```c
#ifndef TYPES
#define TYPES

/*****************************************************************************
 *
 * This header file contains utility types definitions.
 *
 *****************************************************************************/

#include "../h/const.h"

typedef signed int cpu_t;


typedef unsigned int memaddr;


/* Device Register */
typedef struct {
  unsigned int d_status;
  unsigned int d_command;
  unsigned int d_data0;
  unsigned int d_data1;
} device_t;

#define t_recv_status   d_status
#define t_recv_command    d_command
#define t_transm_status   d_data0
#define t_transm_command  d_data1


/* Bus Register Area */
typedef struct {
  unsigned int rambase;
  unsigned int ramsize;
  unsigned int execbase;
  unsigned int execsize;
  unsigned int bootbase;
  unsigned int bootsize;
  unsigned int todhi;
  unsigned int todlo;
  unsigned int intervaltimer;
  unsigned int timescale;
  unsigned int TLB_Floor_Addr;
  unsigned int inst_dev[DEVINTNUM];
  unsigned int interrupt_dev[DEVINTNUM];
  device_t  devreg[DEVINTNUM * DEVPERINT];
} devregarea_t;


/* Pass Up Vector */
typedef struct passupvector {
    unsigned int tlb_refll_handler;
    unsigned int tlb_refll_stackPtr;
    unsigned int execption_handler;
    unsigned int exception_stackPtr;
} passupvector_t;


#define STATEREGNUM 31
typedef struct state_t {
  unsigned int  s_entryHI;
  unsigned int  s_cause;
  unsigned int  s_status;
  unsigned int  s_pc;
  int       s_reg[STATEREGNUM];

} state_t, *state_PTR;

/* Process control block type */
typedef struct pcb_t {
  /* Process queue fields */
  struct pcb_t    *p_next,  /* Pointer to next entry */
        *p_prev,  /* Pointer to prev entry */

  /* Process tree fields */
        *p_prnt,  /* Pointer to parent */
        *p_child, /* Pointer to first child */
        *p_sib,   /* Pointer to sibling */
        *p_sibPrev; /* Pointer to previous sibling */

  /* Process status information */
  state_t**     p_s;    /* Processor state */
  state_t**     p_s;    /* Processor state */
  cpu_t     p_time;   /* CPU time used by the processor */
  int     *p_semAdd;  /* Pointer to sema4 on which process blocked */

  /* Support layer information */
  struct support_t  *p_supportStruct; /* Pointer to support struct */

} pcb_t, *pcb_PTR;

/* Semaphore descriptor type */
typedef struct semd_t {
  struct semd_t *s_next;  /* Next element on the ASL */
  int   *s_semAdd;  /* Pointer to the semaphore */
  pcb_t   *s_procQ; /* Tail pointer to a process queue */
} semd_t;


/* Registry */
#define s_at  s_reg[0]
#define s_v0  s_reg[1]
#define s_v1  s_reg[2]
#define s_a0  s_reg[3]
#define s_a1  s_reg[4]
#define s_a2  s_reg[5]
#define s_a3  s_reg[6]
#define s_t0  s_reg[7]
#define s_t1  s_reg[8]
#define s_t2  s_reg[9]
#define s_t3  s_reg[10]
#define s_t4  s_reg[11]
#define s_t5  s_reg[12]
#define s_t6  s_reg[13]
#define s_t7  s_reg[14]
#define s_s0  s_reg[15]
#define s_s1  s_reg[16]
#define s_s2  s_reg[17]
#define s_s3  s_reg[18]
#define s_s4  s_reg[19]
#define s_s5  s_reg[20]
#define s_s6  s_reg[21]
#define s_s7  s_reg[22]
#define s_t8  s_reg[23]
```

```c
#define s_t8  s_reg[23]
#define s_t9  s_reg[24]
#define s_gp  s_reg[25]
#define s_sp  s_reg[26]
#define s_fp  s_reg[27]
#define s_ra  s_reg[28]
#define s_HI  s_reg[29]
#define s_LO  s_reg[30]



#endif

```