

The Unix Shell

Job Control



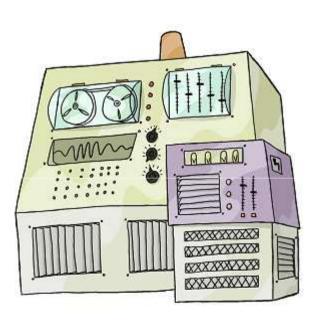
Copyright © Software Carpentry 2010

This work is licensed under the Creative Commons Attribution License

See http://software-carpentry.org/license.html for more information.

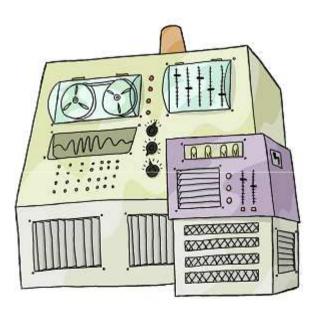










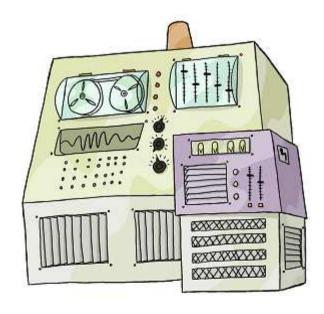


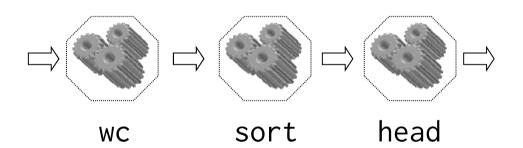








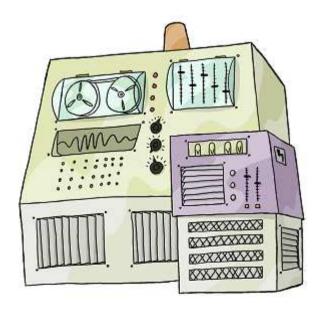


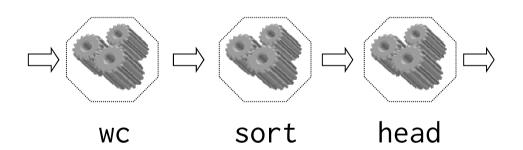










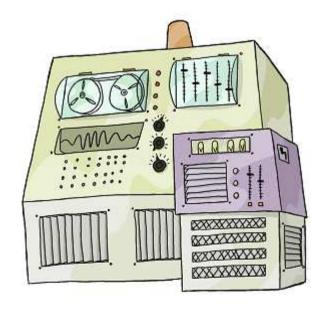


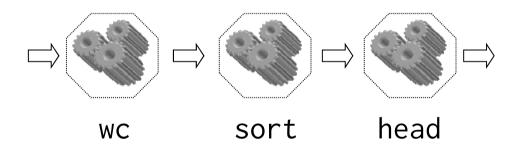
Control programs while they run











processes
Control programs while they run





Some are yours



Some are yours

Most belong to the operating system (or other users)



Some are yours

Most belong to the operating system (or other users)

Use ps to get a list



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
PID
      PPID
             PGID
                   TTY
                         UTD
                                STIME
                                        COMMAND
2152
             2152 con
                       1000 13:19:07
                                        /usr/bin/bash
    2152
           2276 con
                        1000 14:53:48
                                        /usr/bin/ps
2276
$
```



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
PID
      PPID
             PGID
                  TTY
                         UTD
                                STIME
                                        COMMAND
2152
             2152 con 1000 13:19:07
                                        /usr/bin/bash
2276
      2152
           2276 con 1000 14:53:48
                                        /usr/bin/ps
$
```

Process ID (unique at any moment)



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
PID
      PPID
             PGID
                    TTY
                          UTD
                                  STIME
                                          COMMAND
                   con 1000 13:19:07
2152
             2152
                                          /usr/bin/bash
      2152
2276
             2276
                    con 1000 14:53:48
                                          /usr/bin/ps
$
```

Parent process ID



Some are yours

Most belong to the operating system (or other users)

Use ps to get a list

```
$ ps
      PPID
PID
                                  STIME
             PGID
                    TTY
                         UTD
                                         COMMAND
2152
             2152 con 1000 13:19:07
                                         /usr/bin/bash
      2152
2276
             2276
                    con 1000 14:53:48
                                         /usr/bin/ps
$
```

Parent process ID

What process created this one?



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
                    TTY
      PPID
              PGID
PID
                          UTD
                                  STIME
                                          COMMAND
             2152
2152
                         1000 13:19:07
                                         /usr/bin/bash
                    con
    2152
             2276
                         1000 14:53:48
2276
                    con
                                          /usr/bin/ps
$
```

Process group ID



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
             PGID
                     TTY
                          UID
PID
      PPID
                                  STIME
                                          COMMAND
             2152
                         1000 13:19:07
2152
                    con
                                          /usr/bin/bash
                         1000 14:53:48
     2152
                                          /usr/bin/ps
2276
            2276
                    con
$
```

What terminal (TTY) is it running in?



Some are yours

Most belong to the operating system (or other users)

Use ps to get a list

```
$ ps
              PGID
                     TTY
                           UID
PID
       PPID
                                   STIME
                                            COMMAND
              2152
2152
                     con
                          1000 13:19:07
                                           /usr/bin/bash
      2152
                                           /usr/bin/ps
2276
              2276
                          1000 14:53:48
                     con
$
```

What terminal (TTY) is it running in? '?' indicates a system service (no TTY)



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
                    TTY
PID
      PPID
             PGID
                                 STIME
                                         COMMAND
                        [1000 13:19:07 /usr/bin/bash
             2152
                  con
2152
    2152
           2276
                         1000 14:53:48
                                        /usr/bin/ps
2276
                    con
$
```

The user ID of the process's owner



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
                    TTY
PID
      PPID
             PGID
                                 STIME
                                         COMMAND
                        [1000 13:19:07 /usr/bin/bash]
                  con
2152
            2152
2276 2152
           2276
                        1000 14:53:48
                                        /usr/bin/ps
                    con
$
```

The user ID of the process's owner Controls what the process can read, write, execute, ...



Some are yours

Most belong to the operating system (or other users)

Use ps to get a list

```
$ ps
                          UID
                                  STIME
PID
      PPID
             PGID
                    TTY
                                         COMMAND
                         1000
                               13:19:07
                                        /usr/bin/bash
2152
             2152 con
    2152
           2276 con
                                         /usr/bin/ps
2276
                         1000
                               14:53:48
$
```

When the process was started



Some are yours

Most belong to the operating system (or other users)
Use ps to get a list

```
$ ps
                                         COMMAND
                                 STIME
PID
      PPID
             PGID
                    TTY
                         UID
             2152
                                         /usr/bin/bash
2152
                  con 1000 13:19:07
           2276 con 1000 14:53:48
2276 2152
                                        /usr/bin/ps
$
```

The program the process is executing





\$./analyze results*.dat



\$./analyze results*.dat

...a few minutes pass...



```
$ ./analyze results*.dat
...a few minutes pass...
^C
$
```





```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$
```



```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$
```

Run in the background



```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$
```

Run in the *background*Shell returns right away instead
of waiting for the program to finish



```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$ fbcmd events
$
```

Can run other programs in the *foreground* while waiting for background process(es) to finish



```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$ fbcmd events
$ jobs
[1] ./analyze results01.dat results02.dat results03.dat
$
```





```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$ fbcmd events
$ jobs
[1] ./analyze results01.dat results02.dat results03.dat
$ fg
```





```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$ fbcmd events
$ jobs
[1] ./analyze results01.dat results02.dat results03.dat
                            Bring background job to foreground
                            Use fg %1, fg %2, etc. if there are
```

Job Control Introduction

several background jobs



```
$ ./analyze results*.dat
...a few minutes pass...
^C
$ ./analyze results*.dat &
$ fbcmd events
$ jobs
[1] ./analyze results01.dat results02.dat results03.dat
$ fg
...a few minutes pass...
$
                       And finally it's done
```



Use ^z to pause a program that's already running



Use ^Z to pause a program that's already running fg to resume it in the foreground



Use ^Z to pause a program that's already running fg to resume it in the foreground

Or bg to resume it as a background job



\$./analyze results01.dat



```
$ ./analyze results01.dat
^Z
[1] Stopped ./analyze results01.dat
$
```



```
$ ./analyze results01.dat
^Z
[1] Stopped ./analyze results01.dat
$ bg %1
$
```



```
$ ./analyze results01.dat
^Z
[1] Stopped ./analyze results01.dat
$ bg %1
$ jobs
[1] ./analyze results01.dat
$
```



```
$ ./analyze results01.dat
^Z
[1] Stopped ./analyze results01.dat
$ bg %1
$ jobs
[1] ./analyze results01.dat
$ kill %1
$
```



Job control mattered a lot when users only had one terminal window



Job control mattered a lot when users only had one terminal window

Less important now: just open another window



Job control mattered a lot when users only had one terminal window

Less important now: just open another window

Still useful when running programs remotely



created by

Greg Wilson

August 2010



Copyright © Software Carpentry 2010
This work is licensed under the Creative Commons Attribution License
See http://software-carpentry.org/license.html for more information.