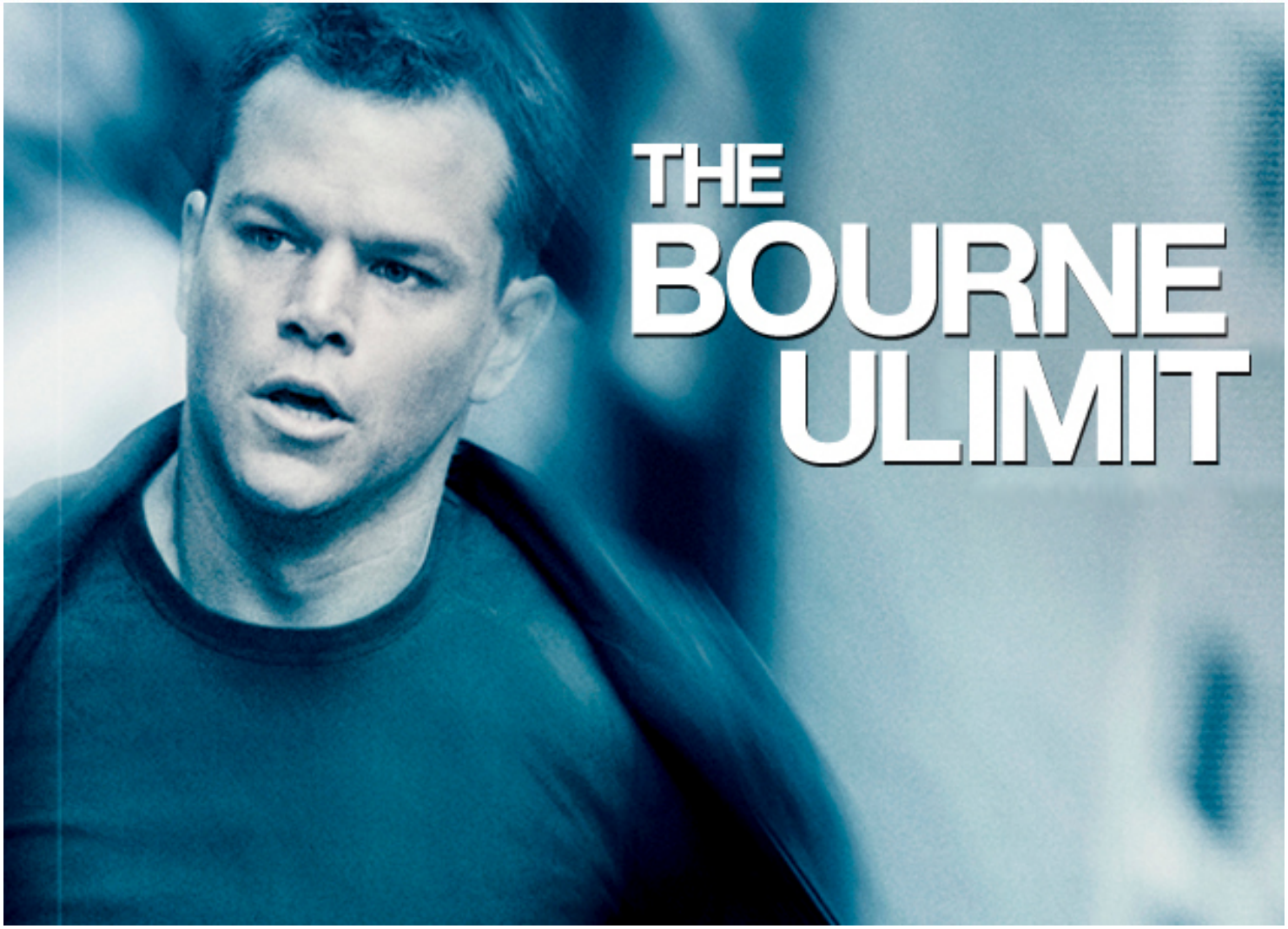
UNIX RTFM: ulimit(1)/limit(1)

Gilbert Detillieux, Computer Science

Presented to MUUG, March 2018

University of Manitoba



Uhm… No!

What is ulimit?

- Built-in command in Bourne Shell and derivatives. (ksh, bash, etc.)
- Used for getrlimit(2)/setrlimit(2) handling, in scripts or command line.
- Displays or sets various per-process, system-dependent resource limits.
- Settings affect current shell and child processes.

ulimit(1posix) Man Page

- NAME

  ulimit --- set or report file size limit

- SYNOPSIS

  ulimit [-f] [*blocks*]

- DESCRIPTION

  The *ulimit* utility shall set or report the file-size writing limit imposed on files written by the shell and its child processes (files of any size may be read). Only a process with appropriate privileges can increase the limit.

- OPTIONS

  The *ulimit* utility shall conform to the Base Definitions volume of POSIX.1-2008, *Section 12.2, Utility Syntax Guidelines*.

  The following option shall be supported:

  -f Set (or report, if no *blocks* operand is present), the file size limit in blocks. The **-f** option shall also be the default case.

- OPERANDS

  The following operand shall be supported:

  blocks The number of 512-byte blocks to use as the new file size limit.

  …

bash(1) Man Page

ulimit [**-HSabcdefiklmnpqrstuvxPT** [*limit*]]

Provides control over the resources available to the shell and to processes started by it, on systems that allow such control. The **-H** and **-S** options specify that the hard or soft limit is set for the given resource. A hard limit cannot be increased by a non-root user once it is set; a soft limit may be increased up to the value of the hard limit. If neither **-H** nor **-S** is specified, both the soft and hard limits are set.

…

-a All current limits are reported
-b The maximum socket buffer size
-c The maximum size of core files created
-d The maximum size of a process's data segment
-e The maximum scheduling priority ("nice")
-f The maximum size of files written by the shell and its children

…

bash(1) Man Page

ulimit [**-HSabcdefiklmnpqrstuvxPT** [*limit*]]

Provides control over the resources available to the shell and to processes started by it, on systems that allow such control. The **-H** and **-S** options specify that the hard or soft limit is set for the given resource. A hard limit cannot be increased by a non-root user once it is set; a soft limit may be increased up to the value of the hard limit. **If neither -H nor -S is specified, both the soft and hard limits are set.**

…

-a All current limits are reported
-b The maximum socket buffer size
-c The maximum size of core files created
-d The maximum size of a process's data segment
-e The maximum scheduling priority ("nice")
-f The maximum size of files written by the shell and its children

…

ulimit Default – Careful Now!

- Shows only soft limit(s):

  ulimit -f

- Shows only hard limit(s):

  ulimit -Hf

- But, sets ***both*** soft and hard limit:

  ulimit –f 409600

- Non-root users can't increase hard limits!
  (Soft limits can't be increased beyond hard limits.)

ulimit - Examples

- Show all (soft) limits:

  ulimit –a

  ulimit -Sa

- Show all hard limits:

  ulimit -Ha

- Set file size limit:

  ulimit –Sf 409600

  ulimit –Sf unlimited

ulimit - Output

- $ ulimit -Sa
- core file size (blocks, -c) 0
- data seg size (kbytes, -d) unlimited
- scheduling priority (-e) 0
- file size (blocks, -f) unlimited
- pending signals (-i) 128448
- max locked memory (kbytes, -l) 64
- max memory size (kbytes, -m) unlimited
- open files (-n) 1024
- pipe size (512 bytes, -p) 8
- POSIX message queues (bytes, -q) 819200
- real-time priority (-r) 0
- stack size (kbytes, -s) 8192
- cpu time (seconds, -t) unlimited
- max user processes (-u) 128448
- virtual memory (kbytes, -v) unlimited
- file locks (-x) unlimited

ulimit – More Examples

- Suppress core dumps:

  ulimit –Sc 0

- Limit program data segment size:

  ulimit –Sd 409600

- Limit number of open file descriptors:

  ulimit –Sn 256

- Limit number of processes/user:

  ulimit –Su 1024

tcsh(1) Man Page

limit [**-h**] [*resource* [*maximum-use*]]

Limits the consumption by the current process and each process it creates to not individually exceed *maximum-use* on the specified *resource*. If no *maximum-use* is given, then the current limit is printed; if no *resource* is given, then all limitations are given. If the **-h** flag is given, the hard limits are used instead of the current limits. The hard limits impose a ceiling on the values of the current limits. Only the super-user may raise the hard limits, but a user may lower or raise the current limits within the legal range.

Controllable resources currently include (if supported by the OS):

cputime	the maximum number of cpu-seconds to be used by each process
filesize	the largest single file which can be created
datasize	the maximum growth of the data+stack region via sbrk(2) beyond the end of the program text

…

tcsh(1) Man Page

…

unlimit [**-hf**] [*resource*]

Removes the limitation on *resource* or, if no *resource* is specified, all *resource* limitations. With **-h**, the corresponding hard limits are removed. Only the super-user may do this. Note that **unlimit** may not exit successful, since most systems do not allow *descriptors* to be unlimited. With **-f** errors are ignored.

…

limit (tcsh) - Output

```
% limit
cputime         unlimited
filesize        unlimited
datasize        unlimited
stacksize       8192 kbytes
coredumpsize    0 kbytes
memoryuse       unlimited
vmemoryuse      unlimited
descriptors     1024
memorylocked    64 kbytes
maxproc         128448
maxlocks        unlimited
maxsignal       128448
maxmessage      819200
maxnice         0
maxrtprio       0
maxrttime       unlimited
```

ulimit –f macOS Weirdness

- $ ulimit -Sf 409600;ulimit -f;/bin/echo ‘WTF?';ulimit –f
- 409600
- WTF?
- unlimited
- $

TRAILBLAZER ADVENTURER
INNOVATOR DEFENDER CHALLENGER
ADVENTURER TRAILBLAZER DEFENDER VISIONARY
VISIONARY ADVENTURER TRAILBLAZER CHALLENGER DEFENDER VISIONARY
ADVENTURER TRAILBLAZER CHALLENGER DEFENDER VISIONARY ADVENTURER TRAILBLAZER CHALLENGER

Questions?

UNIVERSITY OF MANITOBA