

システムソフトウェア・試験の解答と解説

2020年度 (2020年11月26日・試験時間100分)

1. (a) 2

解説 ブートブロック (0), スーパーブロック (1) に続くブロックになるのでブロック番号は2となる.

(b) 42

解説 $\text{logstart} + \text{nlog} = 42$.

(c) 61

解説 1 ブロックあたりの dinode 構造体の数は $1024/64 = 16$, inode ブロックの個数は $\lceil \text{ninode}/16 \rceil = 19$ なので, bmapstart は $\text{inodestart} + 19 = 61$.

(d) 1938

解説 総ブロック数 (size)2000 から, ブートブロック (1個), スーパーブロック (1個), ログブロック (40個), inode ブロック (19個), ビットマップブロック (1個) を引いた数 1938 が nblock となる.

2. B, E, F, G, (H), I, J

解説

A, B ある一つの sleeplock (L とする) について, acquiresleep 内の sleep で複数のスレッドが待っているとする. L について releasesleep が実行されると wakeup によってそれらのスレッド全てが起こされる (RUNNABLE になる) が, acquiresleep の while が if になっているとそれらが全て L を獲得できることになってしまう (実際には二番目以降に release を実行しようとしたスレッドが panic を発生させる). したがって, while 文で再度 $\text{lk} \rightarrow \text{locked}$ を調べるようにすることで, 最初に L を獲得したスレッド以外は再

び sleep を実行するようにしている. ここで, 起こされたスレッドが sleep を抜ける際に当該 sleeplock に付随するスピンロックを再度確保する必要があるため, while 文の条件式を複数のスレッドが同時に検査することはない. したがって, L を確保できるのは1つのスレッドとなる.

C, D, E コード3の4行目では, 現在実行中のカーネルスレッドに対応するプロセスに関するプロセス構造体 (P とする) に付随するスピンロックを確保している. その後9行目の sched を実行すると, swtch によってコード5の13行目に制御が移動する. この時点での変数 p (関数 scheduler のローカル変数) には P へのポインタが格納される. したがって, P に付随するスピンロックはコード5の15行目にある release によって解放される.

コード3の12行目にある release が解放するのはコード5の8行目にある acquire が確保したスピンロックである. また, コード4の9行目にある release が解放するのはコード4の4行目にある acquire が確保したスピンロックである.

G, H いずれも同じスコープ内の変数であり, 代入が行われていないためその値は等しい.

補足 H については for ループによる繰り返しの各回それぞれについて8行目と15行目で等しくなるという意図で出題した. しかし問題の表現が曖昧で, ループによる変化を考えると「常に等しい」とは言えないため, この間については正しいとも誤りとも解釈できる. よってこの間については全員に得点を与えている.

I, J, K xv6-riscv のソースコードをみるとわかるが, swtch の呼び出しは scheduler と sched のみで行われている. sched の呼び出しは sleep , yield および exit

のみであり、それぞれ呼び出しの前にプロセスの状態を SLEEPING, RUNNABLE および ZOMBIE に設定している。

L 複数の CPU コアで動作している場合、各コアにおいて scheduler が動作しており、それぞれが proc から RUNNABLE なプロセスを見つけ次第 swtch によって制御を移している。そのためコード 6 の 7,8 行目と、9 行目では実行している CPU コアが異なることがある。したがって、コード 7 のように変更することはできない。

3. (a) 4096

解説 xv6-riscv が動作する RISC-V プロセッサでは 1 ページは 4096 バイトになっている。

(b) 4194304

解説 $1024 \times 1024 \times 4 = 4194304$

(c) 1025

解説 $4194304 = 4096 * 1024$ であるが、malloc のヘッダに利用する 16 バイトの領域のためにもう 1 ページ必要になる。

(d) 1025 (中間のページテーブルのエントリを入れた場合は 1029~2562)

解説 (物理ページを指す) ページテーブルエントリ数は物理ページ数と等しい。中間のページテーブルエントリを考慮に入れる場合は、1 段あたり $4096/8 = 512$ 個のエントリを取り得る。中間のページテーブルであるが、第 1 段は常に 1 ページである。この問題の場合、第 2 段が 1 ページ、第 3 段が 3 ページとる場合が最小となり、合計 4 エントリ必要なので、合計 1029 となる。実際に物理メモリに確保できるかどうかは別として、第 2 段を 512 ページ、第 3 段を 1025 ページとした場合が最大となるが、この場合第 2 段のエントリ数は 1025、第 3 段も 1025 となるため、合計 $512 + 1025 + 1025 = 2562$ となる。

(e) sbrk

解説 sbrk はプロセスが使用するメモリ領域を増加・減少させるシステムコールである。

pipe はプロセス間通信を行うパイプを作るため、mknod はデバイスファイルを作るため、fstat はファイルの情報を得るため、dup はファイルディスクリプタをコピーするためのシステムコールである。