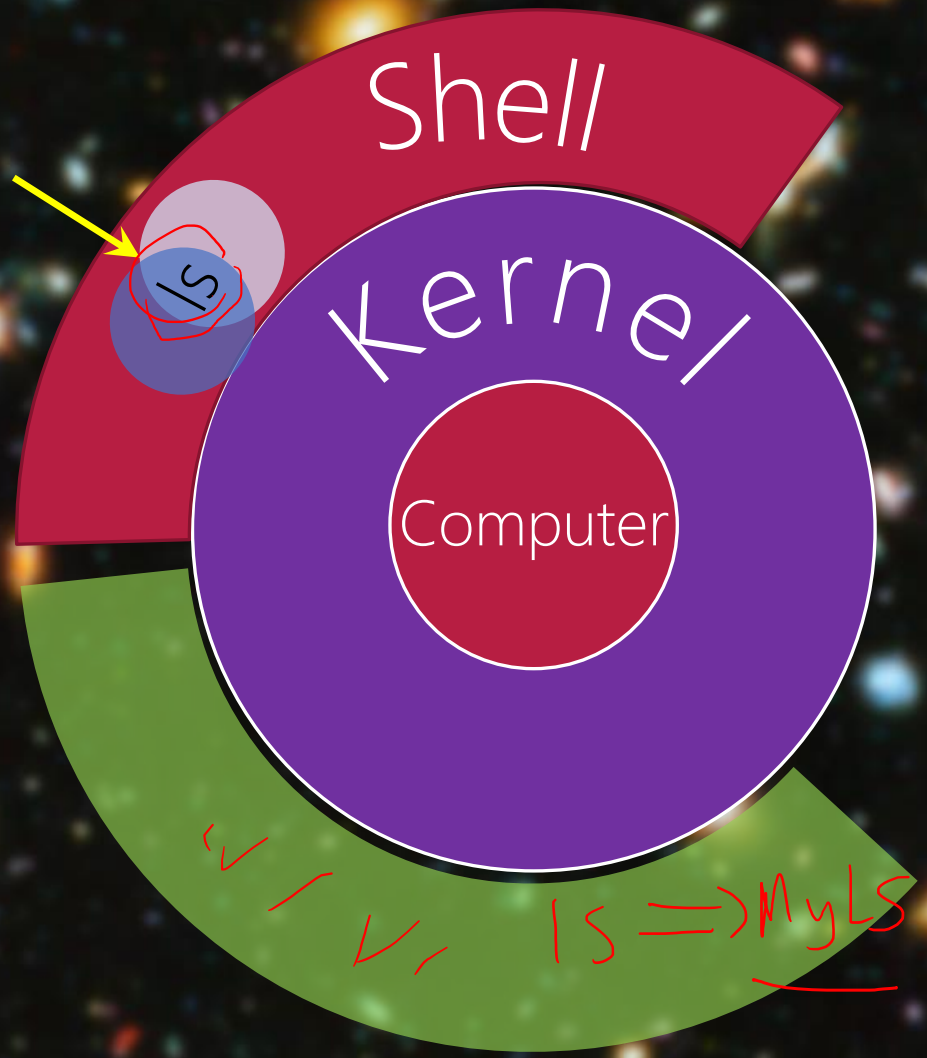
Common Questions as Part of Shell

Linked to the Shell.

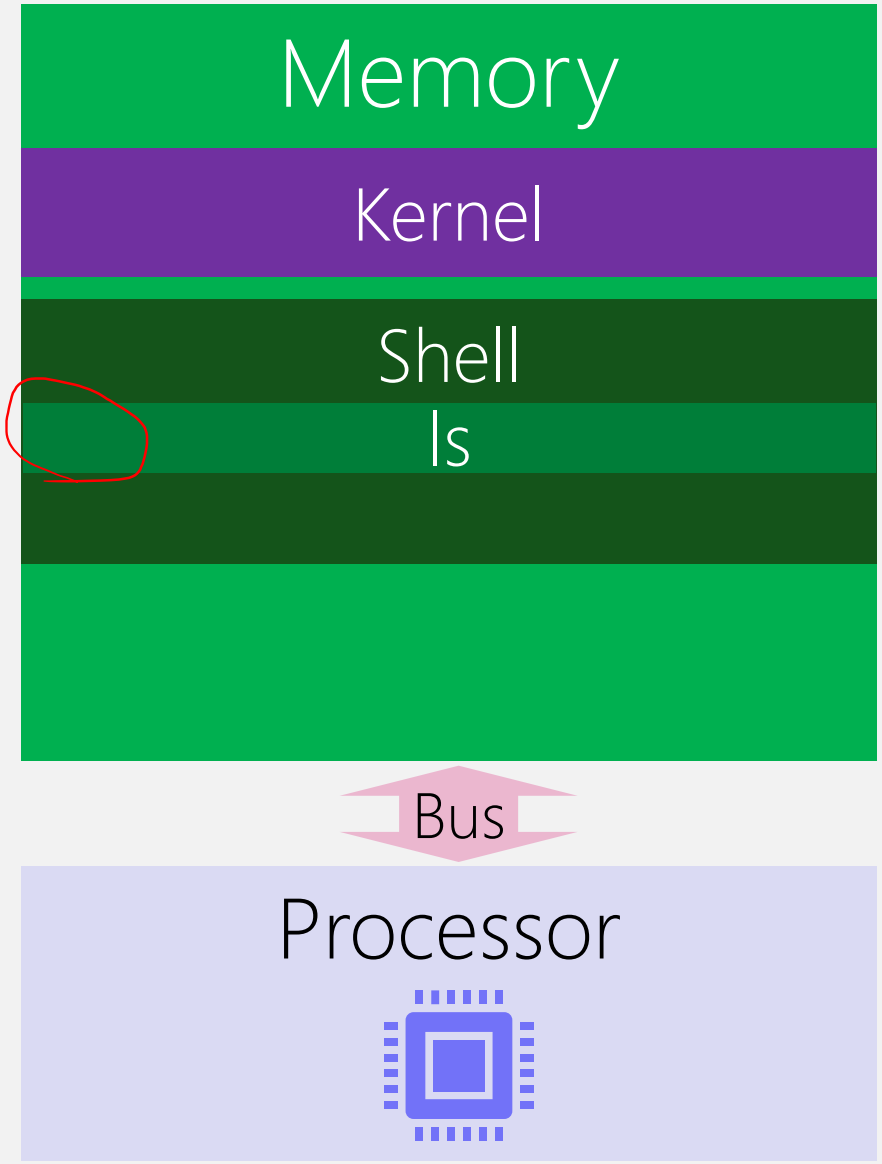


Common Questions as Part of Shell

Linked to the Shell.
Statically


Shell
ls
Kernel
Computer
ls =>MyLS

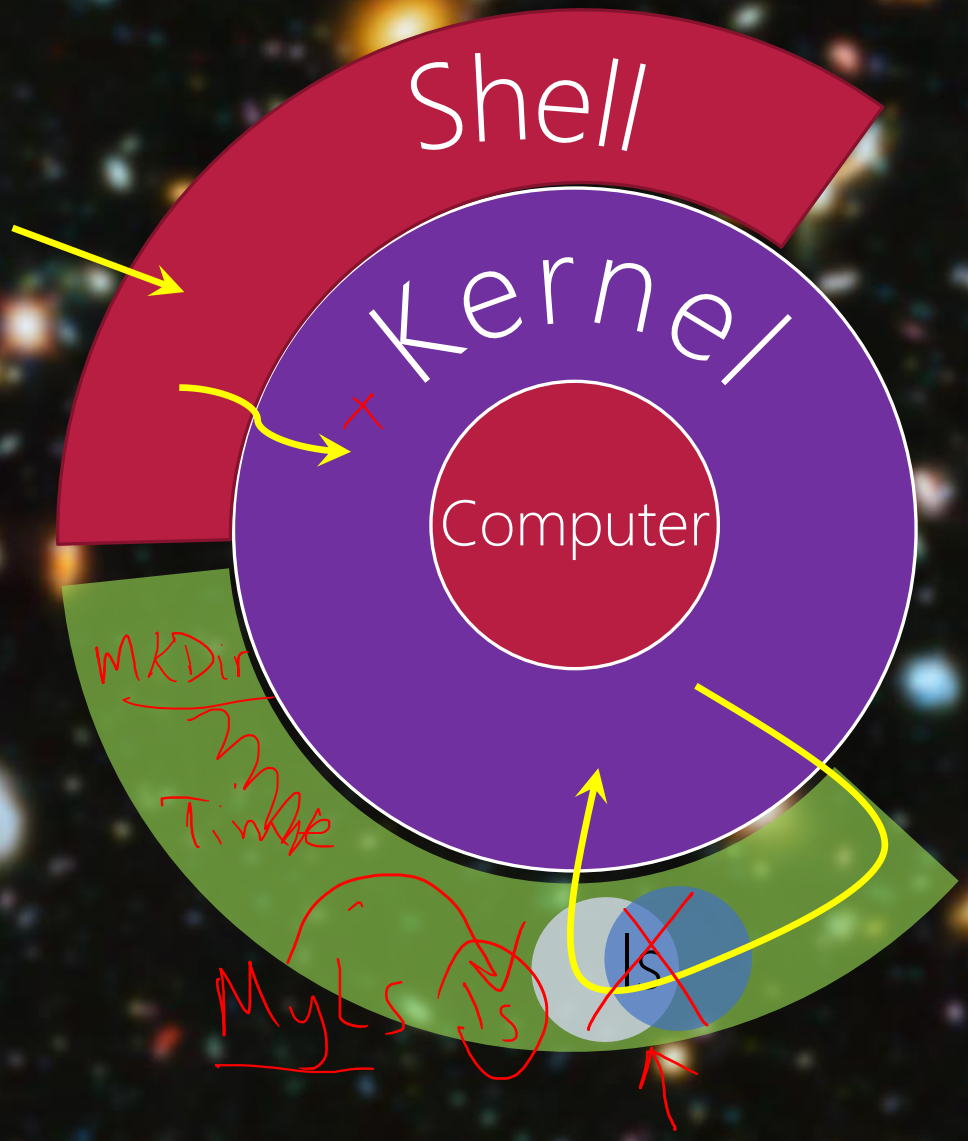
Computer


Memory
Kernel
Shell
ls
Bus
Processor

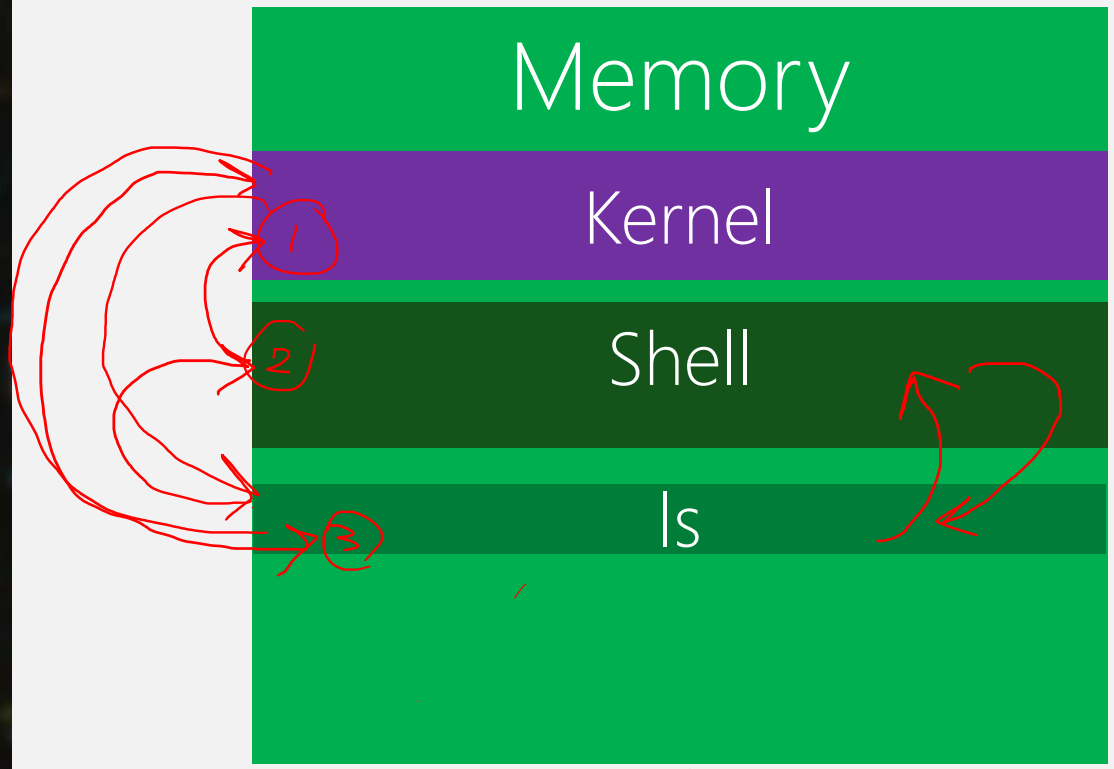
Common Questions as Part of Shell

Linked to the Shell.

→ *Dynamically*

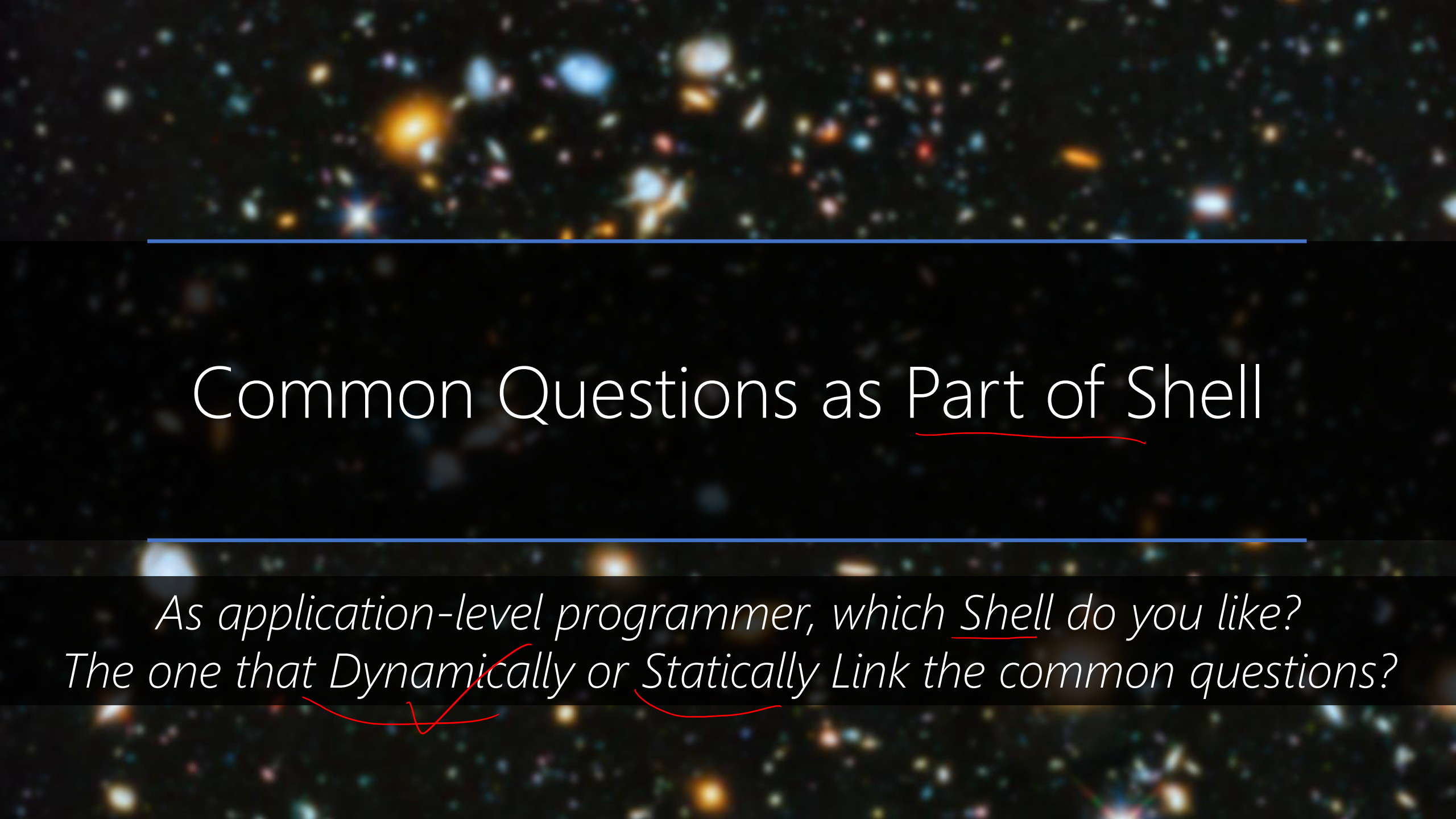


Computer



Bus

Processor

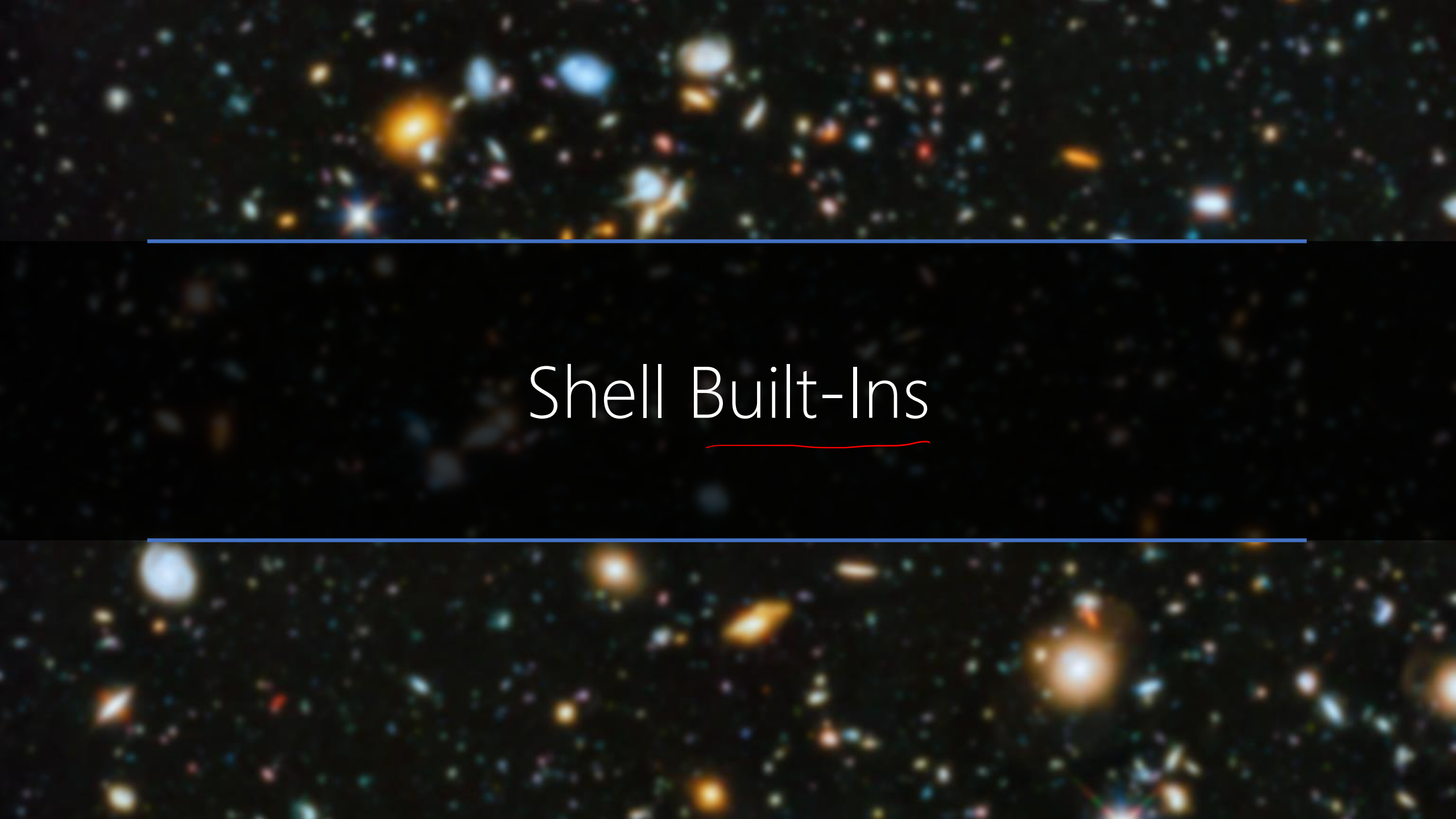


Common Questions as Part of Shell

As application-level programmer, which Shell do you like?
The one that Dynamically or Statically Link the common questions?

```
hfani@alpha:~$ ls
```

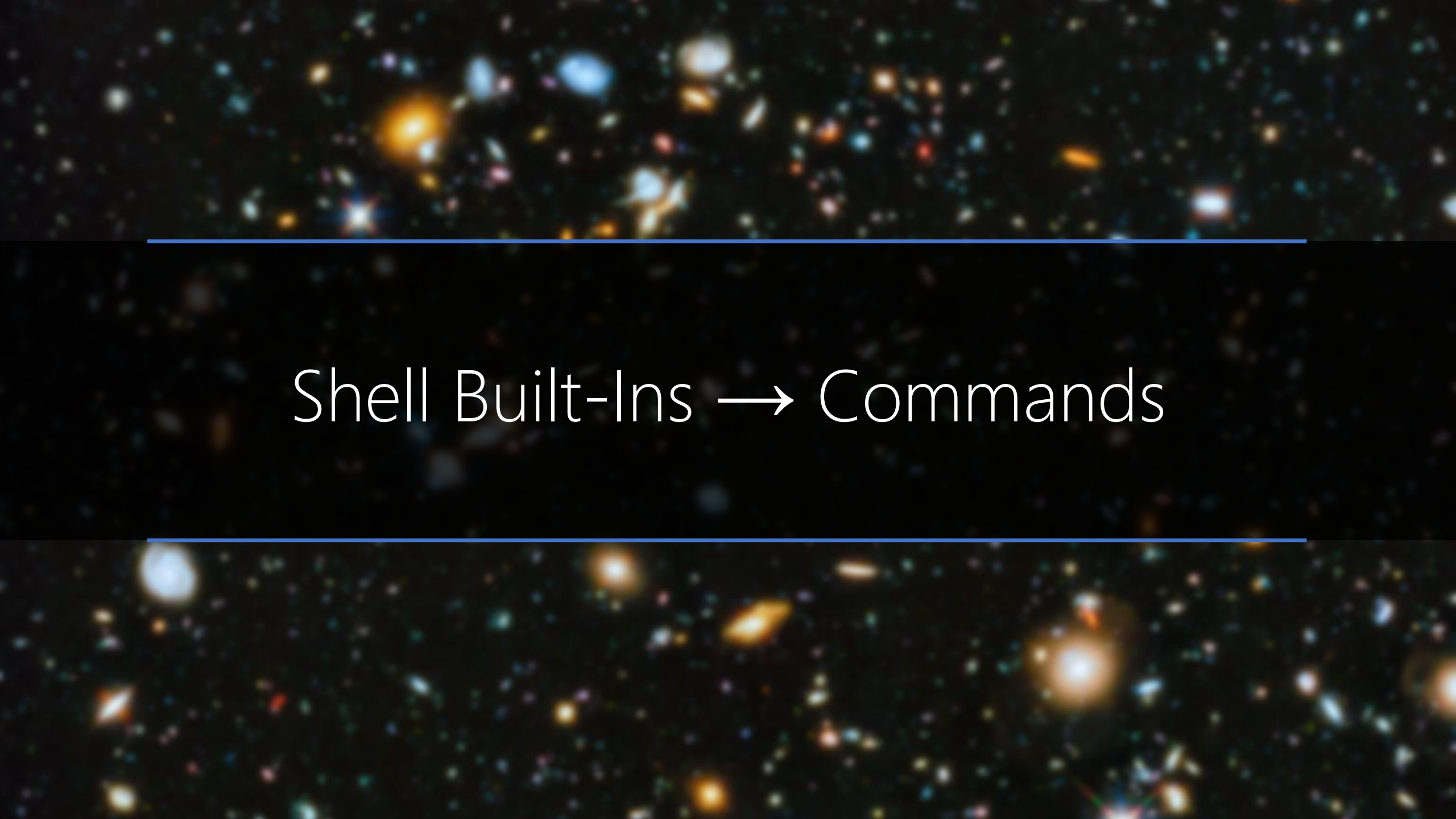
```
hfani@bravo:~$ ls
build_lab03.sh  Downloads          hello.c       increment.o  main    main.s       Pictures   test.sh
Desktop         eclipse-workspace  hellocat.txt  increment.s  main.c  Music        Public     Videos
Documents       hello              increment.c   ls.c         main.o  myscript.sh  Templates
hfani@bravo:~$ ls /
amd64_packages  dev     home           lib    libx32      media        opt   run   srv  usr      vmlinuz.old
bin             etc     initrd.img     lib32  local       mnt          proc  sbin  sys  var
boot            global  initrd.img.old  lib64  lost+found  nonexistent  root  seed  tmp  vmlinuz
hfani@bravo:~$ ls /home
60214      chen1fl    gill142    kobtic     nawazn     rebkowes     swayzed
60280      chen1k2    gill149    kogelt     nazal      reelyt       syalo
60311      chen2d     gill14l    kohli111   ndiayep    reen         syed11r
aanku      chen74     gill15q    kohli8     ndibanj    reid137      syed126
abbas51    chen9i     gill22     kojicl     nealj      reingolv     syed129
abbasit    cheng134   gill59     kollarm    necioo     rempilljj    syeda1
abbiedyck  cheng143   gill95     kollere    needsm     renau11s     syedai
abdall51   cherkup    gilliamd   komarin    neisari    renau121     syedmuz1
abdelba    cherryn    giorlan1   kondepu    nesarajm   renaud2p     sylvest7
abduelm1   chertova   giwaf      kongw      newto115   renaud2y     symonsl
abdull1u   chhabrat   gliu       kooplic    ng122      renaud91     szeen
abdull1v   chhabri    gloria     kopco111   ng12a      repmannc     szuckia
abdulghh   chiangb    glovern    kopliku    ngaiv      reynold5     szucs1
abdulnan   chiarcod   godavars   koratk     ngo113     rezaei       tabbeno
abesi      chibuzo    godfather  korkis1    nguy       rezaeia      tadros2
abeygunj   chikhalt   godhania   koshiyap   nguye115   rezaz        tahan
abidaf     chikkamb   godin      koshtit    nguye141   rgras        tahay
abouali    chittle5   godlewso   koshula    nguye145   riaz9        taher5
abouass    chittleb   gomezi     kotha111   nguye14b   ricci6       taifour
abouelg    chohanu    gomezma    kotha113   nguye159   rice118      tailo115
aboughat   chokshia   gongj      kothapa1   nguyen1v   rice11c      tainga
aboughm    cholaghc   gongo      kothapap   nguyen2r   richar63     takacha
abouha11   chopdan    goodisoc   kottoort   nguyen43   richard      takamorr
abrah111   chopr11b   goodm111   kouhang    nguyen4e   richard.arsenault  takev
```



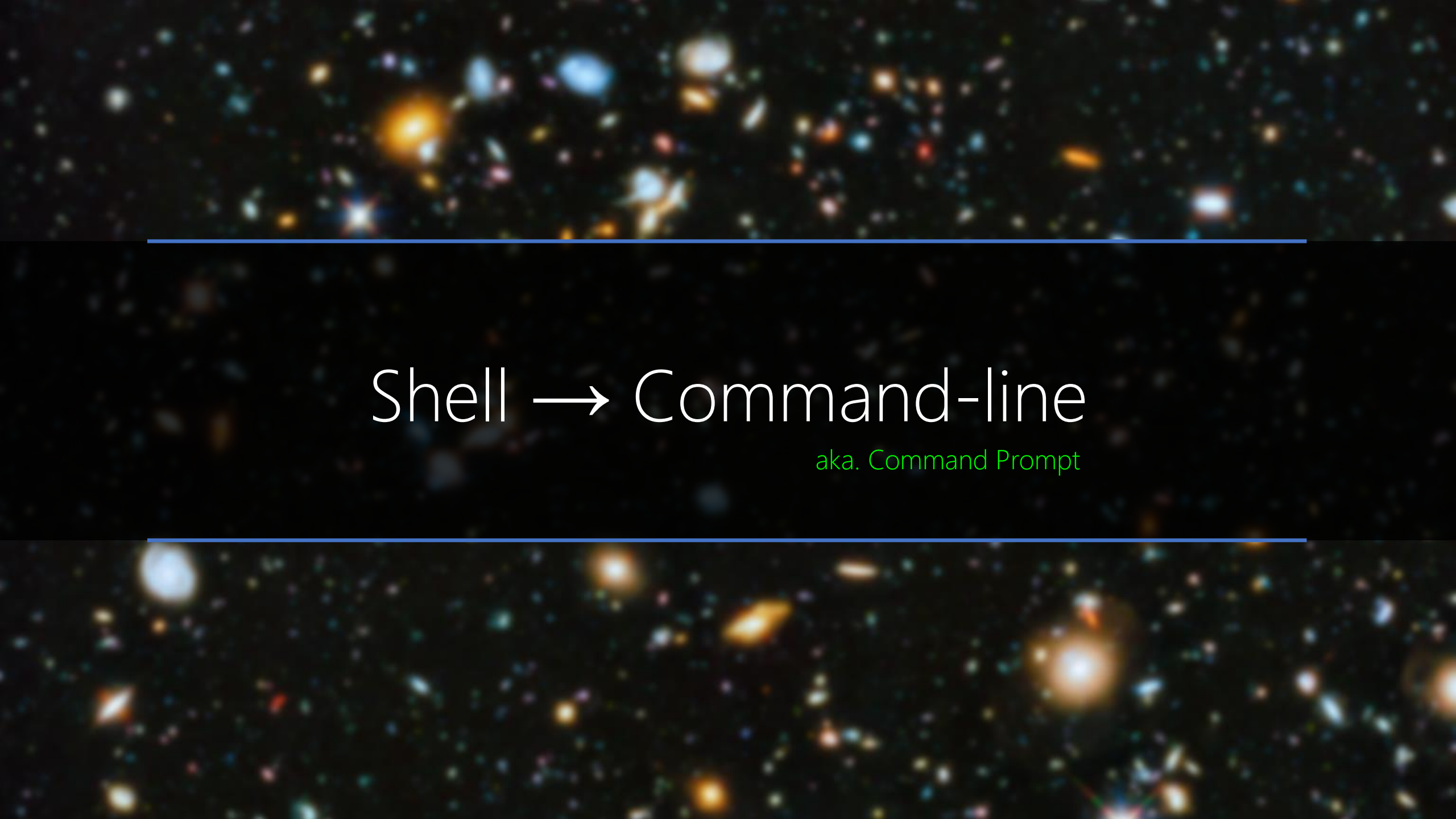
Shell Built-Ins

Not Only Questions But Also Commands

Delete this file, please!
Remove this folder and all its content, please!
Make this file hidden, please!
Move this file from location A to B, please!
Copy this file from location A to B, please!
...



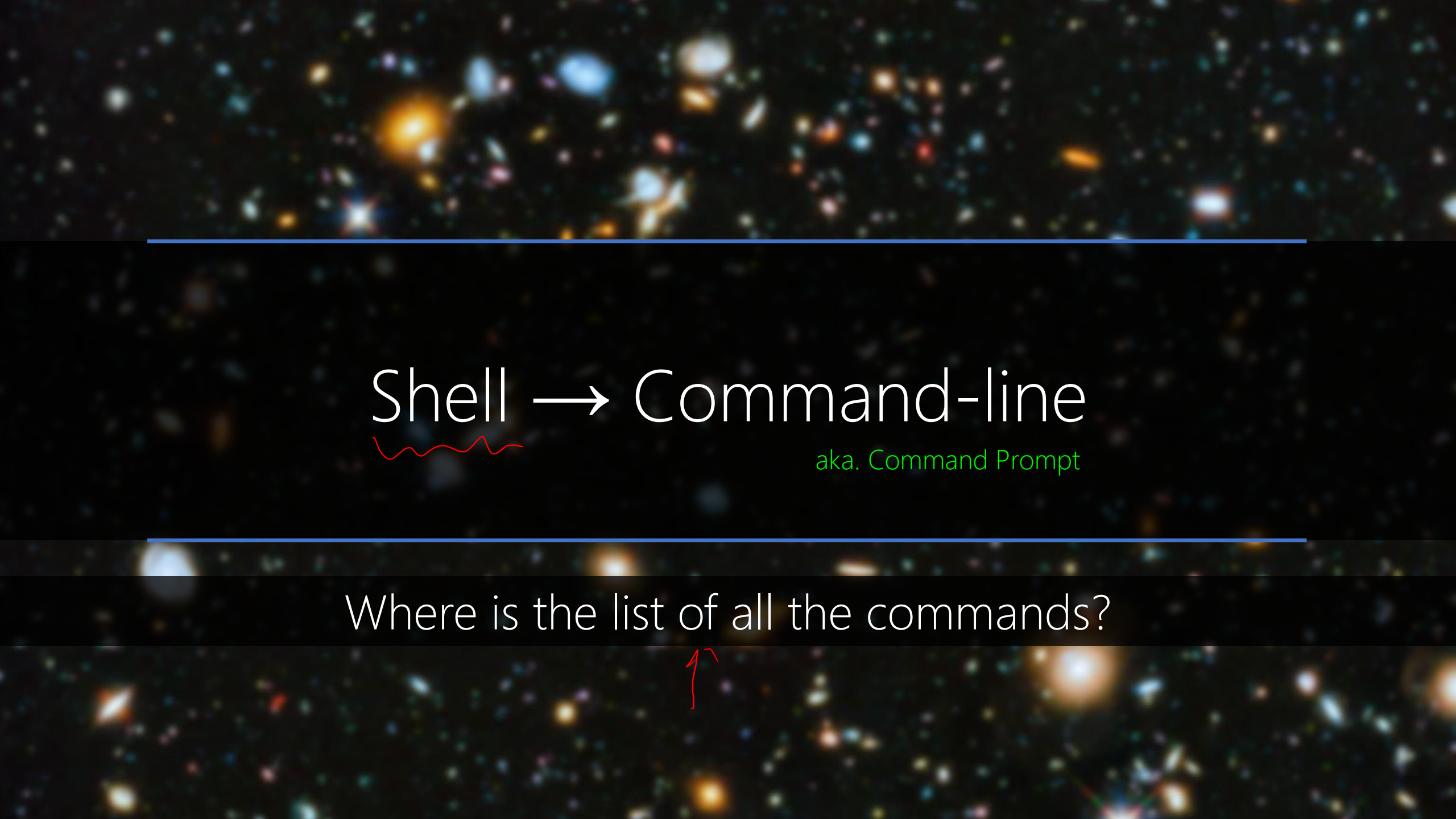
Shell Built-Ins ⟶ Commands

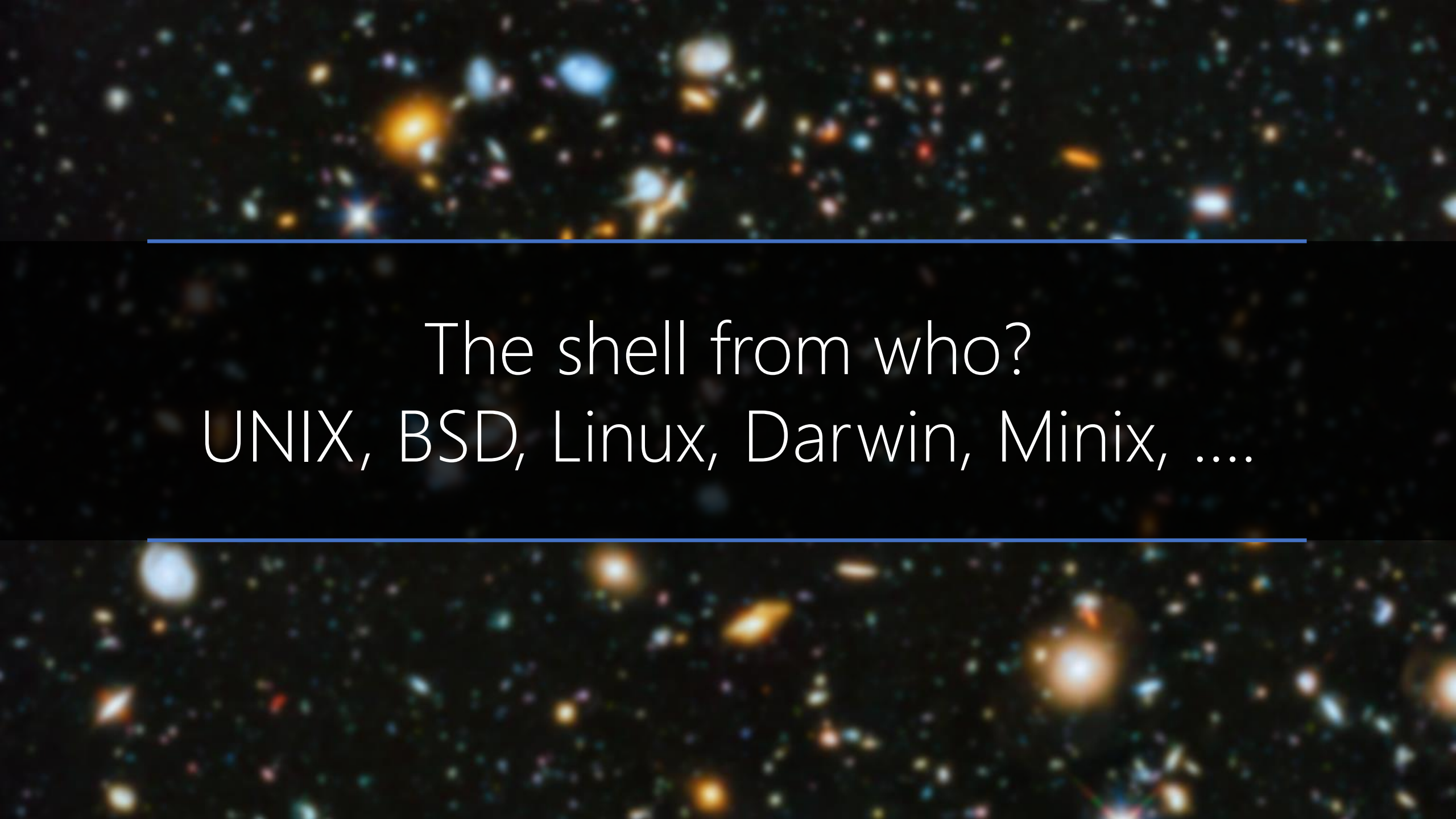


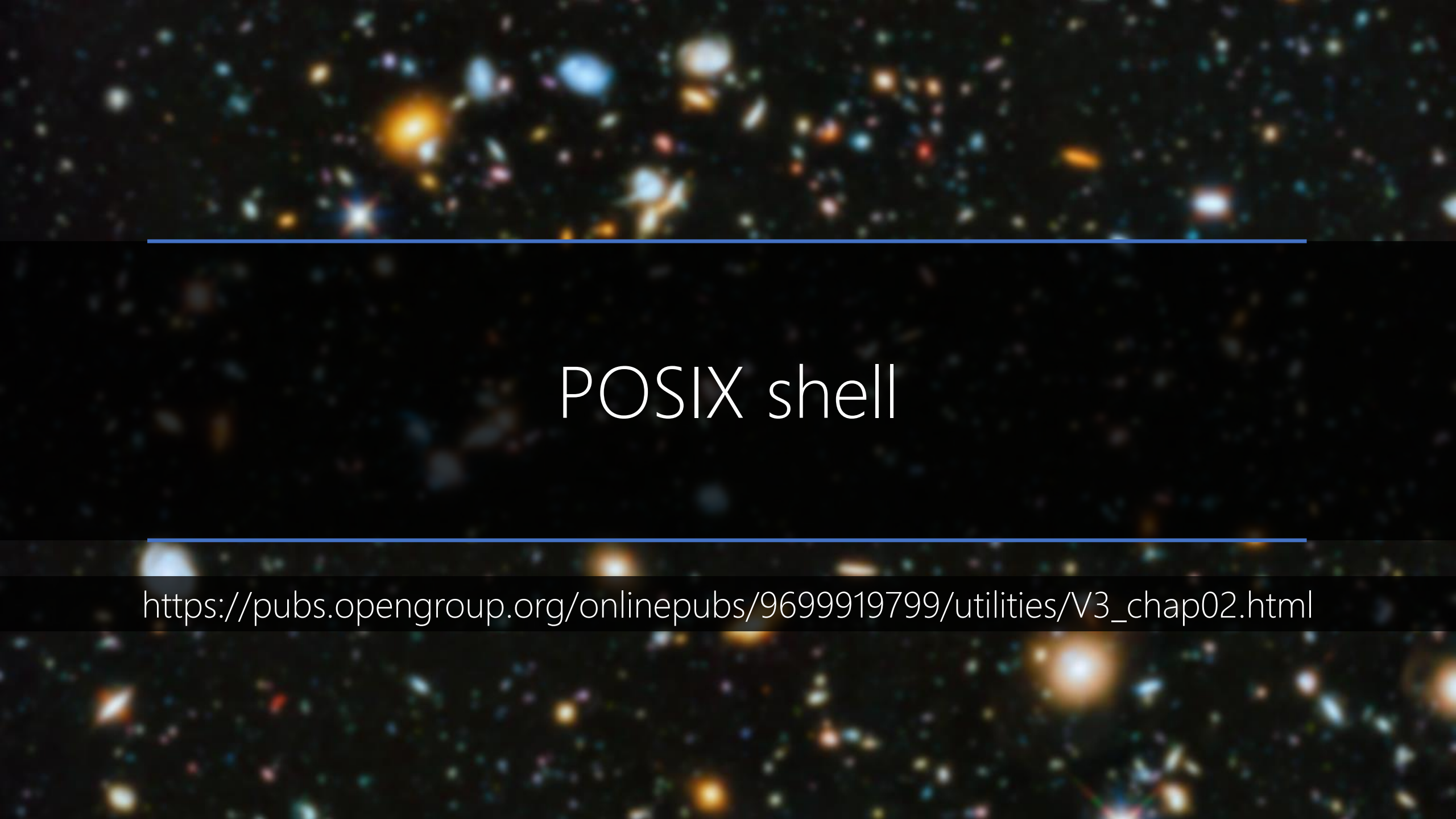
Shell → Command-line

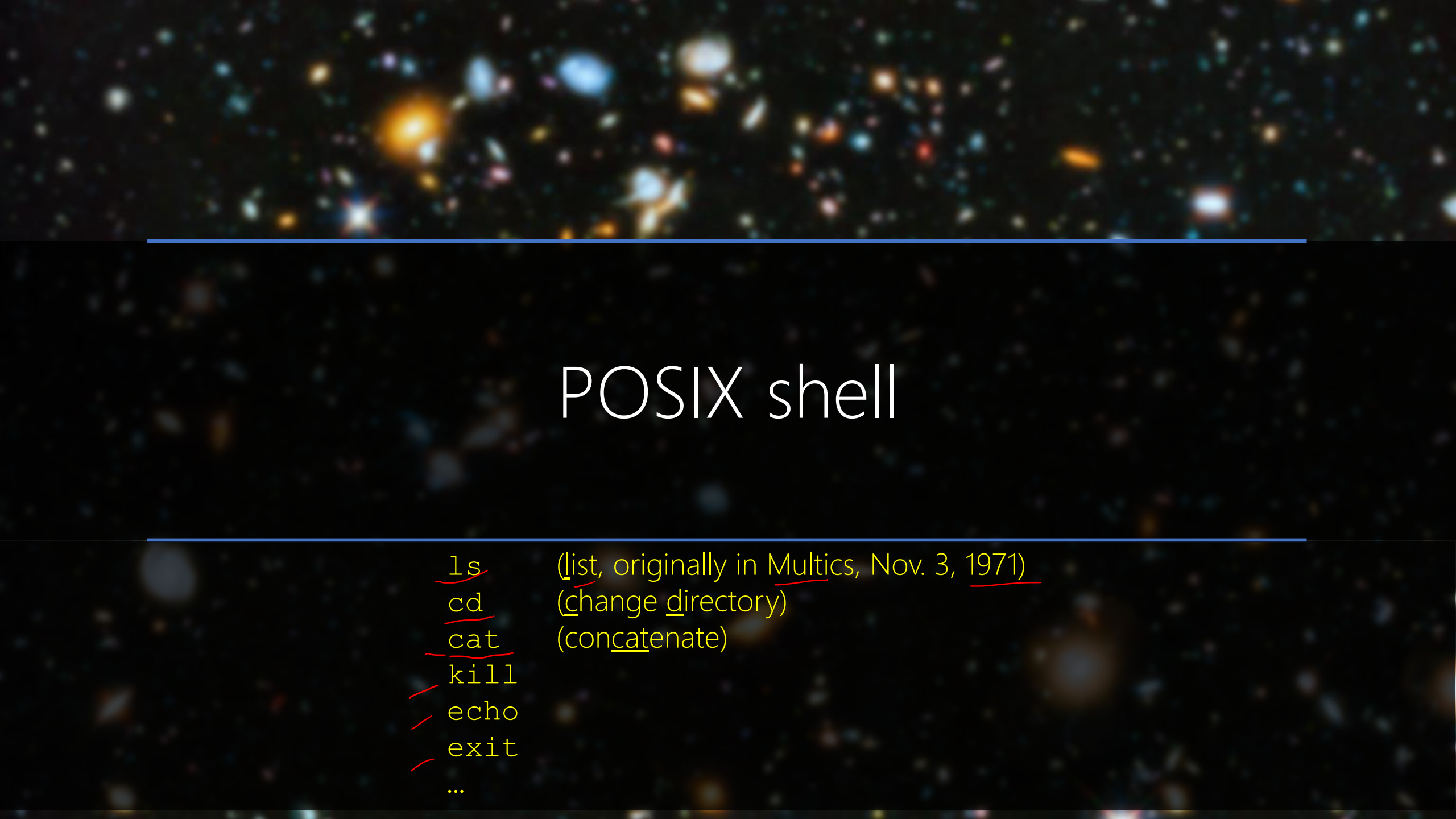
aka. Command Prompt

```
hfani@alpha:~$ ls
```


Shell → Command-line
aka. Command Prompt
Where is the list of all the commands?


The shell from who?
UNIX, BSD, Linux, Darwin, Minix, ...


POSIX shell
https://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html



POSIX shell

`ls`	(list, originally in Multics, Nov. 3, 1971)
`cd`	(change directory)
`cat`	(concatenate)
`kill`	
`echo`	
`exit`	
…	

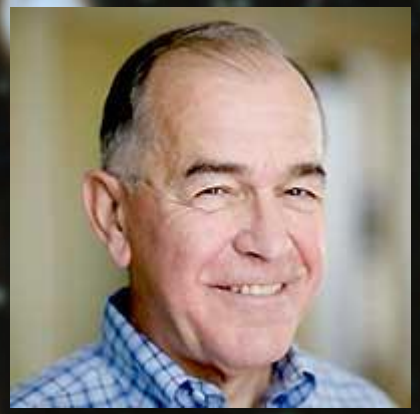
POSIX shell

ls =>
echo

https://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html

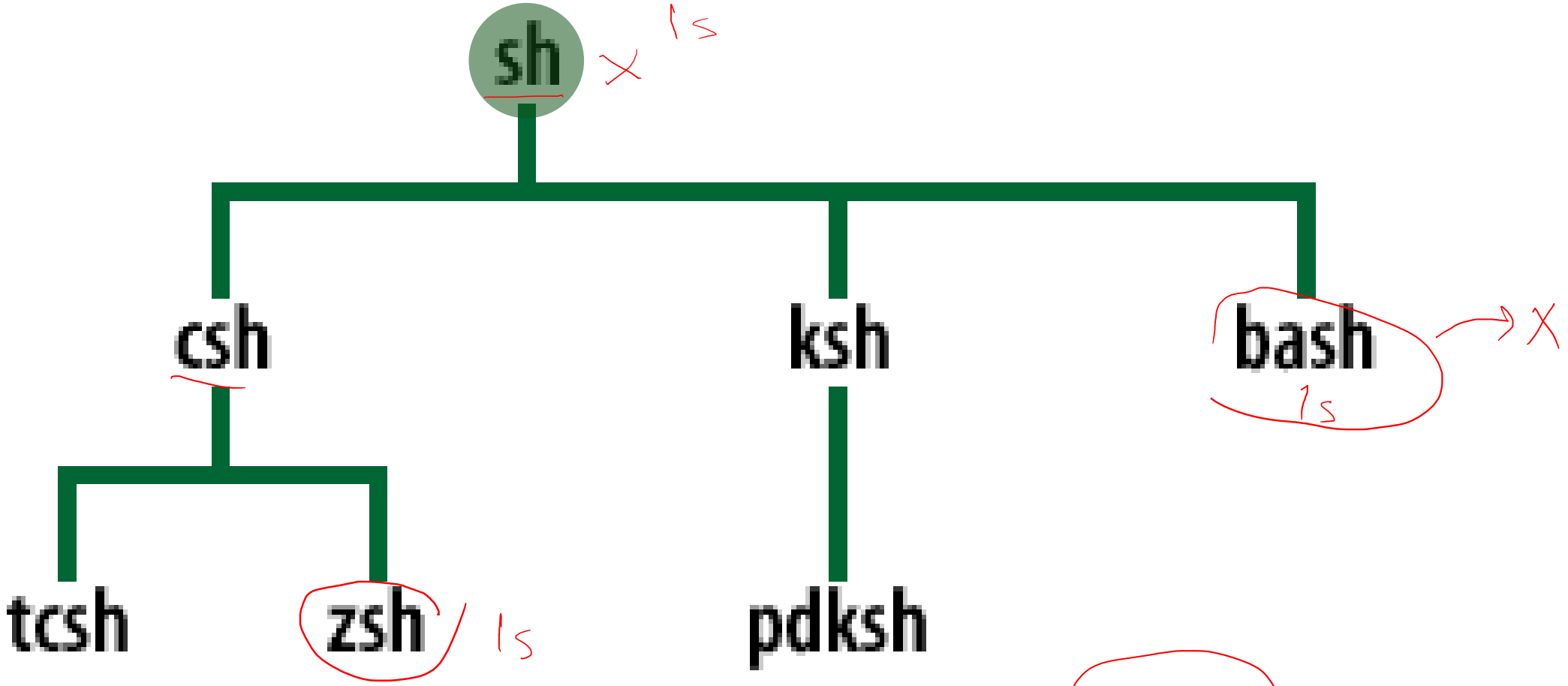
Does it have actual codes (C/ASM/Opcode) for the commands?

sh

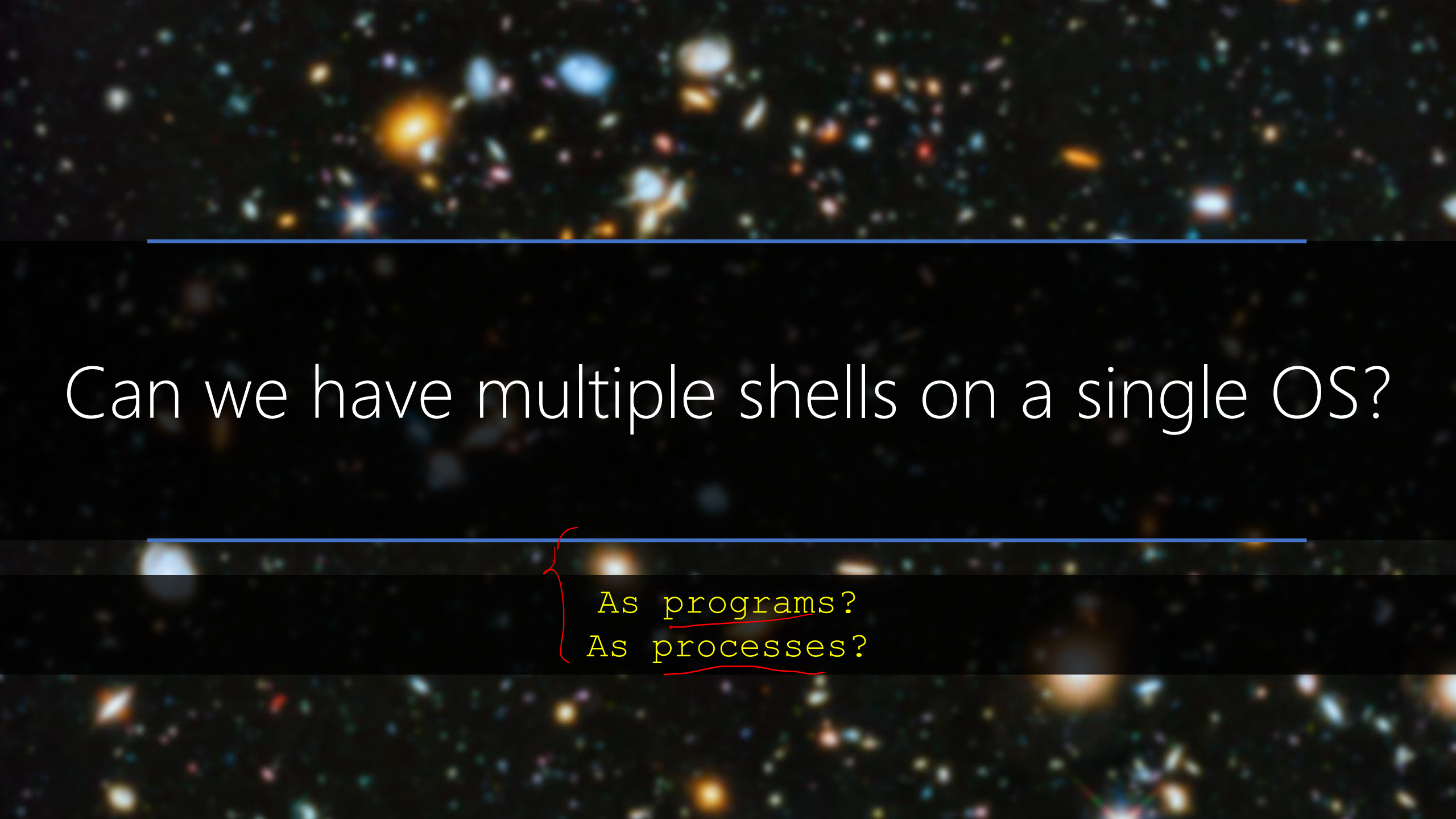


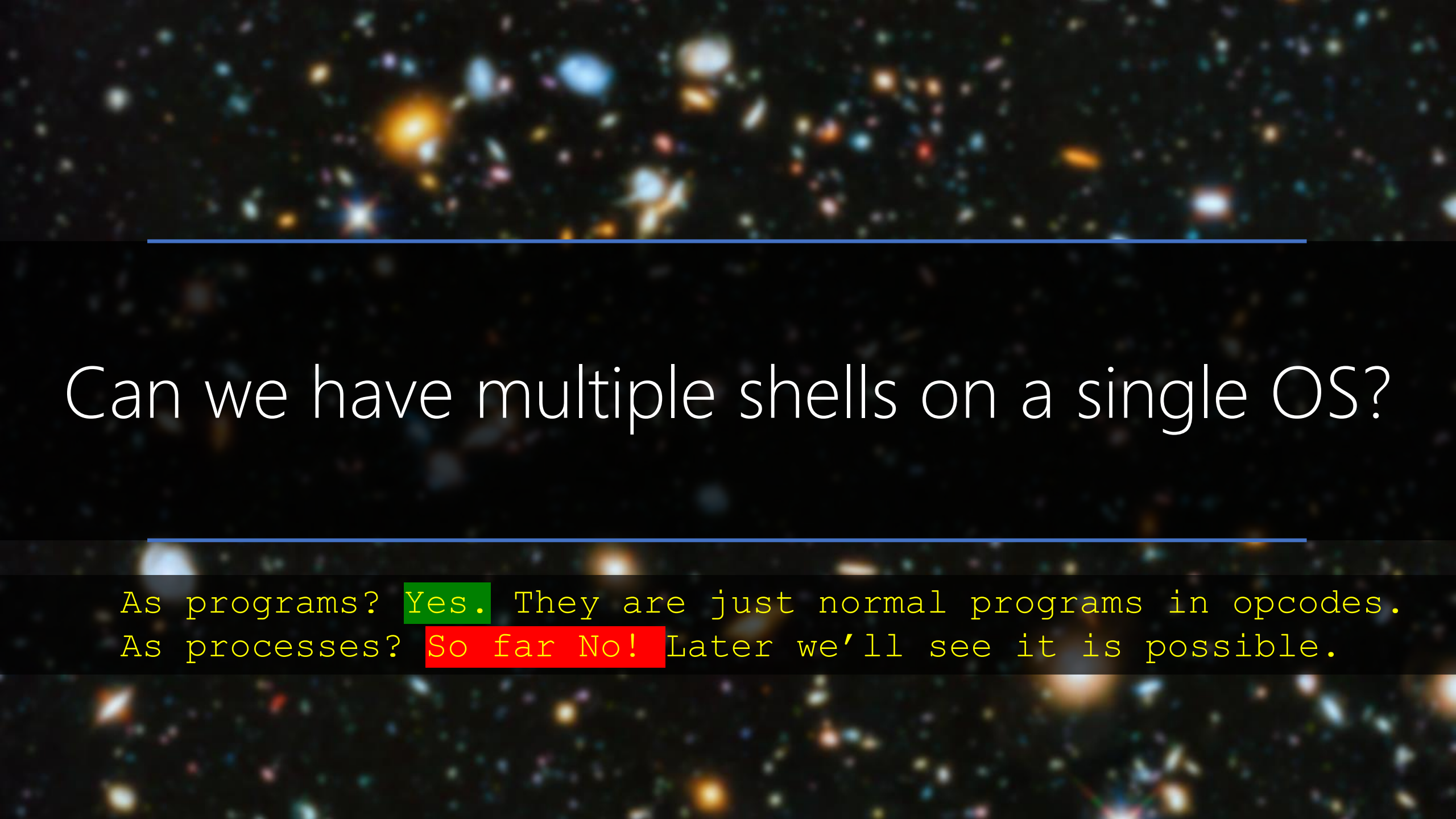
Stephen Richard "Steve" Bourne

- Bell Labs
- The first shell
- minimal, chosen as standard (POSIX)



Name	Path	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10
Bourne shell	`/bin/sh`	•	•	copy of `bash`	•
Bourne-again shell	`/bin/bash`	optional	•	•	•
C shell	`/bin/csh`	link to `tcsh`	optional	link to `tcsh`	•
Korn shell	`/bin/ksh`	optional	optional	•	•
TENEX C shell	`/bin/tcsh`	•	optional	•	•


Can we have multiple shells on a single OS?
As programs?
As processes?

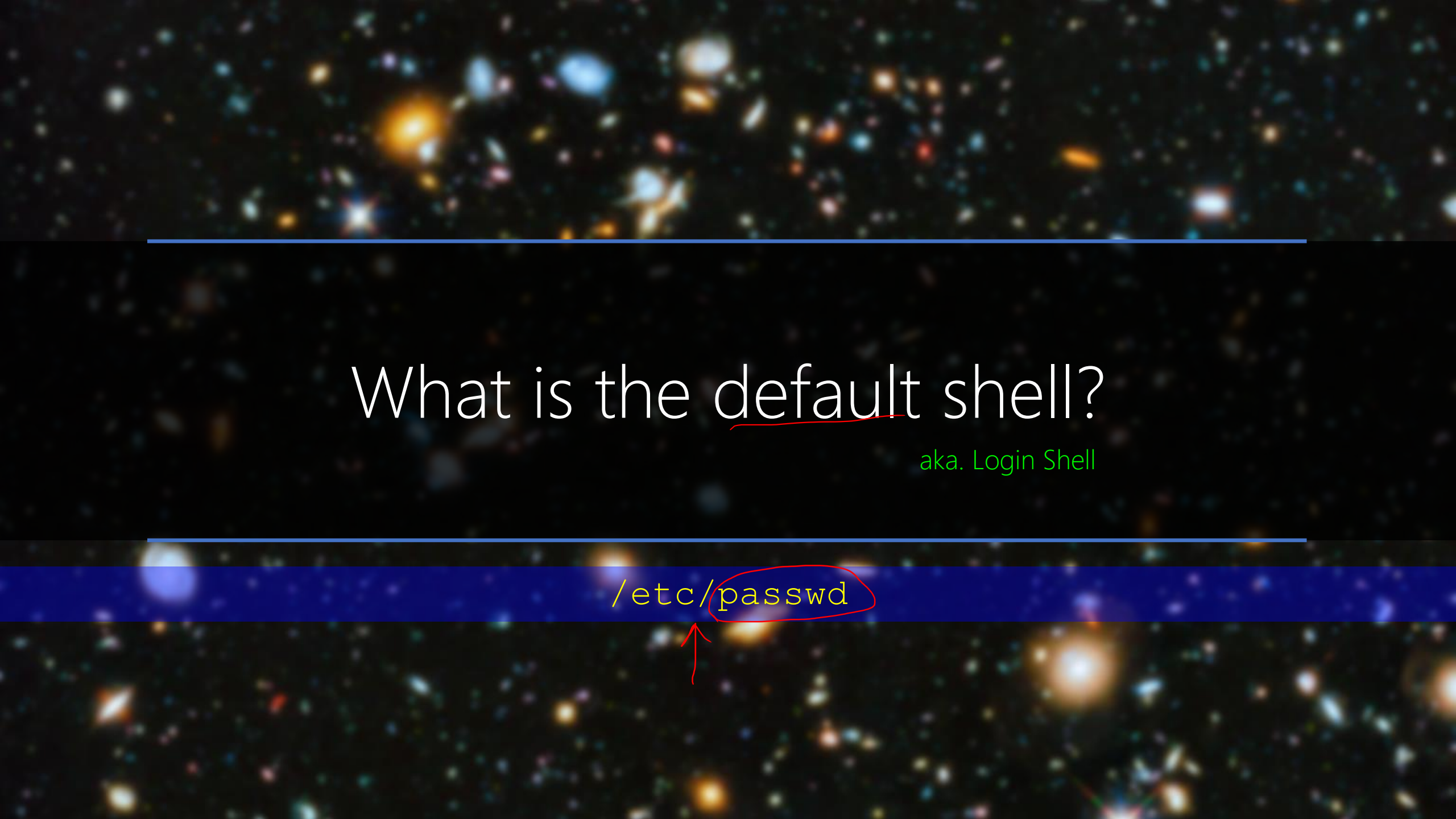


Can we have multiple shells on a single OS?

As programs? Yes. They are just normal programs in opcodes.
As processes? So far No! Later we'll see it is possible.


List of available shells?
/etc/shells

```
hfani@alpha:~$ vi /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/bin/tmux
/bin/mksh
/bin/mksh-static
/usr/lib/klibc/bin/mksh-static
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

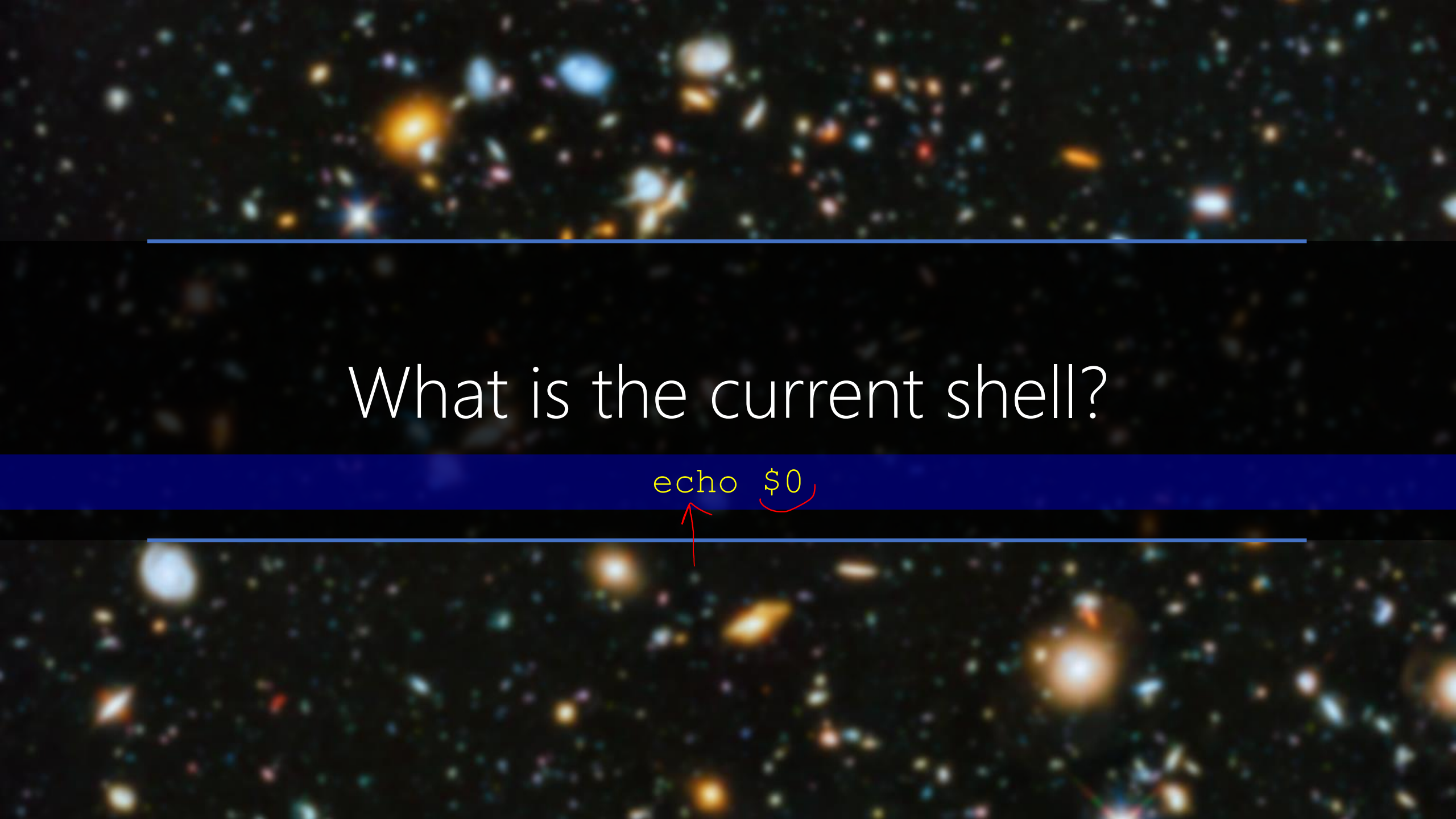

What is the default shell?
aka. Login Shell
/etc/passwd

```
hfani@alpha:~$ vi /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
messagebus:x:104:109::/var/run/dbus:/bin/false
pulse:x:105:110:PulseAudio daemon,,,:/var/run/pulse:/bin/false
avahi:x:106:114:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
Debian-exim:x:107:116::/var/spool/exim4:/bin/false
statd:x:108:65534::/var/lib/nfs:/bin/false
colord:x:109:120:colord colour management daemon,,,:/var/lib/colord:/bin/false
```

Default shell for the user root

Are these shell?!

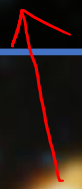
Where is Hossein's default shell?


What is the current shell?
echo $0

```
hfani@alpha:~$ echo $0
-bash
```

Change the shell

{name of the shell}

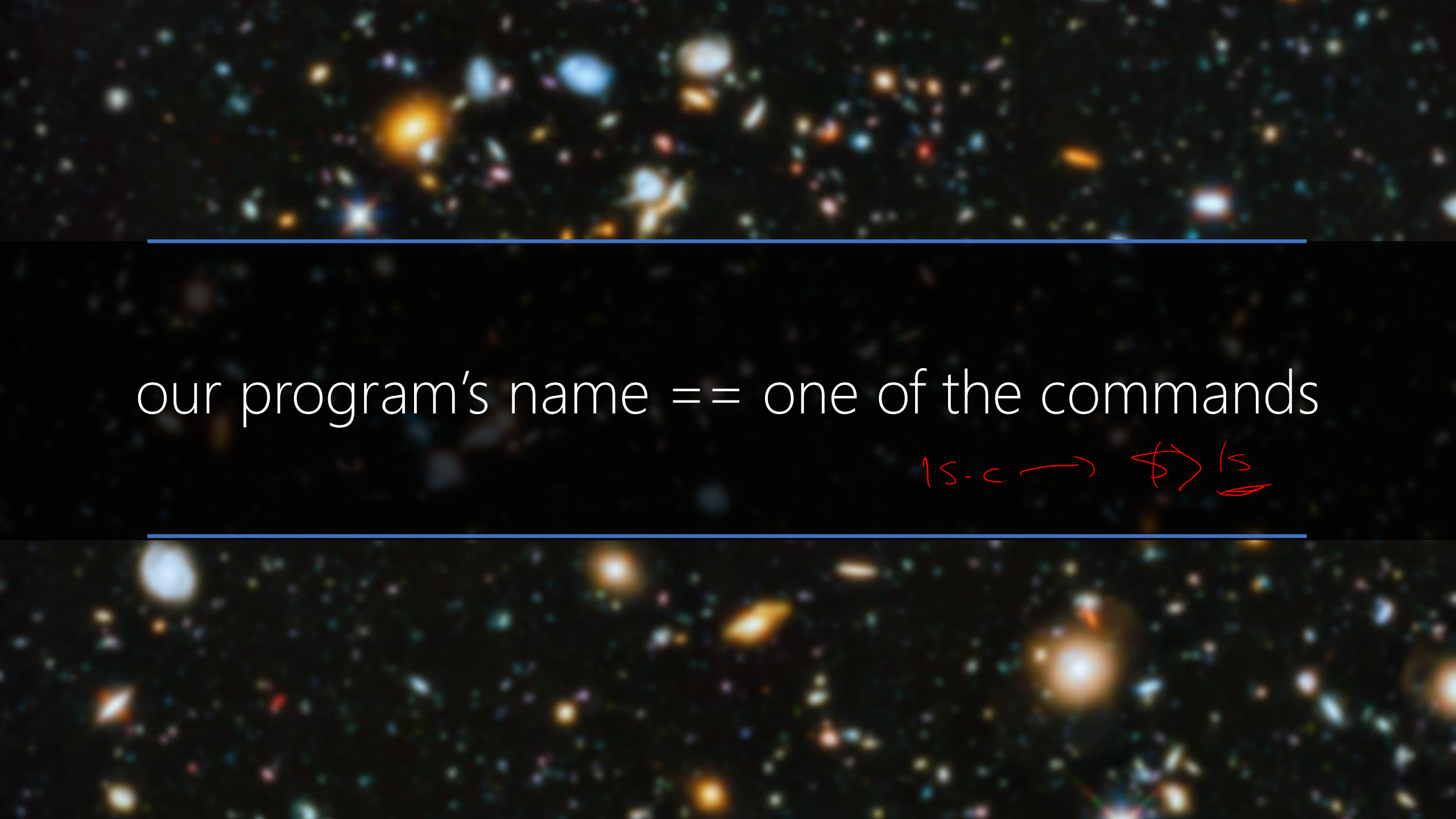


```
hfani@alpha:~$ echo $0
-bash
hfani@alpha:~$ sh
\[\e]0;\u@\h: \w\a\]\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]$ echo $0
sh
```

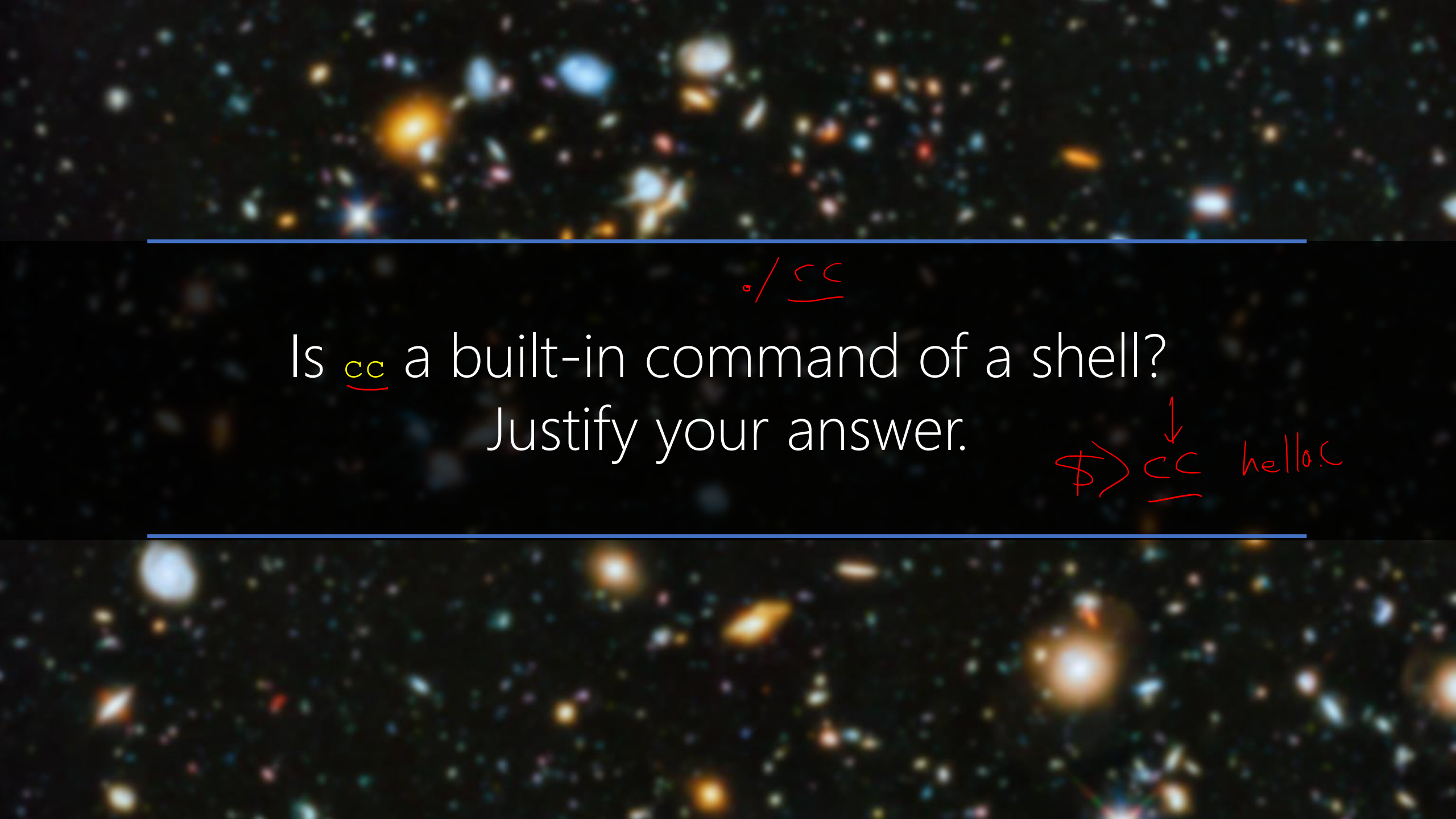
Exit the shell (back to previous one)

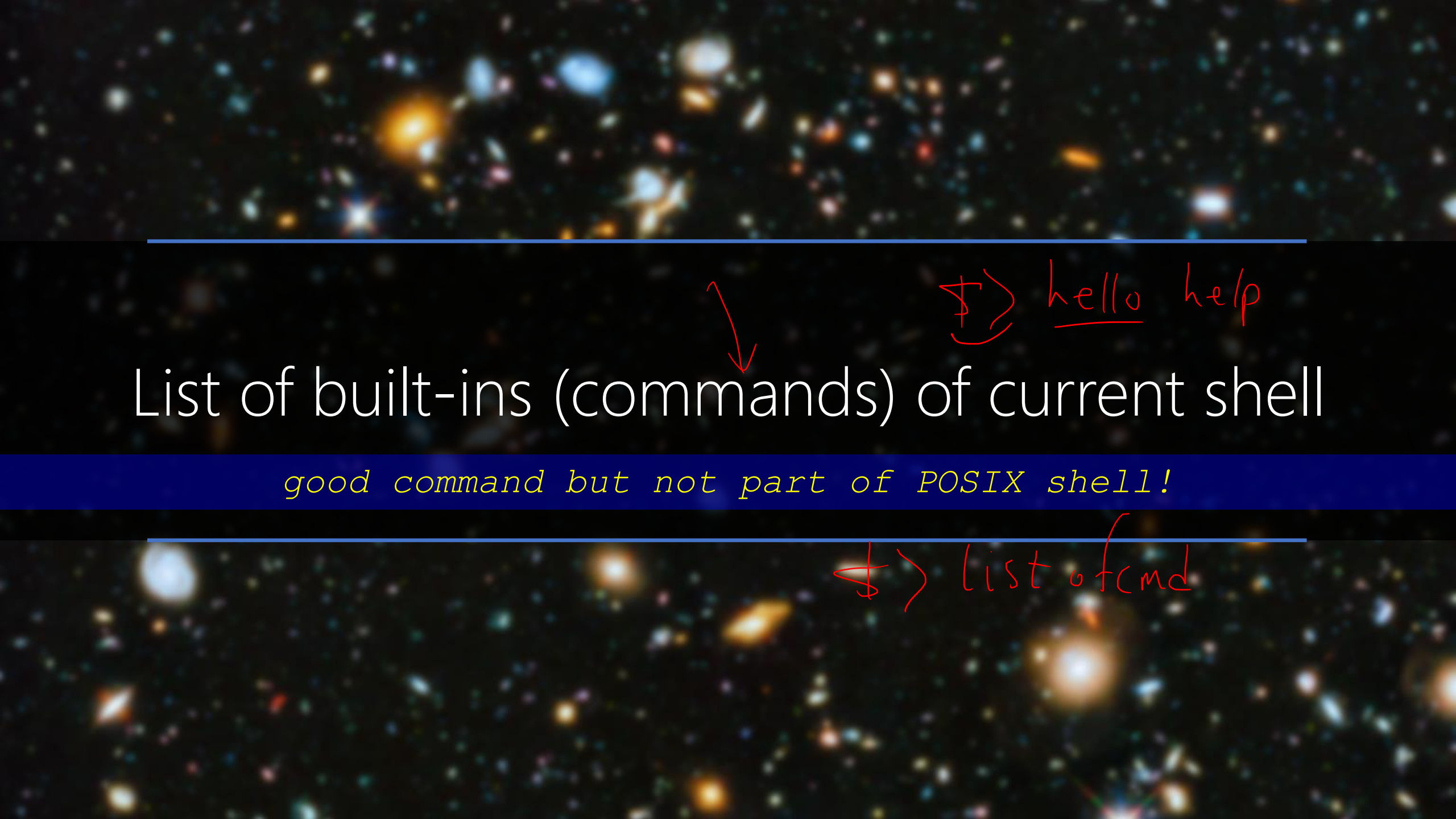
```
exit
```

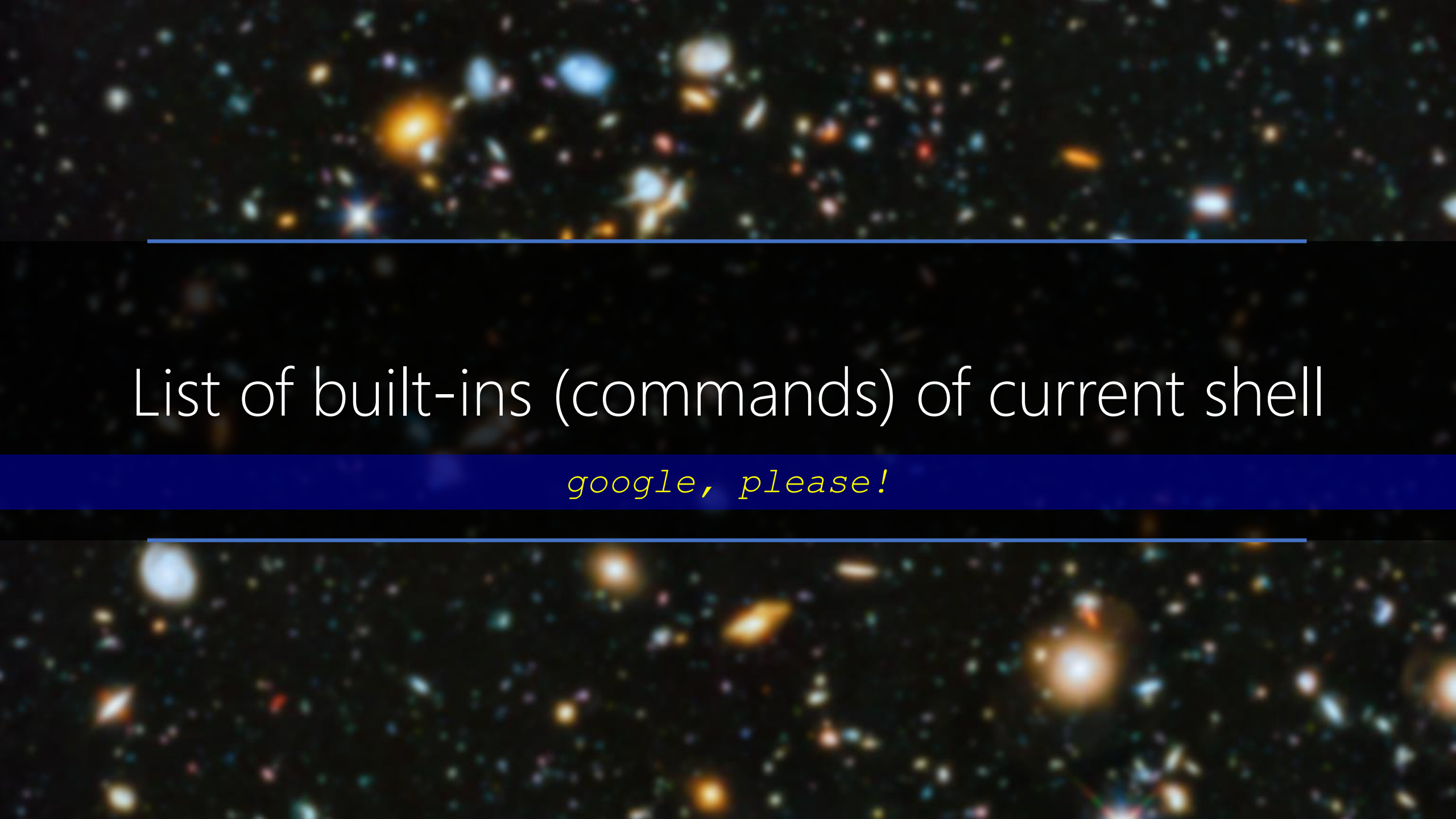
```
hfani@alpha:~$ echo $0
-bash
hfani@alpha:~$ sh
\[\e]0;\u@\h: \w\a\]\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]$ echo $0
sh
\[\e]0;\u@\h: \w\a\]\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]$ exit
hfani@alpha:~$ echo $0
-bash
hfani@alpha:~$
```


our program's name == one of the commands
ls.c → ls

our program's name == one of the commands

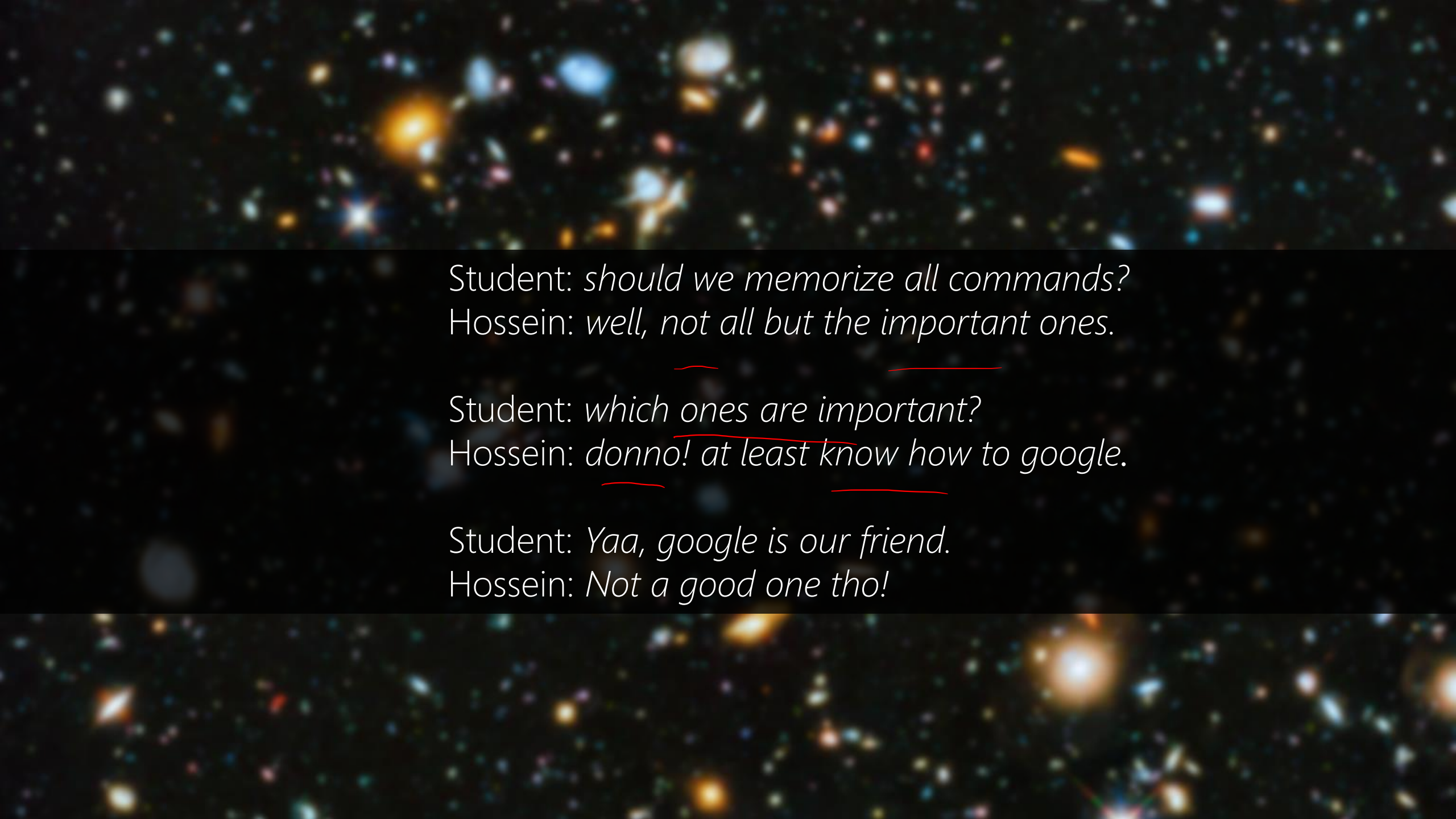

./cc
Is cc a built-in command of a shell?
Justify your answer.
$> cc hello.c


$ hello help
List of built-ins (commands) of current shell
good command but not part of POSIX shell!
$ list of cmd



List of built-ins (commands) of current shell

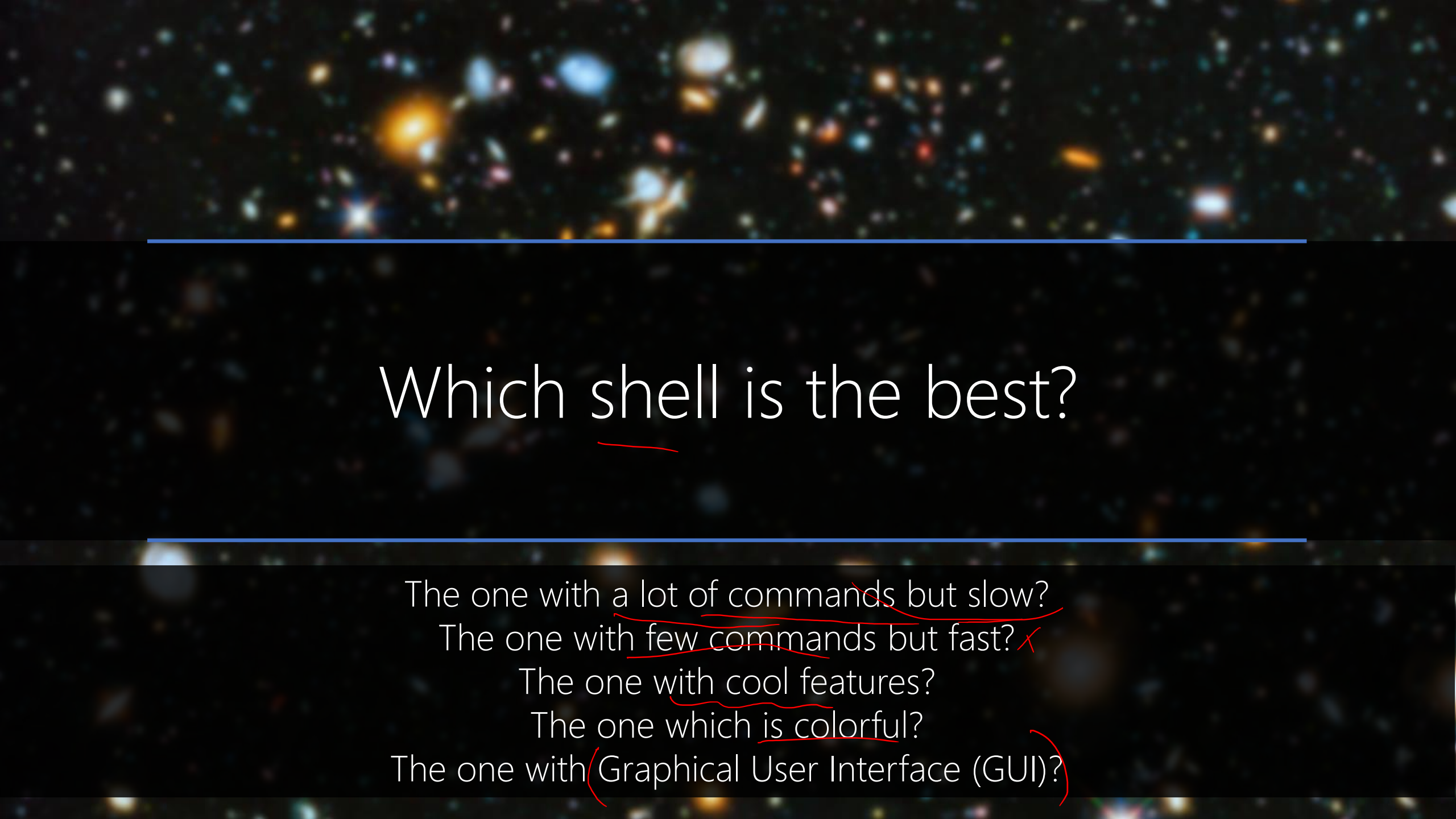
google, please!



Student: *should we memorize all commands?*
Hossein: *well, not all but the important ones.*

Student: *which ones are important?*
Hossein: *donno! at least know how to google.*

Student: *Yaa, google is our friend.*
Hossein: *Not a good one tho!*

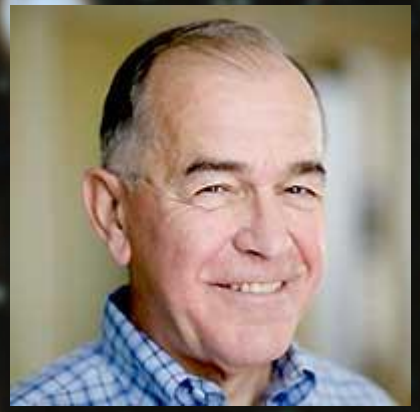


Which shell is the best?

The one with a lot of commands but slow?
The one with few commands but fast?
The one with cool features?
The one which is colorful?
The one with Graphical User Interface (GUI)?

Which shell is the best?

`sh`



Stephen Richard "Steve" Bourne

- Bell Labs
- The first shell
- minimal, chosen as standard (POSIX)

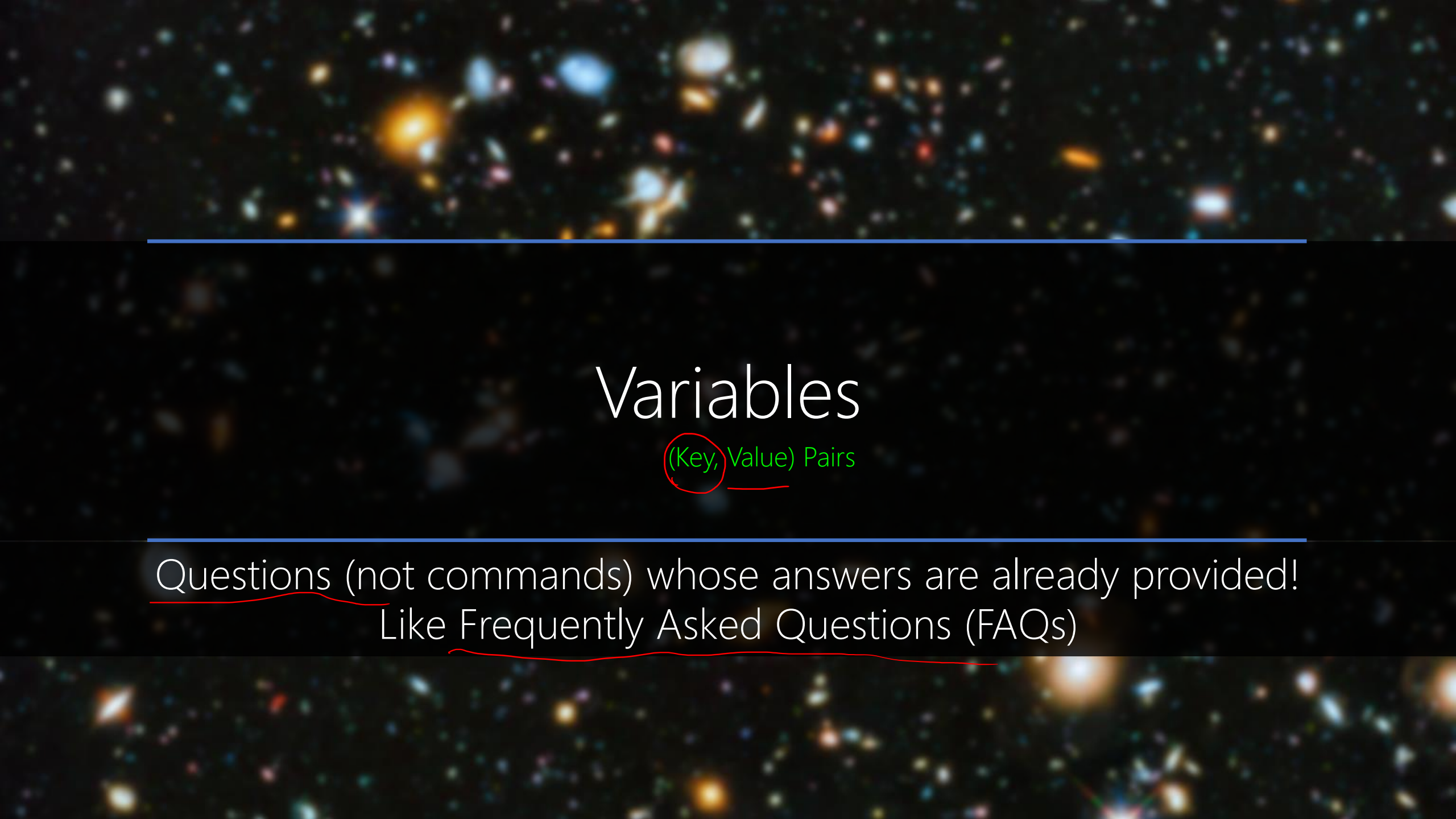

MATT DAMON
THE
BOURNE ULTIMATUM

Which shell is the best?

`bash`

Brian Jhan Fox

- Bourne again (born again) sh
- GNU project
- Default for most Linux distributions
- Available alternative in macOS
- https://www.gnu.org/software/bash/manual/bash.html
- https://ftp.gnu.org/gnu/bash/



Variables

(Key, Value) Pairs

Questions (not commands) whose answers are already provided!
Like Frequently Asked Questions (FAQs)

System Variables

aka. Environment Variables, Global Variables, Unix Variables

Hossein: Kernel Variables

By convention, keys are UPPERCASE

To see the value, `echo ${KEY}`

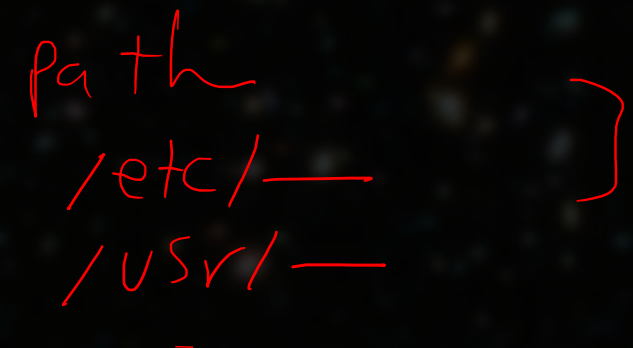
```
hfani@alpha:~$ echo $OSTYPE
linux-gnu
hfani@alpha:~$ echo $USER
hfani
hfani@alpha:~$ echo $LOGNAME
hfani
hfani@alpha:~$ echo $HOME
/home/hfani
hfani@alpha:~$ echo $HOST

hfani@alpha:~$ echo $DISPLAY

hfani@alpha:~$ echo $EDITOR

hfani@alpha:~$ echo $SHELL
/bin/bash
hfani@alpha:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/opt/maple2021/bin:/opt/netlogo:/
ome/hfani/.dotnet/tools
hfani@alpha:~$
```

Not Set! Unset.

An important one! Very important actually.

PATH

Colon(:)-delimited list of directories



Tells the shell where to look/search when you request a particular program

```
hfani@alpha:~$ cd /usr/bin
hfani@alpha:/usr/bin$ ./cc hello.c -o hello
cc: error: hello.c: No such file or directory
cc: fatal error: no input files
compilation terminated.
hfani@alpha:/usr/bin$
```

The actual location of program file for C compiler

Either you have to copy your files to /usr/bin
Or copy cc to your directory

Both are impossible due to lack of administrative privileges

```
hfani@alpha:~$ cd /usr/bin
hfani@alpha:/usr/bin$ ./cc hello.c -o hello
cc: error: hello.c: No such file or directory
cc: fatal error: no input files
compilation terminated.
hfani@alpha:/usr/bin$ cd ~
hfani@alpha:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/opt/maple2021/bin:/opt/netlogo:/opt/eclip
ome/hfani/.dotnet/tools
hfani@alpha:~$ cc hello.c -o hello
hfani@alpha:~$
```

./cc /home/hfani/hello.c

Back to home directory

hello.c

Because shell also searched these locations

/usr/bin/cc h

System Variables

aka. Environment Variables, Global Variables, Unix Variables

Hossein: Kernel Variables

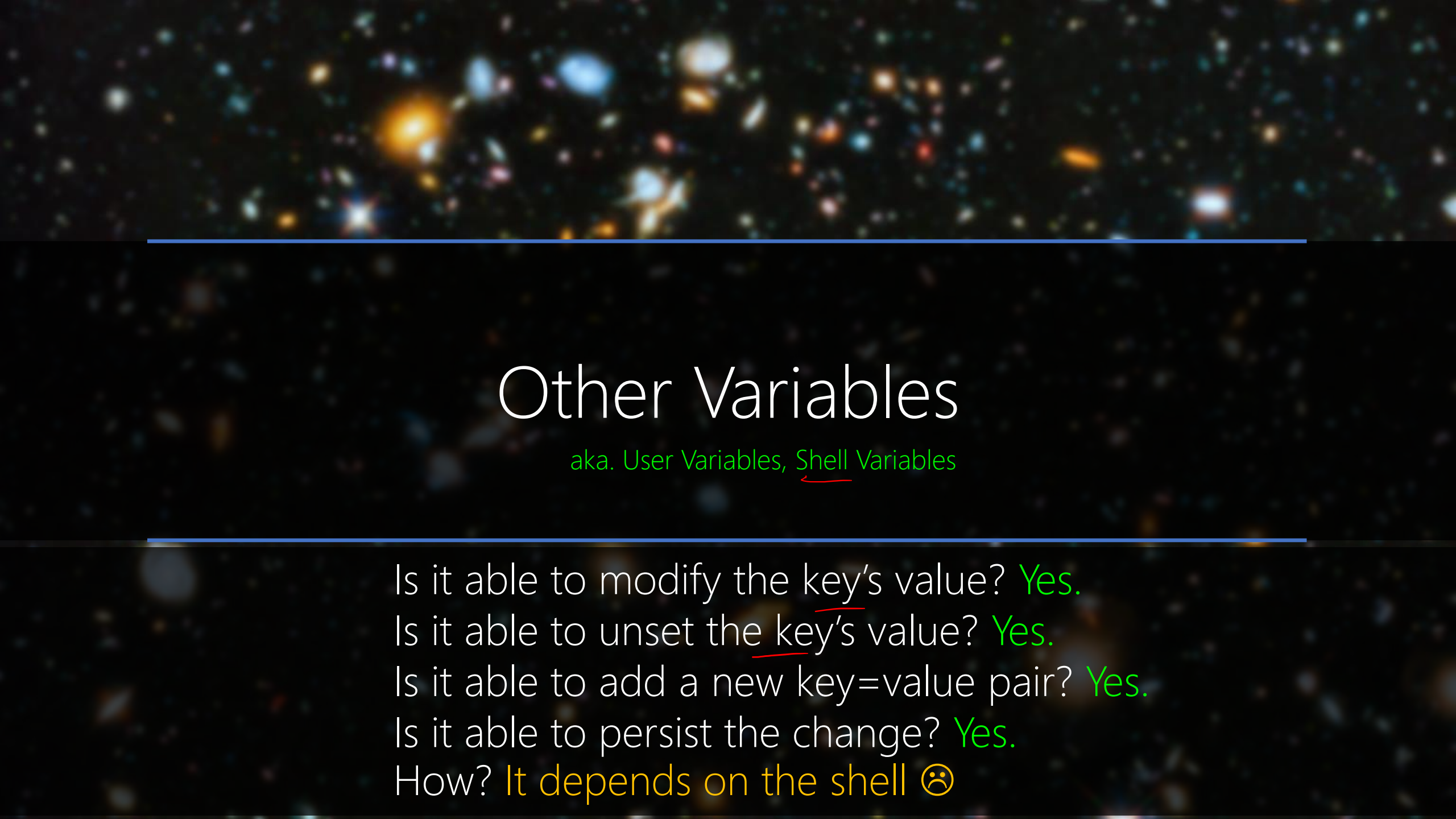
Is it able to modify the KEY's value? Yes.
Is it able to unset the KEY's value? Yes.
Is it able to add a new KEY=Value pair? Yes.
Is it able to persist the change? Yes.
How? It depends on the shell ☹

Other Variables

aka. User Variables, Local Variable, Shell Variables

By convention, keys are lowercase

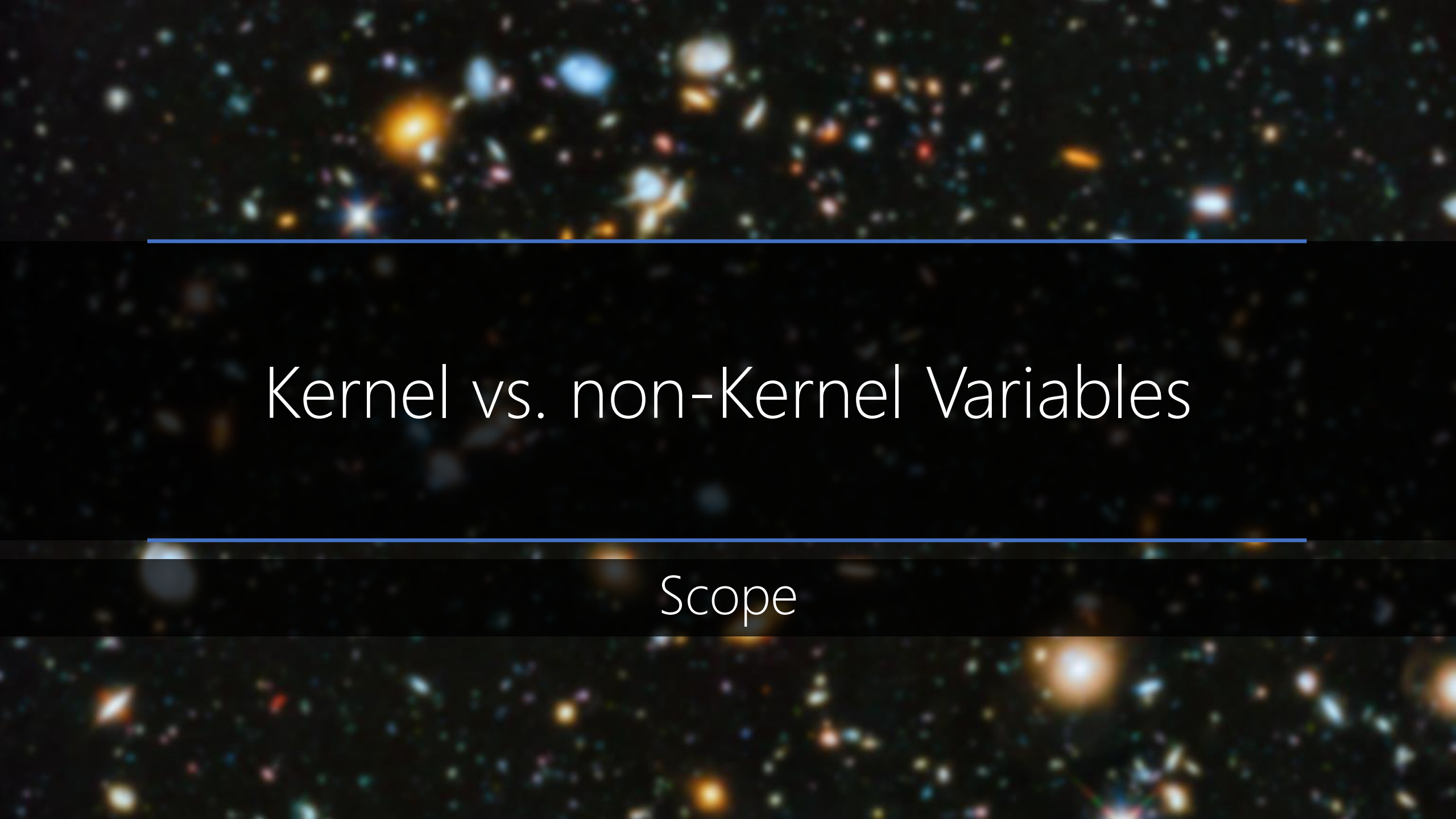
To see the value, `echo ${key}`



Other Variables

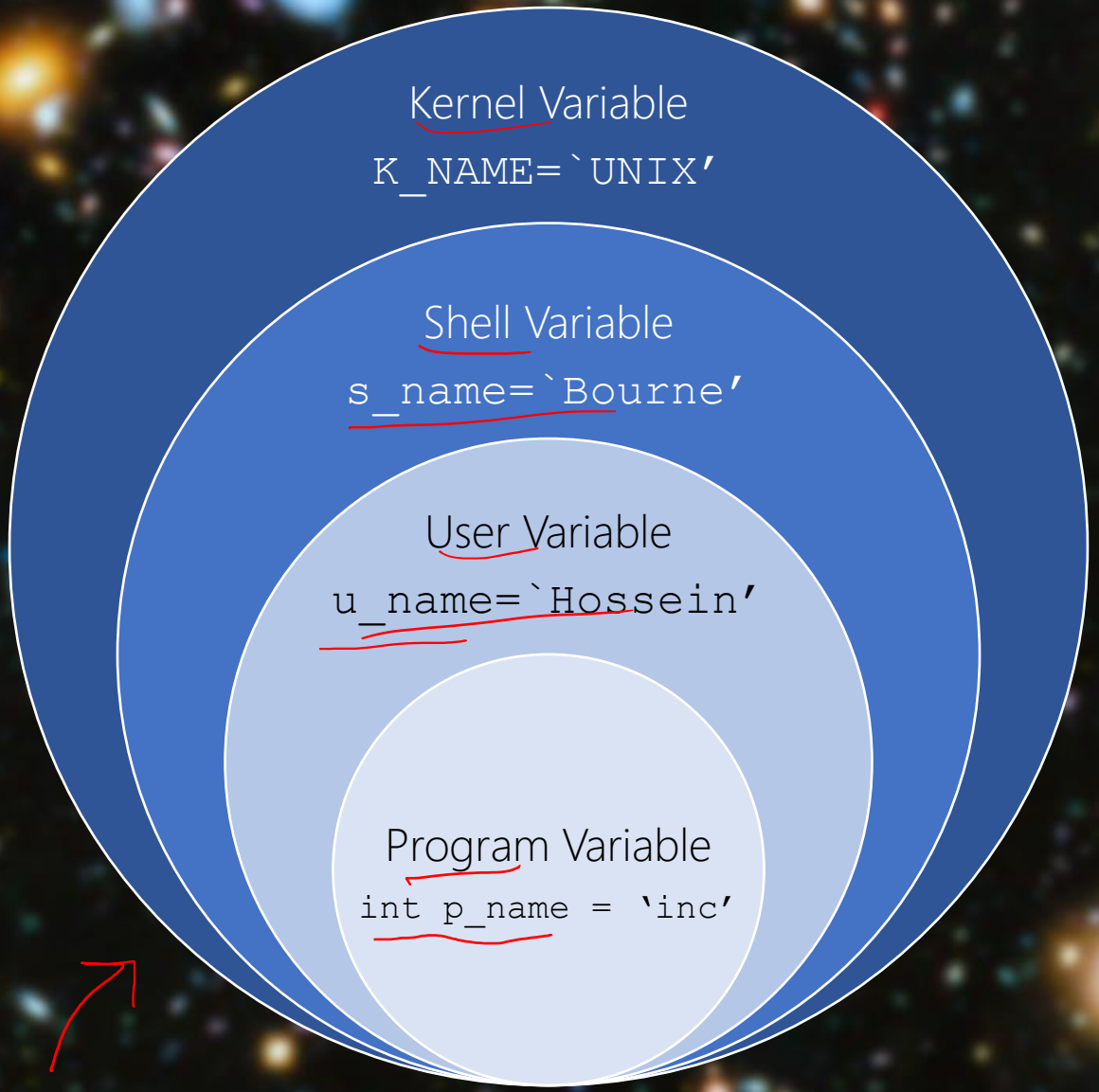
aka. User Variables, Shell Variables

Is it able to modify the key's value? Yes.
Is it able to unset the key's value? Yes.
Is it able to add a new key=value pair? Yes.
Is it able to persist the change? Yes.
How? It depends on the shell ☹



Kernel vs. non-Kernel Variables

Scope


Kernel Variable
K_NAME=`UNIX'
Shell Variable
s_name=`Bourne'
User Variable
u_name=`Hossein'
Program Variable
int p_name = 'inc'


Kernel Variable
K_NAME=`UNIX'
Shell Variable
s_name=`Bourne'
It's not that clear!
User Variable
u_name=`Hossein'
Program Variable
int p_name = 'inc'

Access Kernel Variables by Call to Library Routine

Lab02

echo

```
#include <stdlib.h>;
char *getenv(const char *KEY)
```

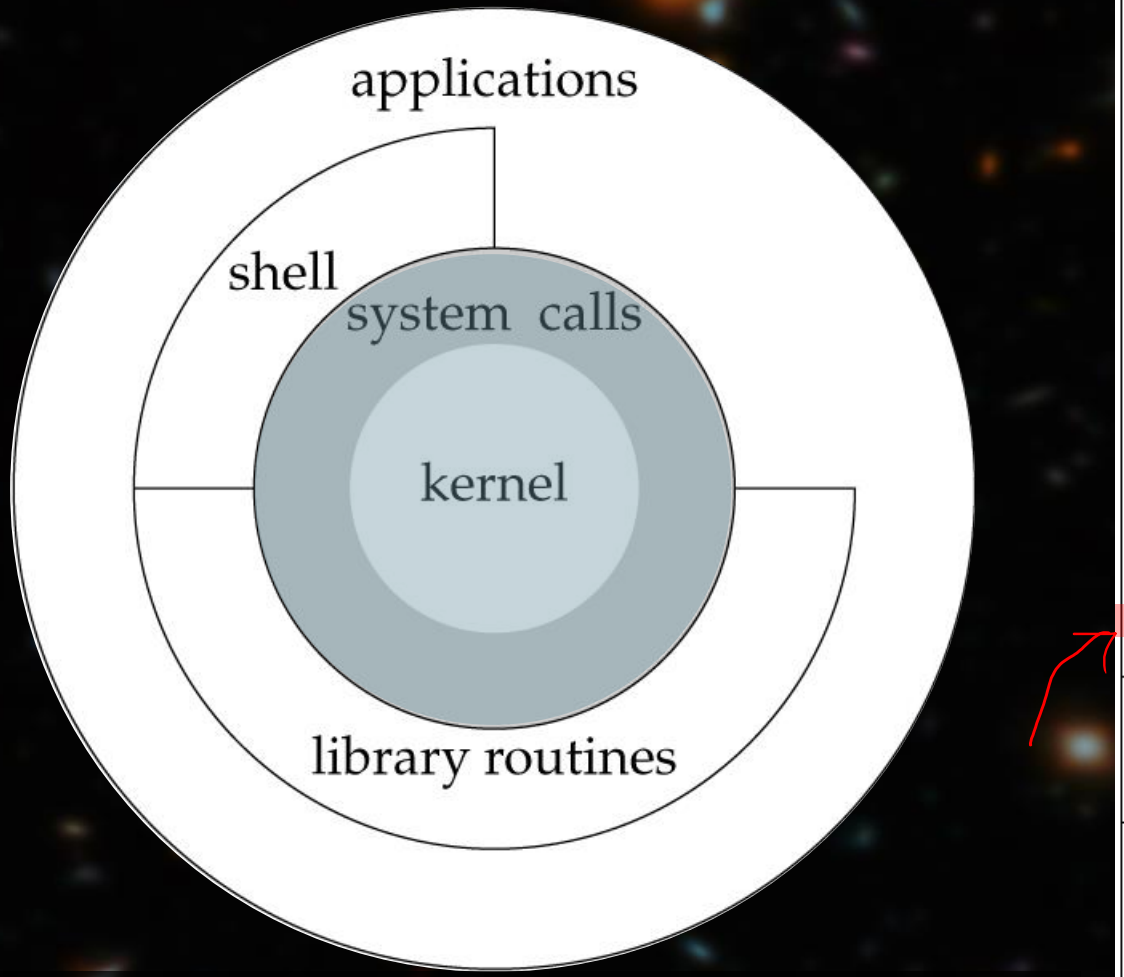
```c
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
int main()
{
  printf("Hello World! This is hfani@uwindsor.ca, StudentID: 123456789\n");
  time_t t = time(NULL);
  struct tm tm = *localtime(&t);
  printf("now: %d-%02d-%02d %02d:%02d:%02d\n", tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_m
  printf("%s@shell:%s$\n", getenv("USER"),getenv("PWD"));

}
```

The library routine that does system call to Kernel
Your program statically linked this library!

Getting the values of USER and PWD (path of working directory)

Access Kernel Variables by System Call

```
#include <unistd.h>;
extern char **environ
```



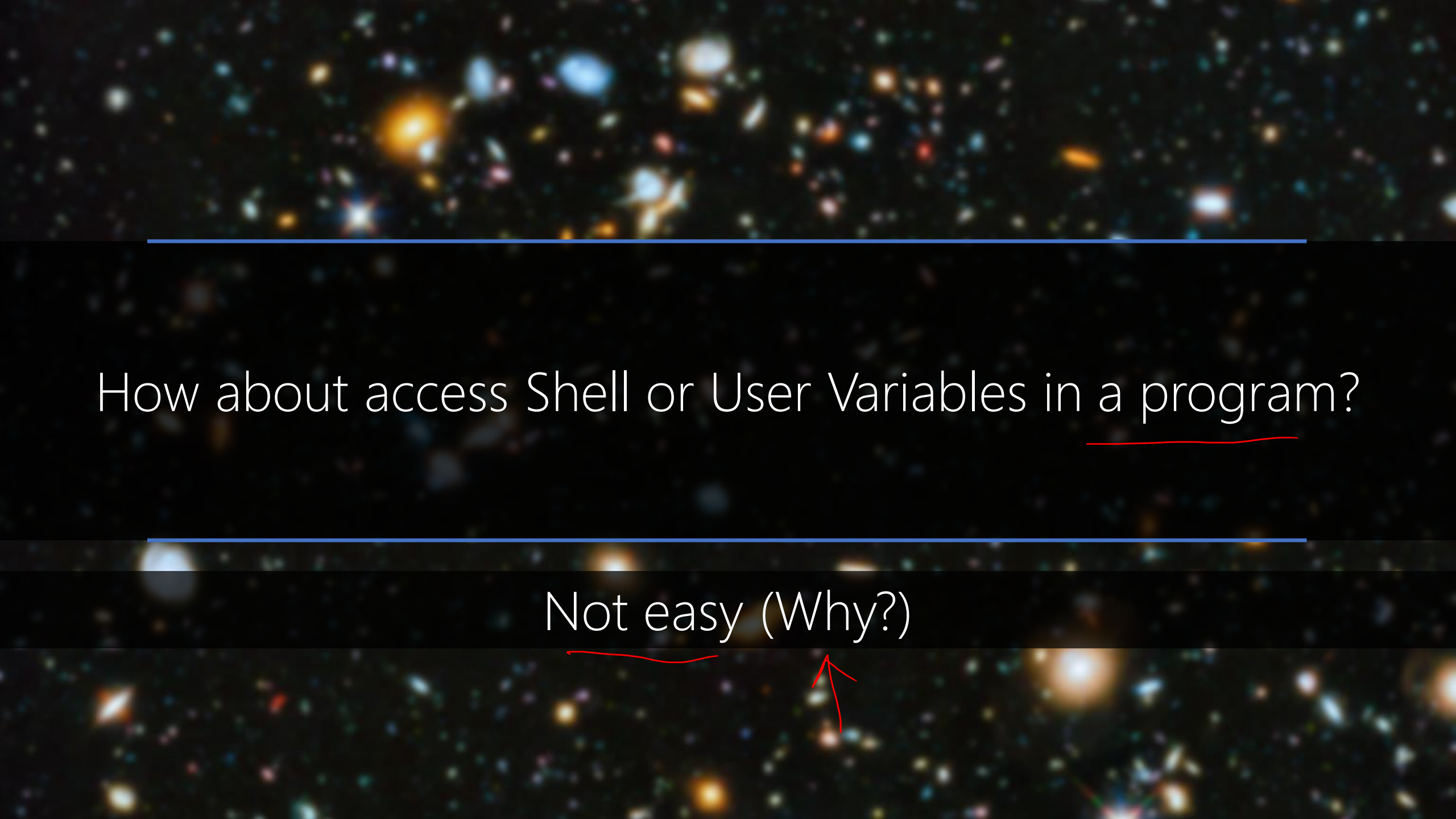
Header	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10	Description
<aio.h>	•	•	•	•	asynchronous I/O
<cpio.h>	•	•	•	•	cpio archive values
<dirent.h>	•	•	•	•	directory entries (Section 4.22)
<dlfcn.h>	•	•	•	•	dynamic linking
<fcntl.h>	•	•	•	•	file control (Section 3.14)
<fnmatch.h>	•	•	•	•	filename-matching types
<glob.h>	•	•	•	•	pathname pattern-matching and generation
<grp.h>	•	•	•	•	group file (Section 6.4)
<iconv.h>	•	•	•	•	codeset conversion utility
<langinfo.h>	•	•	•	•	language information constants
<monetary.h>	•	•	•	•	monetary types and functions
<netdb.h>	•	•	•	•	network database operations
<nl_types.h>	•	•	•	•	message catalogs
<poll.h>	•	•	•	•	poll function (Section 14.4.2)
<pthread.h>	•	•	•	•	threads (Chapters 11 and 12)
<pwd.h>	•	•	•	•	password file (Section 6.2)
<regex.h>	•	•	•	•	regular expressions
<sched.h>	•	•	•	•	execution scheduling
<semaphore.h>	•	•	•	•	semaphores
<strings.h>	•	•	•	•	string operations
<tar.h>	•	•	•	•	tar archive values
<termios.h>	•	•	•	•	terminal I/O (Chapter 18)
<unistd.h>	•	•	•	•	symbolic constants
<wordexp.h>	•	•	•	•	word-expansion definitions
<arpa/inet.h>	•	•	•	•	Internet definitions (Chapter 16)
<net/if.h>	•	•	•	•	socket local interfaces (Chapter 16)
<netinet/in.h>	•	•	•	•	Internet address family (Section 16.3)
<netinet/tcp.h>	•	•	•	•	Transmission Control Protocol definitions
<sys/mman.h>	•	•	•	•	memory management declarations
<sys/select.h>	•	•	•	•	select function (Section 14.4.1)
<sys/socket.h>	•	•	•	•	sockets interface (Chapter 16)
<sys/stat.h>	•	•	•	•	file status (Chapter 4)
<sys/statvfs.h>	•	•	•	•	file system information
<sys/times.h>	•	•	•	•	process times (Section 8.17)
<sys/types.h>	•	•	•	•	primitive system data types (Section 2.8)
<sys/un.h>	•	•	•	•	UNIX domain socket definitions (Section 17.2)
<sys/utsname.h>	•	•	•	•	system name (Section 6.9)
<sys/wait.h>	•	•	•	•	process control (Section 8.6)

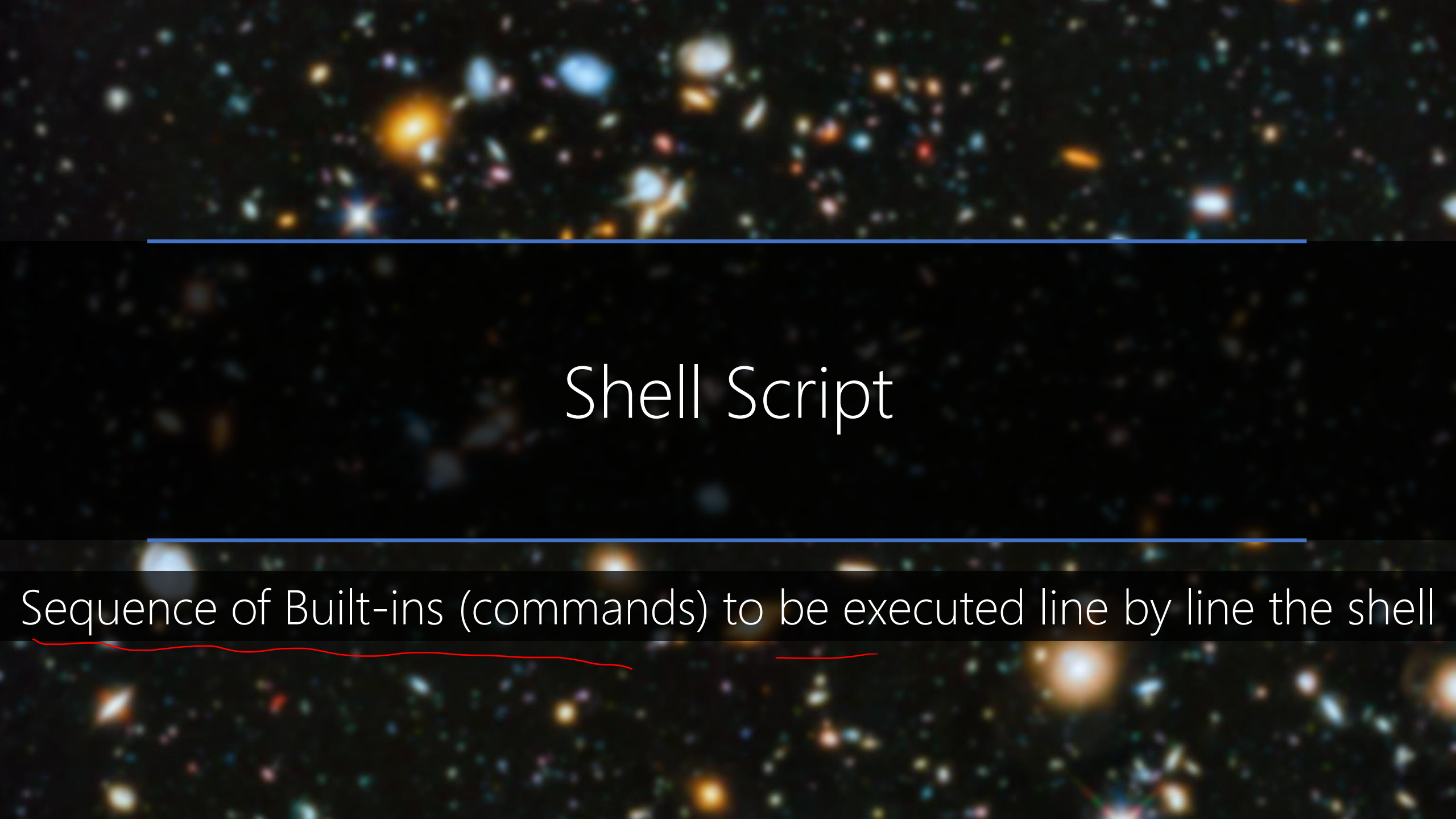
```c
#include <stdio.h>
#include <unistd.h>

extern char **environ;
int main(int argc, char *argv[])
{
    int index = 0;
    printf("Environment variables:\n");
    index = 0;
    while (environ[index])
    {
        printf("envp[%d]: %s\n", index, environ[index]);
        ++index;
    }
    return 0;
}
```

System call to Kernel!
Either statically or dynamically linked.

```
Environment variables:
envp[0]: SHELL=/bin/bash
envp[1]: LANGUAGE=en_CA:en
envp[2]: NO_AT_BRIDGE=1
envp[3]: TWO_TASK=cs01
envp[4]: PWD=/home/hfani
envp[5]: LOGNAME=hfani
envp[6]: XDG_SESSION_TYPE=tty
envp[7]: PRINTER=cs_commons
envp[8]: MOTD_SHOWN=pam
envp[9]: VIRTUALENVWRAPPER_SCRIPT=/usr/share/virtualenvwrapper/virtualenvwrapper.sh
envp[10]: HOME=/home/hfani
envp[11]: LANG=en_CA.UTF-8
envp[12]: LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:
:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;
=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.l
*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01
r=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:
;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35
1;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:
;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:
;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.f
:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00
=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36
0;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
envp[13]: _VIRTUALENVWRAPPER_API= mkvirtualenv rmvirtualenv lsvirtualenv showvirtualenv workon add2virtualenv cdsitepackages cd
nv lssitepackages toggleglobalsitepackages cpvirtualenv setvirtualenvproject mkproject cdproject mktmpenv wipeenv allvirtualenv
alenv rmvirtualenv lsvirtualenv showvirtualenv workon add2virtualenv cdsitepackages cdvirtualenv lssitepackages toggleglobalsit
s cpvirtualenv setvirtualenvproject mkproject cdproject mktmpenv wipeenv allvirtualenv
envp[14]: ORACLE_HOME=/usr/lib/oracle/12.1/client64
envp[15]: SSH_CONNECTION=137.207.140.134 63217 137.207.82.51 22
envp[16]: WINEDLLOVERRIDES=winemenubuilder.exe=d
envp[17]: LESSCLOSE=/usr/bin/lesspipe %s %s
envp[18]: XDG_SESSION_CLASS=user
envp[19]: TERM=xterm
envp[20]: LESSOPEN=| /usr/bin/lesspipe %s
envp[21]: USER=hfani
```

echo $0


How about access Shell or User Variables in a program?
Not easy (Why?)

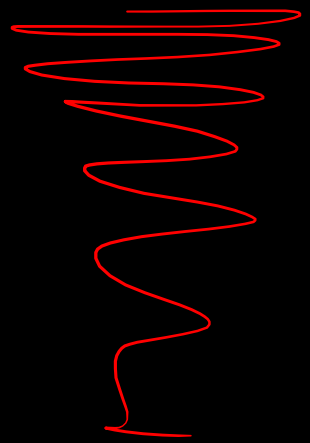

Shell Script
Sequence of Built-ins (commands) to be executed line by line the shell

```
hfani@alpha:~$ vi myscript.sh
#!/bin/bash
echo "hello world!"
echo "my name is $USER"
echo "here is the content of my home directory:"
ls ~/
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

By convention, the extension is .sh

Important: under what shell?

#!/bin/sh



```
hfani@alpha:~$ chmod +x myscript.sh
hfani@alpha:~$ ./myscript.sh
hello world!
my name is hfani
here is the content of my home directory:
Desktop    Downloads           hello    hellocat.txt  increment.o  ls.c    main.o
Documents  eclipse-workspace   hello.c  increment.c   increment.s  main.c  main.s
hfani@alpha:~$
```

Important: make it executable


Shell Script
Lab03

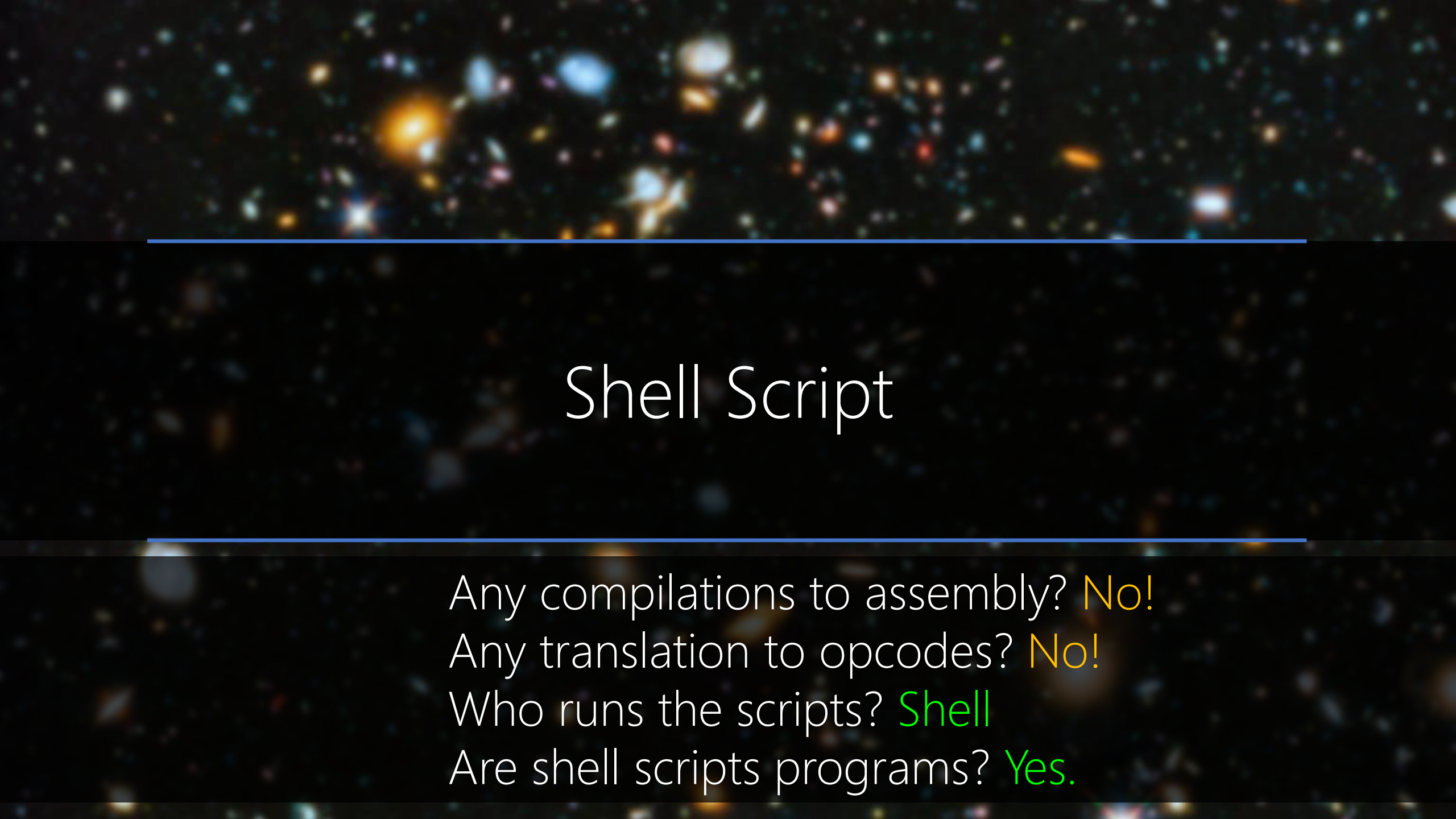
```
hfani@alpha:~$ vi build_lab03.sh
```

```
#!/bin/bash
echo "start building lab03 program:"
echo "compiling to assembly lines ..."
cc main.c -S
cc increment.c -S
echo "translating to opcodes ..."
cc main.s -c
cc increment.s -c
echo "statically linking all required opcodes ..."
cc main.o increment.o -o main
echo "build successfully done!"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

```
hfani@alpha:~$ chmod +x build_lab03.sh
hfani@alpha:~$ ./build_lab03.sh
start building lab03 program:
compiling to assembly lines ...
main.c: In function ‘main’:
main.c:5:6: warning: implicit declaration of function ‘increment’ [-Wimplicit-function-de
    5 |  a = increment(a);
      |      ^~~~~~~~~
translating to opcodes ...
statically linking all required opcodes ...
build successfully done!
hfani@alpha:~$
```

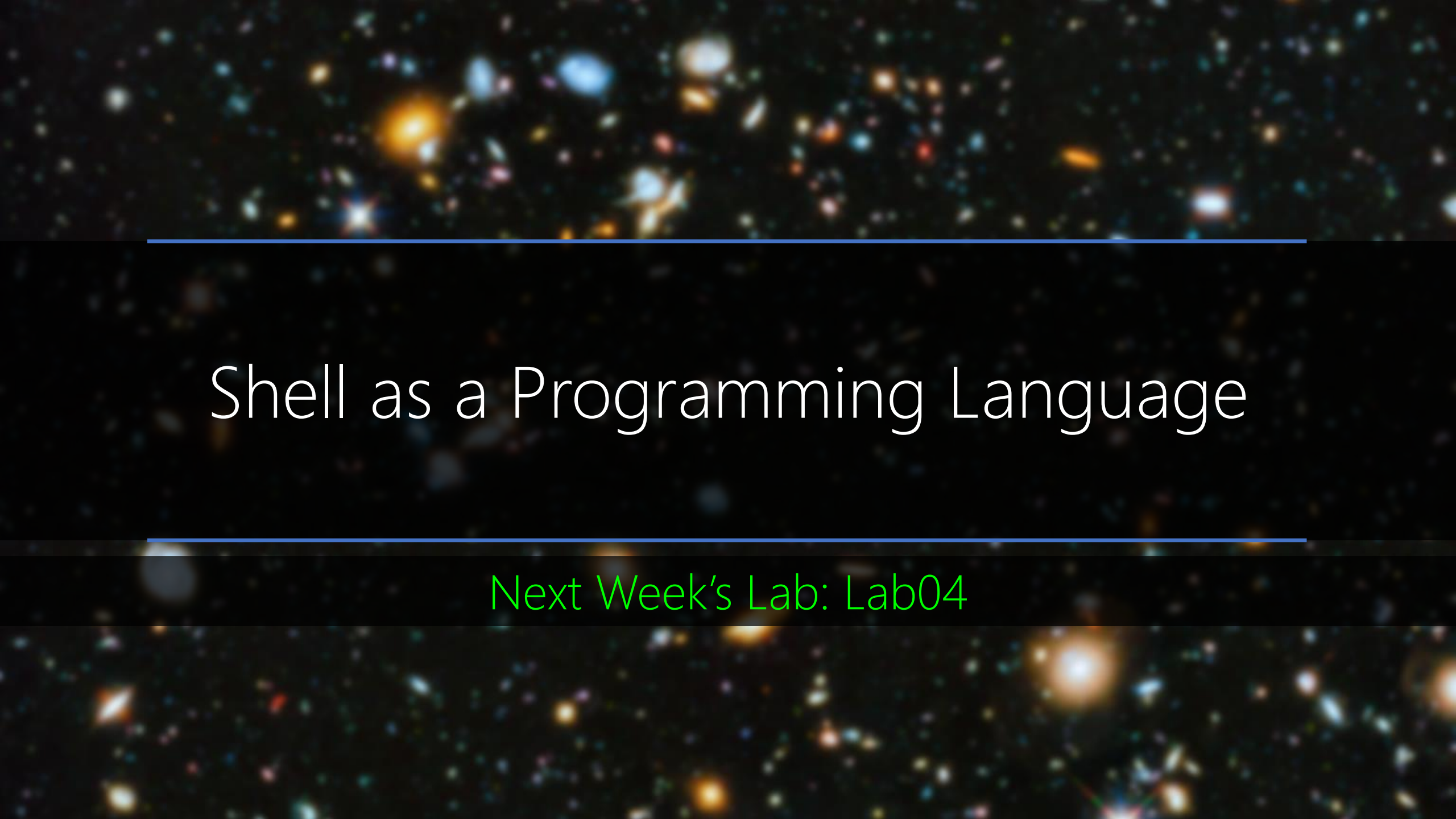
Shell Script

Any compilations to assembly?
Any translation to opcodes?
Who runs the scripts?
Are shell scripts programs?



Shell Script

Any compilations to assembly? No!
Any translation to opcodes? No!
Who runs the scripts? Shell
Are shell scripts programs? Yes.



Shell as a Programming Language

Next Week's Lab: Lab04