PA 4 Y86 Emulator and Dissembler

Design and Implantation

Y86 Emulator

```
typedef enum {

    AOK, /*Everything is fine; no errors*/
    HLT, /*Halt instruction has occurred; program ends normally*/
    INS,
       ADR
    } ProgramStatus;
```

ProgramStatus is a variable that stores the status of the program.

```
union converter {
    char byte[4];
    int integer;
};
```

Converter can convert 4-byte char to a corresponding integer.

```
char * append (char * str, char c);
int hextodec(char * num);
```

This function convert a string to hex

```
char * hextobin(char c) ;
void executeprog();
```

This part is the core of the program, it will run based on the cases. For example, 00 is Nop.

```
int bintodec(char * num);
char * copy (char * str);
int gettwobytes(char * str, int position);
void printmemory (int size);
void printstatus ();
void getargs(unsigned char * arg1, unsigned char * arg2);
```

Y86 dissembler is similar with the emulator, the only difference is to printf the command line.

Difficulties I met:

The major work I did is to handle different cases. It is not hard but there is

tons of work to do.