

# Sistemi Operativi

22 Gennaio 2003 - compito 1

Cosa è una lista di controllo degli accessi (*access control list* o *ACL*) nel contesto dei meccanismi di protezione?

---

Spiegare quali sono i possibili stati in cui può trovarsi un processo e quando si verificano le transizioni da uno stato a un altro.

---

Descrivere come funzionano i sistemi RAID di livello 4 e di livello 5. Quali sono i vantaggi del livello 5 rispetto al livello 4? Motivare la risposta.

---

Spiegare sinteticamente cos'è un semaforo e cosa fanno le relative primitive up e down.

---

Abbiamo due processi  $P$  e  $Q$  che vogliono sincronizzarsi alla fine delle sezioni 1 e 3. In altre parole, vogliono assicurarsi di iniziare le sezioni 2 e 4 solo dopo che entrambe le sezioni 1 e 3 sono state eseguite.

```
void P(void) {
    int msgp;

    ..sezione 1..
    receive(Q,&msgp);
    send(Q,&msgp);
    ..sezione 2..
}
```

```
void Q(void) {
    int msgq;

    ..sezione 3..
    send(P,&msgq);
    receive(P,&msgq);
    ..sezione 4..
}
```

```
void Q(void) {
    int msgq;

    ..sezione 3..
    receive(P,&msgq);
    send(P,&msgq);
    ..sezione 4..
}
```

Supponendo che la receive sia bloccante (blocca il processo se non c'è nessun messaggio pronto da ricevere), quali dei due processi  $Q$  indicati sopra è quello giusto? Motivare la risposta.

# Sistemi Operativi

22 Gennaio 2003 - compito 2

Cose è una lista di *capability* nel contesto dei meccanismi di protezione?

---

Illustrare la differenza tra dispositivi a blocchi e a caratteri. Dare anche qualche esempio delle due classi di dispositivi.

---

Spiegare sinteticamente il funzionamento delle primitive di comunicazione **send** e **receive** nel caso di comunicazione sincrona e nel caso di comunicazione asincrona.

---

Spiegare qual è la differenza tra un processo e un thread. Quali sono gli elementi comuni a tutti i thread di un processo e quali gli elementi privati di ogni thread?

---

Abbiamo due processi  $P$  e  $Q$  che vogliono sincronizzarsi alla fine delle sezioni 1 e 3. In altre parole, vogliono assicurarsi di iniziare le sezioni 2 e 4 solo dopo che entrambe le sezioni 1 e 3 sono state eseguite.

```
semaforo p1=0;
semaforo p2=0;
```

```
void P(void) {
    ..sezione 1..
    down(&p1);
    up(&p2);
    ..sezione 2..
}
```

```
void Q(void) {
    ..sezione 3..
    down(&p1);
    up(&p2);
    ..sezione 4..
}

void Q(void) {
    ..sezione 3..
    up(&p1);
    down(&p2);
    ..sezione 4..
}
```

Quali dei due processi  $Q$  indicati sopra è quello giusto? Motivare la risposta.