PAGE No.
DATE / /

Assignment - B1

Q1] Explain linking with an example.

→ Link editors are commonly known as linkers. The compiler automatically invokes the linker as the last step in compiling a program. The linker inserts code (or maps in shared libraries) to resolve program library references, and/or combines object modules into an executable image suitable for loading into memory. On Unix-like systems, the linker is typically invoked with ld command.

Static linking is the result of the linker copying all library routines used in program into executable image. This may require more disk space & memory than dynamic linking. but is both faster & more portable.

Dynamic linking is accomplished by placing the name of a sharable library in the executable image. Actual linking with library routines does not occur until the image is run, when both the executable & the library are placed in memory.

eg: /* main.c */
```
void swap ();
int buf [2]={3,4};
int main ()
{
swap ()
return 0;
}
```

```
/* swap.c */
extern int buf [];
int *bufp0 = &buf [0];
int *bufp1;
void swap ()
{ int temp;
bufp1 = &buf [1];
temp = bufp0
*bufp0 = ~~temp~~ *bufp1
*bufp1 = temp;
}
```

PAGE No.
DATE / /

The example program consists of two source files, main.c and swap.c. The main function initializes two element array of ints, and then calls the swap function to swap the pair.

as [other args] - o/tmp/main.o/tmp/main.s

The driver goes through the same process to generate swap.o finally it runs the linker program ld, which combines main.o & swap.o along with the necessary system object files, to create the executable object file. p:

-o p [system object files and args]
/tmp/ main.o/ tmp/swap.o.

To run executable p, we type its name on Unix shell's command line:

unix > ./p.

The shell invokes a function in the OS called loader, which copies the codes and in the executable file p into the memory and then transfers control to beginning of the program.

Q2] What are the advantages of dynamic linking library?

→ 1) Multiple processes that load the same DLL at the same base address share a single copy of the DLL in physical memory. Doing this saves system memory and reduces swapping.

2) When the functions in a DLL change, the applications that use them donot need to be recompiled or relinked as long as the function arguments, calling conventions and return values do not change. In constrast, statically linked object code requires that the application

PAGE No.
DATE / /

be relinked when functions change.

3) A DLL can provide after market support. For eg, a display driver DLL can be modified to support a display that was not available when the application was initially shipped.

Q.3 What are the advantages & disadvantages of static linking library?

→ Advantages:

i) The application can be certain that all its libraries are present & that they are the correct version. This avoids dependency problem.

ii) With static linking, it is enough to include those parts of the library that are directly & indirectly referenced by the target executable.

Disadvantages:

i) In Static linking, the size of the executable becomes greater than in dynamic linking, as the library code is stored within the executable rather than in seperate files.

ii) Static libraries cause a wastage of both of these resources because every program has to possess its own copy of library, on the other hand dynamic linking allows several programs to share same code instead of having individual copies of the code.

PAGE No.
DATE / /

Q4) Explain loading with an example?

→ Supporting seperate compilation requires OS software to combine the code from multiple compilation steps This software is called a link editor or, more simply a linker.

→ · It provides an executable file from several object files. It relocates seperately compiled code segments It resolves external references. Example of program controlled dynamic loading -

```
void myPrintf (** arg)
{ static int loaded = 0;
    if (! loaded)
      load ("printf");
      loaded = 1;
      printf( arg);
}
```