

BENGAL ENGINEERING & SCIENCE UNIVERSITY, SHIBPUR
BE (CST) 6th SEMESTER EXAMINATION, 2012
Systems Programming (CS 605)

F.M: 70

Time: 3 hrs.

- Answer 6 questions taking any 3 from each group.
- Use a single answer-script for both the GROUPS.
- Two marks are reserved in each group for neatness.

GROUP A

1. a) What do you mean by relocation and symbol resolution? Explain with a suitable example.
b) What are the five usual sections that can be found in most of the object files produced by the assembler/compiler?
c) What do you mean by linkable, loadable and shared object files? Name the system software that produces them. [4 + 4 + 3]
2. a) What is a symbol and a symbol table. Describe the structure of the ELF symbol table entry.
b) List the (non-local) symbols of the main.o and swap.o (for the following 'C' program) as it would have been done by READELF tool.

```
/* main.c */ void swap(); int buf[2] = {1, 2};
int main(){
    swap(); return 0;
}
/*swap.c */ extern int buf[]; int bufp0 = &buf[0]; int*bufp1;
void swap() { int t;
    bufp1 = &buf[1];
    t = *bufp0;
    *bufp0 = *bufp1; *bufp1 = t;
}
```

- c) In the context of a linker each relocatable object module (m) has a symbol table in which there are three different kinds of symbols; describe them. [4 + 4 + 3]
3. a) Describe the functions using which you may link dynamically shared libraries from within an application.
b) Write functions to carry out the addition and multiplication of two complex numbers. Now create dynamically shared library (showing the appropriate commands to be used) libcmplx.so. Finally write a main routine which dynamically loads and links the shared library libcmplx.so to add/multiply two complex numbers. [4 + 7]
4. a) Write the rules for resolving multiply defined strong and weak externals by the linker. What will be the output for the following program? Explain and justify your answer.

```
/* m1.c */
char a='a'; char b = 'b'; int c=10;
int main(){void bb(void); a='A'; bb(); printf("%c %c %d\n", a, b, c); return 1;}
/* b1.c */
long a =0; bb(){}
```

- b) Describe the steps taken by LINUX system for loading an executable object file. [6 + 5]
5. Write short notes on: a) INTEL 8-bit hex object file; b) ELF object file. [5½ × 2]

GROUP B

6. a) What is a control flow? What is an Exceptional Control Flow (ECF)? With the help of suitable diagrams show the transfer of control, due to different types of exceptions, from a user process to the exception handler and the return (optional) of control back to the user process.
- b) Show the organisation of the private address space provided by the LINUX system to each process. Explain the meaning of *kernel* mode and *user* mode and how the mode is changed from *user* to *kernel* and vice versa. How can we access/modify the kernel data structure in LINUX system? [5 + (3 + 3)]

7. a) What is a context of a process? What is the information stored as the context of a process? Show with diagram how the context switching between two concurrent processes; namely process 'A' and process 'B' is done.
- b) Explain what would be the output of the following program. Assume all appropriate header files have been included at the beginning of the source file. [8 + 3]

```
void doit()
{ fork(); fork(); printf("hello\n"); return; }

int main()
{ doit(); printf("hello\n"); exit (0); }
```

8. a) What is a signal? Describe the corresponding events and the default actions for the signals; SIGINT, SIGILL, SIGALRM, SIGCHLD and SIGKILL. State for what reasons (or events) the kernel sends a signal to a destination process. Also state the action that is taken by the receiving process once the signal is delivered to it.
- b) The following program with appropriate header catches the SIGINT signal. Make necessary changes in the program so that it catches SIGFPE signal instead of SIGINT. [Hint: SIGFPE signal is delivered to the process by the kernel indicating divide by zero error] [7 + 4]

```
void handler (int sig) { printf("caught SIGINT\n"); exit (0);}
int main(){ if (signal(SIGINT, handler)) == SIG_ERR)
    printf("signal error\n");
    pause(); exit (0); }
```

9. With a page size of 4Kb, alignment of 8 bytes, show the memory allocation of a program having three functions; namely, main, funcA and funcB whose text, data and bss segment sizes (in Hex) are shown in the table. The start address is 0X08048000. [11]

Functions	text	Data	bss
main	345A	120	320
funcA	12AF	A98	123
funcB	23BC	12AB	456

10. Write short notes on: a) PIC; b) Object file handling tools in UNIX.

[5½ × 2]