SOFTENGER
Strive to Excel

Process in OS

Unix

Process

SOFTENGER
Strive to Excel

- A process is a name given to a program instance that has been loaded into memory and managed by the operating system
- A process is simply a program in execution: an instance of a program execution.
- Every process has a unique process identification number (PID).
- The kernel uses PIDs to track and manage their associated processes.
- The root user and regular users use PIDs to identify and control processes.
- There are several different ways you can obtain a list of the processes currently running on the system.
  - using the prstat / top command
  - using the ps command
  - using the Common Desktop Environment (CDE) Process Manager

The ps command

SOFTENGER
Strive to Excel

- The ps command enables you to check the status of the system's current processes and to obtain detailed information about them.
- The most commonly used command options used in combination with ps include
  - -a enables you to obtain a list of the most frequently requested processes.
  - -c option enables you to obtain a process formatted on the basis of the scheduler.
  - -G option enables you to obtain information about processes run by a specific user group.
  - -A option enables you to obtain a list of all current processes.
  - -f option enables you to obtain a comprehensive set of data about current processes that allows you, for example, to diagnose system bottlenecks and to determine which commands and applications are most commonly requested
  - -l option provides you with all information collected by the system about every current process.

The ps command examples

SOFTENGER
Strive to Excel

- To see every process on the system using standard syntax:

  ps -e

  ps -ef

  ps -eF

  ps –ely

- To get info about threads:

  ps -eLf

  ps axms

- To get security info:

  ps -eo euser, ruser, suser, fuser,f , comm,label

- To see every process running as root (real & effective ID) in user format:

  ps -U root -u root u

The ps command

- The Process Manager displays several columns containing specific types of information relating to each current process. These types of information are
  - PID
  - process name
  - process owner
  - CPU percentage used
  - total memory used
  - total swap space used
  - time of process start
  - parent process
  - process command

Signals

SOFTENGER
Strive to Excel

- You can control processes by using the kill command. This command sends a specific signal to the process. The signal results in one of the following:
  - a type of process death
  - the creation of a file called a “core dump” that contains a memory trace
  - OS incorporates a range of signal types. Each of these types is identified by a specific code.
  - For example, the SIGKILL signal, which is identified by the code 9, causes the relevant process to die immediately.
- This is the syntax of the kill command:

  # kill *signal-code PID*
- So, for example, to immediately kill the process assigned a PID of 300, you use this command:

  # kill -9 300
- Caution: **Kill** command is very sensitive command. You always need written approval from owner of the process to get it terminated using **kill** command.

Signals

SOFTENGER
Strive to Excel

- *Signals with specific codes are listed according to their code, name, and event:*

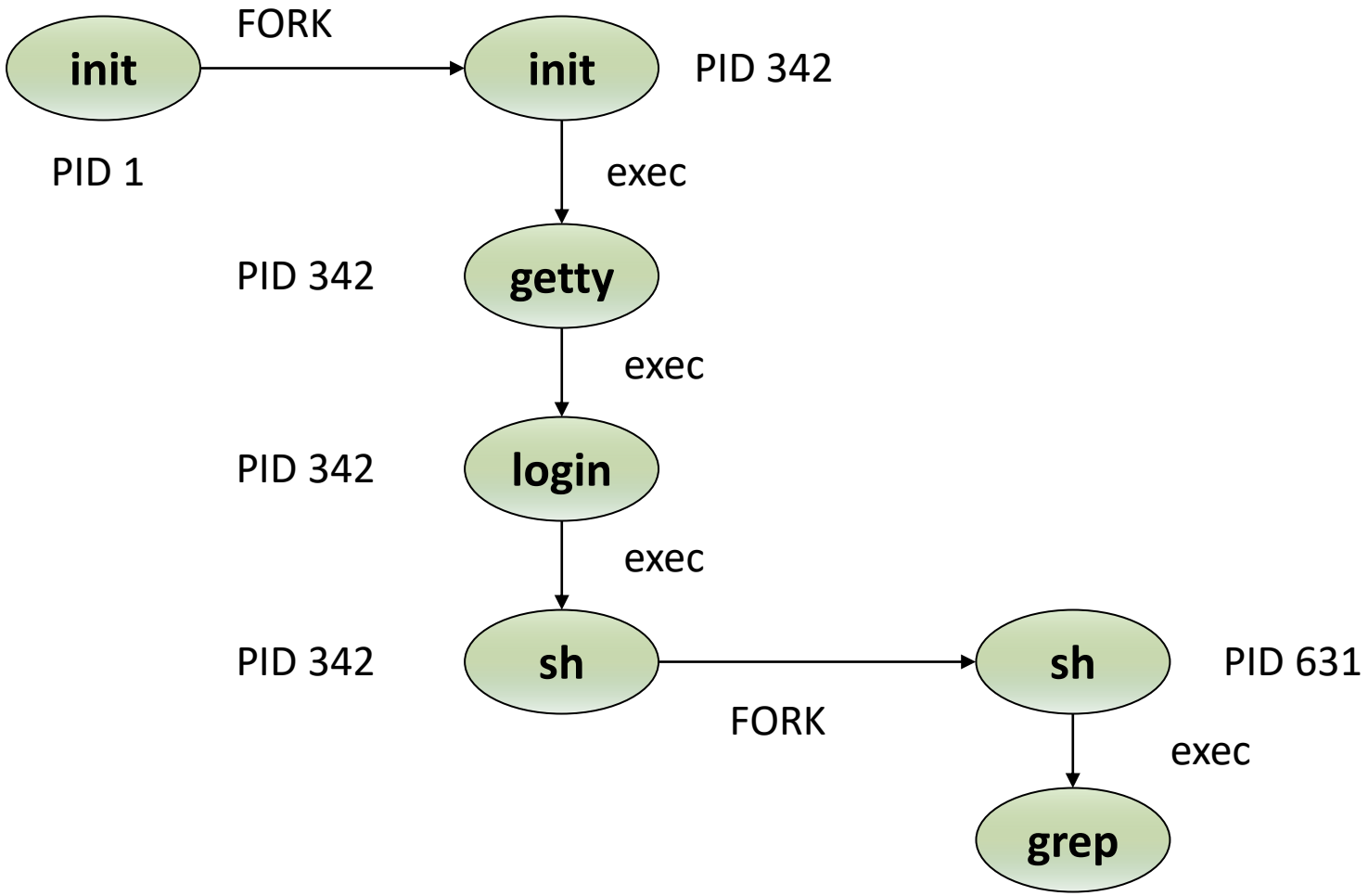
1 SIGHUP Hangup
2 SIGINT Interrupt process
3 SIGQUIT Quit process
4 SIGILL Illegal instruction
5 SIGTRAP Trace or breakpoint trap
6 SIGABRT Abort process
7 SIGEMT Emulation trap
8 SIGFPE Arithmetic exception
9 SIGKILL Kill process
10 SIGBUS Bus error
11 SIGSEGV Segmentation fault
12 SIGSYS Bad system call
13 SIGPIPE Broken pipe
14 SIGALRM Alarm clock
15 SIGTERM Terminated
16 SIGUSR1 User signal 1
17 SIGUSR2 User signal 2
18 SIGCHLD Child status changed
19 SIGPWR Power fail or restart
20 SIGWINCH Window size change
21 SIGURG Urgent socket condition
22 SIGPOLL Pollable event
23 SIGSTOP Stopped (signal)
24 SIGTSTP Stopped (user)
25 SIGCONT Continued
26 SIGTTIN Stopped (tty input)
27 SIGTTOU Stopped (tty output)
28 SIGVTALRM Virtual timer expired
29 SIGPROF Profiling timer expired
30 SIGXCPU CPU time limit exceeded
31 SIGXFSZ File size limit exceeded
32 SIGWAITING Concurrency signal reserved by threads library
33 SIGLWP Inter-LWP signal reserved by threads library
34 SIGFREEZE Checkpoint freeze
35 SIGTHAW Checkpoint thaw
36 SIGCANCEL Cancellation signal reserved by threads library

Process creation

SOFTENGER
Strive to Excel

- A new process is created because an existing process makes an exact copy of itself. This child process has the same environment as its parent, only the process ID number is different. This procedure is called forking.
- After the forking process, the address space of the child process is overwritten with the new process data. This is done through an exec call to the system.
- The fork−and−exec mechanism switches an old command with a new, while the environment in which the new program is executed remains the same, including configuration of input and output devices, environment variables and priority. This mechanism is used to create all UNIX processes, so it also applies to the Linux operating system.
- Even the first process, init, with process ID 1, is forked during the boot procedure in the so−called bootstrapping procedure.

- fork–and–exec mechanism.



SOFTENGER
Strive to Excel

Ending processes

- When a process ends normally (it is not killed or otherwise unexpectedly interrupted), the program returns its *exit status* to the parent.
- The exit status is a number returned by the program providing the results of the program's execution.
- The system of returning information upon executing a job has its origin in the C programming language in which UNIX has been written.

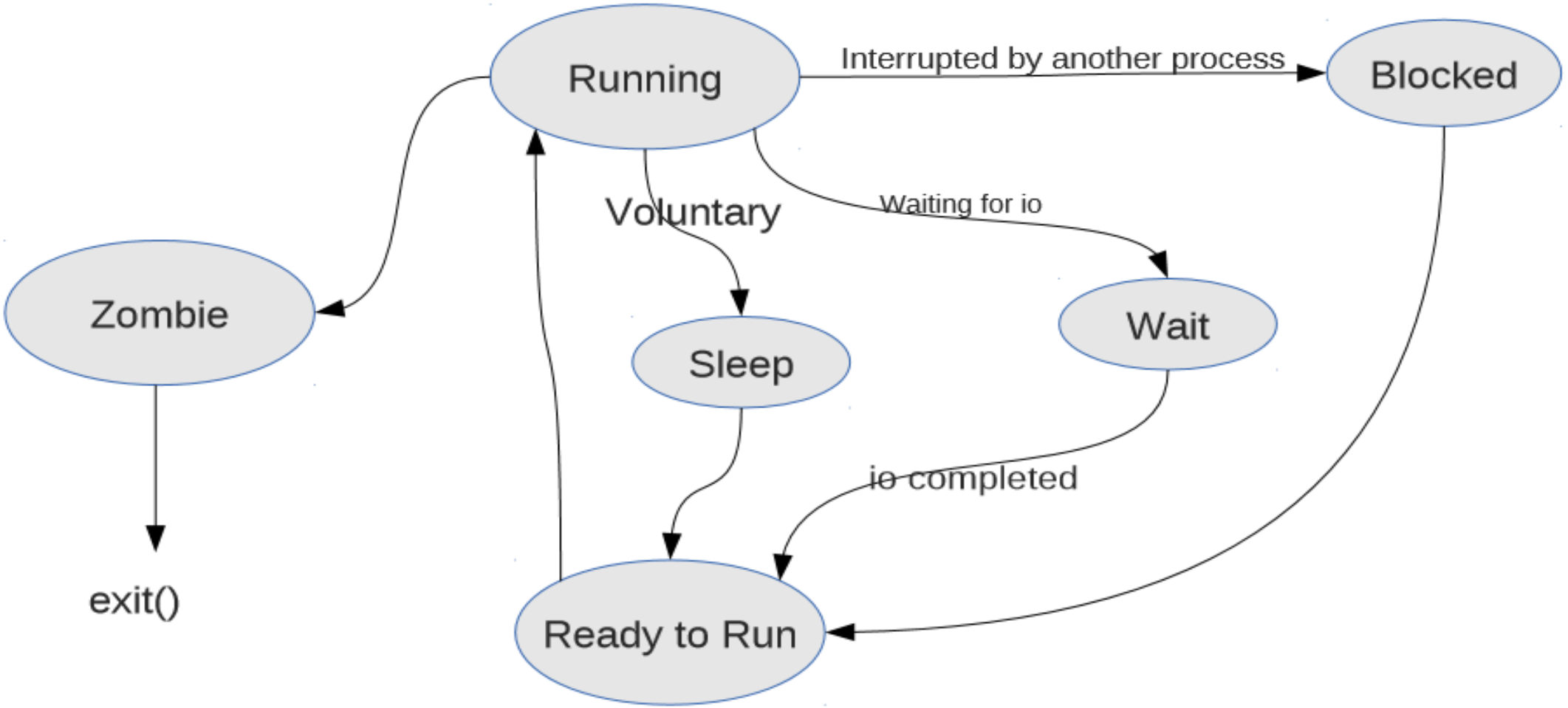
SOFTENGER
Strive to Excel

Ending processes

- The return codes can be interpreted by the parent, or in scripts.
- The values of the return codes are program−specific. This information can usually be found in the man pages of the specified program.
- For example :
  - The grep command returns −1 if no matches are found, upon which a message on the lines of "No files found" can be printed.
  - you can exit code of last executed command by executing:

    echo $?
  - The Bash Builtin command true, which does nothing except return an exit status of 0, meaning success.

Process State Life Cycle



Process States

SOFTENGER
Strive to Excel

- The ps command shows the **process identification number** (listed under PID) for each process you own, which is created after you type a command. This command also shows you the **terminal** from which it was started (TTY), the **cpu time** it has used so far (TIME), and the **command** it is performing (COMMAND).
- If you add the **-l** option to the ps command, the system displays other process information, including the **state** of each running process (listed under S). The following list defines the codes used to describe processes.
  - O – Process is running on a processor
  - S – Sleeping: Process is waiting for an event to complete
  - R - Runnable: Process is on run queue
  - I - Idle: Process is being create.
  - Z – Zombie state: Process terminated and parent not waiting
  - T – Traced: Process stopped by a signal because parent is tracing it
  - X – SXBRK state: Process is waiting for more primary memory.

Process states

- **New** : A process has been created but has not yet been admitted to the pool of executable processes.
- **Ready to run**: Processes that are prepared to run if given an opportunity. That is, they are not waiting on anything except the CPU availability.
- **Running**: The process that is currently being executed. (Assume single processor for simplicity.)
- **Blocked** : A process that cannot execute until a specified event such as an IO completion occurs.
- **Exit**: A process that has been released by OS either after normal termination or after abnormal termination (error).
- **Sleep** : voluntary sleep by programmer

- In Unix/Linux, can effectively run many processes at once
- In fact, OS makes it appear as if many are running simultaneously
- OS runs one job at a time, but quickly switches between jobs
- OS Makes it appear that all jobs are running concurrently
- To view running processes, use ‘ps’

```
prompt$   ps
    PID          TTY          TIME         CMD
  14394          pts/0        00:00:00     bash
  14995          pts/0        00:00:00     ps
```

Outputs the following fields (by default)

- Process ID (PID)
- Terminal from which the process was started (TTY)
- CPU time of the process (TIME)
- The name of the command that started the process (CMD)

Prints the following processes (by default)

- Only processes that belong to you
- Only processes started during your current session

Some ps Options

- To print all processes use ‘ps -e’
  - Incl. those not in current session and those that don’t belong to you
- To display more fields use ‘ps -f’
  - Includes UID (username), PPID (parent PID), C (processor utilization), STIME (start time)
- To display a process tree use ‘ps -H’
- Of course, these can also be used together: ‘ps -efH’

Killing Processes

SOFTENGER
Strive to Excel

- Sometimes, we need to kill an errant process
  - Use ‘kill’ command: ‘kill <pid>’
  - Can only kill processes that belong to you

```
prompt$ ps
  PID   TTY      TIME      CMD
16786   pts/0    00:00:00  errant_proc

prompt$ kill 16786
```

- In some cases, process won’t die with default kill
  - Need -9 option (-9 means *really* kill)

```
prompt$ kill -9 16786
```

Suspending Processes

- Sometimes, want to suspend rather than kill process And continue process later on
- An example:
  - Sorting a really large file
  - Assume it takes a few minutes to sort
  - May want to suspend, do some other work, and continue it later
  - Can do this by typing <ctrl>-z
  - Similar usage to <ctrl>-c

```
prompt$ sort really_large_file
[1]+  Stopped    sort really_large_file
```

‘[1]’ is the job number

Each suspended process gets a unique job number

Continuing the Process: fg

SOFTENGER
Strive to Excel

- To restart the process use ‘fg’
  - Bring the process back to the foreground
  - ‘fg’ continues the most recently suspended process
  - Use ‘fg %<job number>’ to continue another process

```
prompt$ fg
sort really_large_file
```

- To list suspended processes use ‘jobs’

```
prompt$ jobs
[1]+  Stopped    sort really_large_file
```

  - ‘+’ indicates most recently suspended job
  - ‘-’ (not shown here) indicates next most recently suspended job

Processing in the Background

SOFTENGER
Strive to Excel

- Usually, don't want to suspend, but to send to the background
  - Process still runs, but we can continue using the shell
  - Use '&'

```
prompt$ sort really_large_file &
[2] 21130
```

  - Notice that it gets a job number (just like with suspending)

```
prompt$ jobs
[1]+  Stopped    sort really_large_file
[2]-  Running    sort really_large_file &
```

  - Process run without '&' runs in the foreground
  - Only one process can run in foreground at a time

Moving to Background: bg

SOFTENGER
Strive to Excel

- Sometimes, need to move from foreground to the background
  - For example, we forget to add ‘&’ to a long-running command

- Moving a process from foreground to background:
  - First, suspend the process
  - Then, use ‘bg’
  - Sends the most recent job to the background

```
prompt$ bg
[2]+ sort_really_large_file
```

  - To send a less recent job to the background use ‘bg %<job number>’