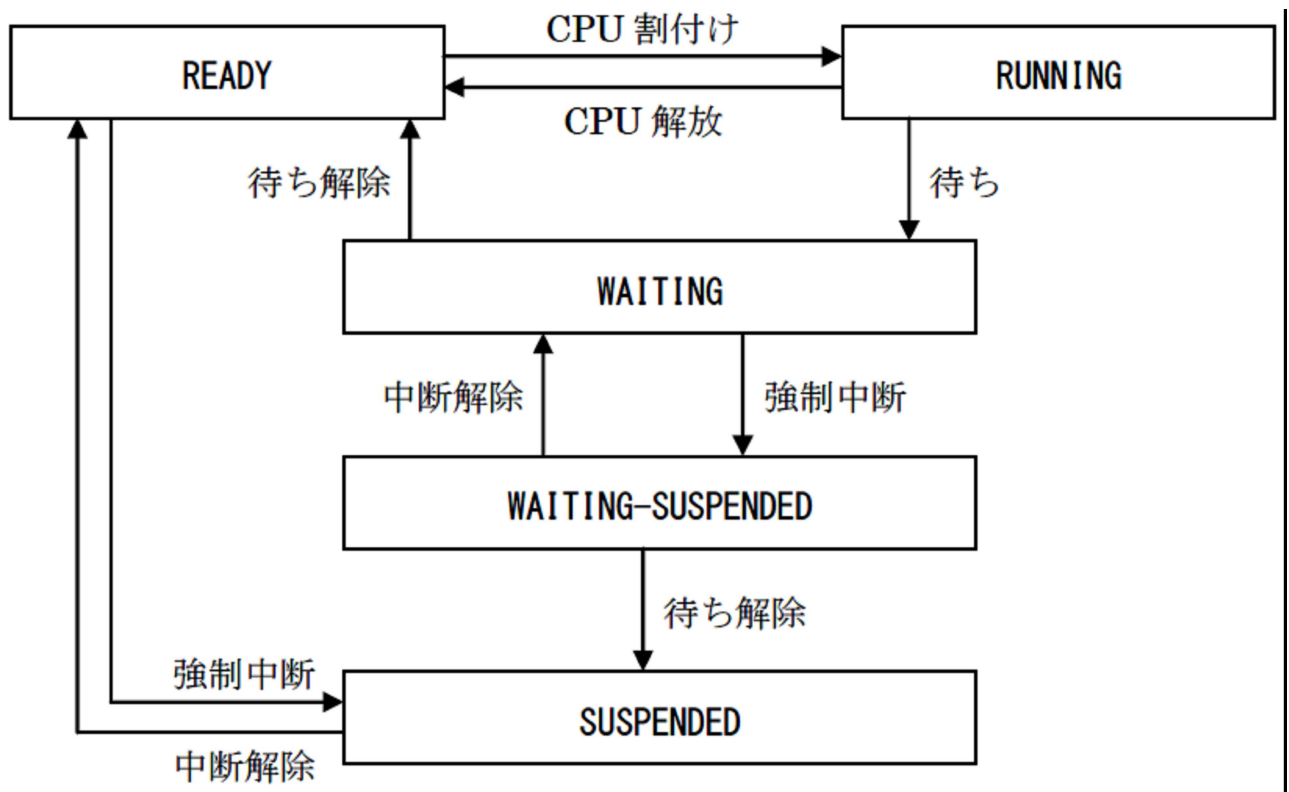


## テーマ：状態遷移



**RUNNING 状態（実行状態）：**タスクが動作している状態。CPU が 1 つの場合、RUNNING 状態のタスクは 1 つだけ。

**READY 状態（実行可能状態）：**タスクが動作できる条件が整っているが、他の優先順位の高いタスクが動作しているため、動作できない状態。通常、READY 状態のタスクは複数存在する。OS がタスクの優先度に従って READY 状態から RUNNING 状態へ移行する。

**WAITING 状態（待ち状態）：**タスクが動作中に自分から WAIT した状態。他のタスクからの指示では WAITING 状態には遷移しない。必ず自らの指示で WAITING 状態に遷移する。自分自身が要求した事象が発生すると、WAITING 状態から READY 状態に移行する。

**SUSPENDED 状態（強制待ち状態）：**他のタスクからの要求によって、タスク自身の意志とは関係なく強制的に中断された状態。

**WAITING-SUSPENDED 状態（二重待ち状態）：**待ち状態と強制待ち状態が重なった状態。待ち状態のタスクに対して、強制待ち状態への移行要求があった場合に、二重待ち状態になる。

【実習 1】今まで学習してきたタスク関連機能コマンドを図中に入力してください。

【実習 2】現在、全てのタスク（タスク 1， 2， 3）が READY 状態にある。その後の状態遷移について考えなさい。

【実習 3】タスク 1， 2， 3 のうち、タスク 1， 3 が WAITING 状態、タスク 2 が RUNNING 状態にある。タスク 1 が READY 状態に移行したときの状態遷移について考えなさい。

```
//Sample 1
/*****
/**** reset process ****/
/****
void uinit(void)
{ /* initialize */
  clock_switch();
  led_init();
  lcd_init();
  sw_init();
  trb_init();
  asm("FSET I");
}
/****
/**** tasks ****/
/****
void tsk01(void) {
  while(1) {
    bz_on();
    tslp_tsk(1);
    bz_off();
    tslp_tsk(1);
  }
}

void tsk02(void) {
  UB k=0;
  lcd_clr();
  while(1) {
    lcd_xy(0,0);
    lcd_int(k);
    k++;
    tslp_tsk(500);
  }
}

void tsk03(void) {
  UB k;
  while(1) {
    k=sw_read();
    led_set(k);
  }
}
```

```
//Sample 2
/****
/**** reset process ****/
/****
void uinit(void)
{ /* initialize */
  clock_switch();
  lcd_init();
  trb_init();
  asm("FSET I");
}
/****
/**** tasks ****/
/****
void tsk01(void) {
  UB k=0;
  while(1) {
    lcd_xy(0,0);
    lcd_int(k);
    k++;
    tslp_tsk(10);
  }
}

void tsk02(void) {
  UB k=0;
  while(1) {
    sus_tsk(1);
    lcd_xy(10,0);
    lcd_int(k);
    k++;
    rsm_tsk(1);
    tslp_tsk(10);
  }
}

void tsk03(void) {
  while(1) {
    slp_tsk();
  }
}
```

【実習 4】タスクの状態が RUNNING 状態から SUSPENDED 状態へ移行することはない。その理由を考えなさい。

【実習 5】WAITING 状態から READY 状態へ移行するのはどのような場合か。全て示しなさい。

【実習 6】Sample1 の状態遷移について検討しなさい。

【実習 7】Sample2 の状態遷移について検討しなさい。