Prova scritta di sistemi operativi del 15 settembre 2021

c.1

Per semplicità, supponiamo che vengano richieste sincronizzazioni di al più `MAX` processi, dove `MAX` è un intero positivo.

```
monitor alrv {

  condition okc[1 : MAX + 1]; // okc[i]: rendez-vous con almeno #i processi
  int alc[1 : MAX]; // alc[i]: w[i]

  p.e. at_least(int n) {
    int s = 0, // sommatoria dei primi #i elementi di alc
        max = 0; // max indice "i" di alc per cui s_i >= i
    ++alc[n]; // conteggia l'at_least attuale
    for (int i = 1; i <= MAX; ++i) {
      s += alc[i];
      if (s >= i)
        max = i;
    }
    if (!max)
      okc[n].wait(); // non abbiamo combinato un rendez-vous (sicuramente n > 1)
    else
      for (int i = 1; i <= max; ++i) // rendez-vous di max processi
        okc[i].signal();
    okc[n].signal();
    --alc[n]; // non conteggiare più l'at_least attuale
  }

}
```

c.2

```
void chained_send(T msg, list_of_pids dests) {
  if (dests != NULL)
    ssend(Pair<T, list_of_pids>(msg, dests.next), dests.value);
}

T chained_recv(void) {
  Pair<T, list_of_pids> res = srecv(ANY);
  chained_send(res.first, res.second);
  return res.first;
}
```

g.1

1. 1,2,3,4,5,1,2,3,6,1,2,3,7,8,6 genera 12 *page fault* con una memoria di 5 *frame*, ma solo 11 con una memoria di 4.
2. Sia n un numero naturale strettamente maggiore di 3. Scegliamo la sequenza k in questione con indici successivi da 1 a $2n+4$ inclusi. Sia i un indice valido per la sequenza. Allora:

$$k_i = \begin{cases} i, & \text{se } 1 \leq i \leq n \\ 1, & \text{se } i = n+1 \\ 2, & \text{se } i = n+2 \\ n+1, & \text{se } i = n+3 \\ i-n-3, & \text{se } n+4 \leq i \leq 2*n+4 \end{cases}$$

g.2

1. Se il processo corrente non è stato interrotto per una chiamata bloccante o perché è terminato, e lo *scheduler* è progettato per rimetterlo nella coda dei processi pronti solo dopo aver individuato il successivo, allora siamo in un caso particolare: la coda dei processi pronti "sembra" vuota, ma in realtà starebbe per essere aggiunto il processo corrente. Per evitare questo inutile passaggio, lo scheduler può semplicemente continuare a usare il processo corrente. Al di fuori di questa particolare situazione, se esistono processi bloccati viene selezionato uno pseudo-processo che, nel caso di *scheduler* con priorità sarebbe di ultima classe, perennemente disponibile: è il processo vuoto. Il sistema operativo sta di fatto attendendo che un qualsiasi processo vero e proprio si sblocchi per riprendere la computazione.
2. Mantenendo ogni dato in due copie, RAID1 chiede sempre il doppio dei dischi effettivamente necessari per la memorizzazione, mentre RAID5 chiede l'equivalente (ma sparso) di uno solo dei dischi (che sono ben più di due) in strip di parità. Questa differenza sostanziale comporta un abbattimaneto dei costi di acquisto, manutenzione e smaltimento. Un'altra distinzione va fatta in quando a prestazioni in scrittura: anche se RAID1 necessita solo di una scrittura su due dischi, mentre RAID5 di una lettura da 1 e di una scrittura su 2, in RAID1 ogni disco è un potenziale collo di bottiglia ad ogni accesso in scrittura. In RAID 5 lavoriamo, fra tutti i dischi, solo con quello che effettivamente contiene i dati che ci interessano e quello che di volta in volta contiene lo *strip* di parità. Questi variano di accesso in accesso.
3. Se il contatore è errato, esso lo può essere per difetto o per eccesso. Procediamo per casi. Se il contatore è troppo basso, ci vorranno meno decrementi del dovuto per farlo giungere a 0, e quindi questo momento arriverà prima del previsto con conseguente eliminazione prematura dell'inode e puntatori penzolanti annessi. Se il contatore invece è troppo alto, anche dopo tutti che tutti i riferimenti a esso si saranno scollegati, esso non giungerà a 0, né verrà eliminato. In altre parole, si ha una fuga di risorse.

4. La valutazione dell'algoritmo del banchiere viene effettuata prima di garantire a un processo l'accesso a una risorsa: se lo stato risulta sicuro, l'accesso può venire garantito anche subito, mentre in caso contrario viene ritardato nella speranza che la valutazione successiva abbia esito diverso.