システムソフトウェア・小課題2

16-9999-9・情工 太郎

2018年11月X日

1　xv6のスピンロック実装におけるxchgの役割について

1.1　xchgを使用せず，CPUコア数を1として実行した結果

```
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=1
Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 5
init: starting sh
$
```

実行結果は、上のようになり、コア数1では正常に起動した。
次に、usertestを実行してみた結果が以下の通りである。

```
$ usertests
usertests starting
...(中略)...
ALL TESTS PASSED
$
```

1.2 CPU コア数を 2 以上にして実行した結果

```
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=2
Booting from Hard Disk..xv6...
cpu1: starting 1
QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=3
Booting from Hard Disk..xv6...
cpu1: starting 1
cpu2: starting 2
QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=4
Booting from Hard Disk..xv6...
cpu1: starting 1
cpu2: starting 2
cpu3: starting 3
QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=5
...(中略)...
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=8
Booting from Hard Disk..xv6...
cpu1: starting 1
cpu2: starting 2
cpu3: starting 3
cpu4: starting 4
cpu5: starting 5
cpu6: starting 6
cpu7: starting 7
cpu0: starting 0
QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$
```

結果として、CPU コア数が 2 以上になると、３０秒ほど待ったとしても、最後の 1 個のコアが起動をせずに止まったままになってしまう。何度か起動しようとしていると、CPU コア数が 5 以上の時には時々、cpu0 が起動することがあるが、それが起動したとしても、固まったまま動かなかった。

1.3 上記の結果になった理由

上記の結果をまとめると、CPU コア数が 1 より大きい時に問題が生じるということだった。その原因としては、スピンロックの状態を確認する　lk-¿locked != 0 という行と　lk-¿locked = 1; という行が別々の CPU で別々に実行される可能性が存在することである。複数の CPU で lk-¿locked

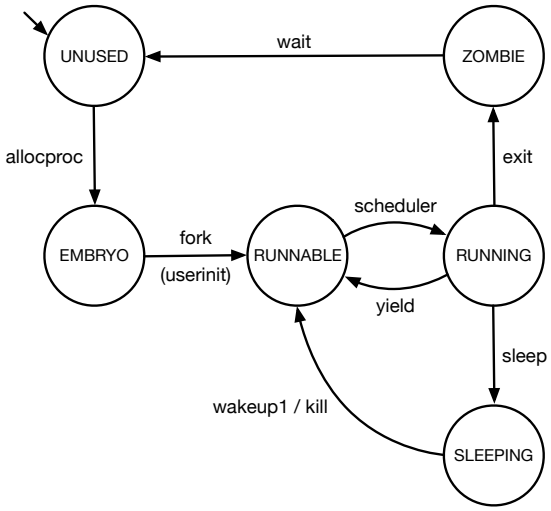

図 1: xv6 のプロセスの状態

が 0 であることを確認した後に、同時に lk-¿locked を 1 してしまうと、ロックを複数の CPU で確保することになり、デッドロックが起こる。

逆に、xchg を呼び出している時は、アセンブリ命令で xchg を用いている。x86 における xchg 命令を用いると、ハードウェアにロックをさせることができ [2]、そのことを用いて、ロックの確保を確実に一つの CPU しかできないようにしている。

従って、ロックの確保をハードウェアレベルで排他していないと正しく実装できないということである。

2　xv6 のスピンロック実装における割込みの扱いについて

2.1　pushcli/popcli を削除した場合の実行結果

```
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=1
Booting from Hard Disk..xv6...
cpu0: starting 0
lapicid 0: panic: mycpu called with interrupts enabled

 8010328f 80103d8d 80103585 80102a74 80102bb1 0 0 0 0 0QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=2
Booting from Hard Disk..xv6...
cpu1: starting 1
lapicid 1: panic: mycpu called with interrupts enabled

  8010328f 80103d8d 80103585 80102a74 80102a8e 705a 0 0 0 0
...(中略)...
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=8
Booting from Hard Disk..xv6...
cpu1: starting 1
lapicid 1: panic: mycpu called with interrupts enabled

 8010328f 80103d8d 80103585 80102a74 80102a8e 705a 0 0 0 0QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$
```

実行結果は、上のように panic が呼び出され、CPU コア数に依らず、起動しない状態になってしまった。

2.2 pushcli/popcli をそれぞれ cli/sti に置き換えた場合の実行結果

```
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=1
Booting from Hard Disk..xv6...

(チラつき)
press Ctrl-B to configure iPXE (PCI 00:03.0)...
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=2
QEMU: Terminated
Booting from Hard Disk..xv6...
cpu1: starting 1
lapicid 1: panic: mycpu called with interrupts enabled

 8010328f 80102a61 80102a8e 705a 0 0 0 0 0 0QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=3
Booting from Hard Disk..xv6...
cpu1: starting 1
lapicid 1: panic: mycpu called with interrupts enabled

 8010328f 80102a61 80102a8e 705a 0 0 0 0 0 0QEMU: Terminated
...(中略)...
cs-mac80:System-Software hoshino.s.af$ make qemu-nox CPUS=8
Booting from Hard Disk..xv6...
cpu1: starting 1
lapicid 1: panic: mycpu called with interrupts enabled

 8010328f 80102a61 80102a8e 705a 0 0 0 0 0 0QEMU: Terminated
cs-mac80:System-Software hoshino.s.af$
```

CPU コア数が 1 の時は、”Booting from Hard Disk..xv6...”まで表示され、”press Ctrl-B to configure iPXE (PCI 00:30.0)...”だと思われる文字列がチラついて表示された。

一方で、CPU コア数が 2 以上の時は、panic が呼び出されて、起動しなかった。

2.3 上記の結果になった理由

まず、pushcli と popcli をコメントアウトした時の実行結果について説明する。

pushcli と popcli は、割り込み禁止を管理している。もし、ある hoge というプロセスが lock0 というロックを確保したあとに、foge というもう一つのプロセスが割り込みをし、lock0 のロックを確保しようとすると、hoge は割り込まれた状態のままで実行されず、lock0 を解放できないので、foge も lock0 の解放を待ち、デッドロックに繋がる状況になってしまう。よって、pushcli と popcli を用いてロックを確保する時に割り込みを禁止する必要があるが、それをコメントアウトすると

デッドロックが生じるはずである。今回、出力の結果としては、panic が呼び出されるということだったが、それは xv6 側で不本意な割り込みに対してエラー処理をしているからである。

次に、pushcli と popcli をそれぞれ cli と sti にした時の実行結果について説明する。

cli と sti は、それぞれ割り込みを禁止、許可するハードウェア命令である。pushcli と popcli を用いているときは、pushcli を呼び出した回数だけ popcli を呼び出さないと割り込みの禁止が解放されないという実装になっている [3]。従って、cli と sti を代わりに使用した場合、一つの CPU で複数のプロセスがロックを確保している時に、ロックの確保で正しく割り込みが禁止されるが、プロセスのうちの一つでもロックを解放すると、割り込み許可状態になり、ロックを確保したプロセスが割り込まれてデッドロックが起きる危険性が生じる。それにより、pushcli と popcli をコメントアウトした時と同じような現象が起こる。

3　感想・意見

参考文献

[1] Russ Cox, Frans Kaashoek & Robert Morris, “xv6, a simple Unix-like teaching operating system”, Sep., 2018, https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf.

[2] Intel, “Intel 64 and IA-32 Architectures Software Developer’s Manual, page 2396 and 2397”, Nov., 2018, https://software.intel.com/sites/default/files/managed/39/c5/325462-sdm-vol-1-2abcd-3abcd.pdf

[3] Russ Cox, Frans Kaashoek & Robert Morris, “xv6, a simple Unix-like teaching operating system”, page 54, Nov., 2018, https://pdos.csail.mit.edu/6.828/2014/xv6/book-rev8.pdf