

# Result Report for Network Print System

- Whole development stage
- Feedback of System Test and Supplement
- Discussion of Whole stage

**Project Team**

**Team 5**

Latest update on:

**2015-12-08**

**201112395 임준혁**

---

**Team Information**

## Table of Contents

1	Introduction	3
1.1	Objectives	3
1.2	Background	3
1.3	References	3
2	Whole Development Stage	3
2.1	개발 방법	3
2.1.1	설계와 개발	3
2.1.2	테스트	4
2.2	개발 기간	4
2.2.1	SA~SD	4
2.2.2	Development	4
2.2.3	Test(unit and system)	4
2.3	개발 환경	4
2.3.1	하드웨어	4
2.3.2	소프트웨어	4
3	Feedback of System test and supplement	5
3.1	스레드 동기화 문제	5
3.2	리눅스 환경에서의 문제점	5
4	Discussion of Whole Stage	5
4.1	어려웠던 점과 느꼈던 점	5

## 1 Introduction

### 1.1 Objectives

Network Printer System(이하 NPS로 명시)의 SA, SD 시작으로 한 설계 과정에서부터 실제 개발과 그에 대한 unit, system 단계에서의 각각의 테스트까지 전체 소프트웨어 개발 과정의 단계와 이를 수행하는 전체 과정을 설명한다. 또한 마지막 System test에서 발견된 프로그램에 문제점에 대해서 설명하고, 이에 대한 대응으로 보완한 점을 설명한다. 마지막으로 전체 과정을 진행하면서 힘들었던 점, 혹은 느꼈던 점을 서술한다.

### 1.2 Background

NPS는 소프트웨어로 구현된 프린트 시스템으로써, 사용자의 요청을 받아 특정 조건이 성립될 때 출력물을 가상의 Paper tray인 메모장에 텍스트파일로 저장하는 기능을 한다. 사용자의 출력 요청뿐만 아니라, 관리자 상태에서의 유저 등록과 삭제, 소모품 충전 또한 가능한 Network Printer이다.

### 1.3 References

2015SE\_B\_NPS\_T5\_SRA VER 2.0

2015SE\_B\_NPS\_T5\_SDS VER 2.0

2015SE\_B\_NPS\_T5\_UTP

2015SE\_B\_NPS\_T5\_UTR

## 2 Whole Development Stage

### 2.1 개발 방법

#### 2.1.1 설계와 개발

Term 프로젝트 과제로 주어진 NPS SRS Ver 1.0을 통해, 요구사항에 대한 documentation을 얻었다. Software Engineering 에서 학습한 개발 및 설계 방법의 일부인 SA, SD 를 이용하여 이 분석된 요구사항을 포함할 소프트웨어의 구조를 설계할 수 있었고, 여기에서 만들어진 결과물인 DFD와 Structure Chart를 이용하여 개발을 했다. 개발은 C언어를 통해서 이루어 졌으며, 개발상의 함수 call관계는 Structure chart를 참고하여 만들어졌다.

### 2.1.2 테스트

Software Engineering 수업에서 배운 바로는 Unit test는 개발자가 진행하되, System test의 경우 개발자가 아닌 개발에 참여하지 않은 사람이 하여서 개발자가 미처 생각하지 못한 오류나 잡아내거나 error masking을 방지하나, 여건이 여의치 않아, 개발한 본인이 Unit test와 System test를 전부 진행하였다. 이는 소스코드 상에서 혹은 input대비 output의 결과값을 확인하면서 진행하였다.

## 2.2 개발 기간

### 2.2.1 SA~SD

2015-11-17~2015-11-24

### 2.2.2 Development

2015-11-24~2015-11-27

### 2.2.3 Test(unit and system)

2015-11-27~2015-12-01

## 2.3 개발 환경

### 2.3.1 하드웨어

Cpu : Intel® Core™ i5-5200U cpu @ 2.20GHZ

RAM : 4GB

System type : 64bit Operation System

OS : Windows 8

### 2.3.2 소프트웨어

컴파일러 : GCC

개발 언어 : C

구현 환경 : Cygwin 64 Terminal

### 3 Feedback of System test and supplement

#### 3.1 스레드 동기화 문제

스레드를 통해서 프린트 액션을 하는 데에 있어서, 서로 다른 스레드가 실행되면 같은 자원을 사용할 경우 이에 대한 동기화가 필요해서, 이 것이 같은 자원을 사용할 수 없도록 동기화 및 스레드가 동시에 실행되지 않도록 설정했다.

#### 3.2 리눅스 환경에서의 문제점

System test를 진행 중에 발견된 리눅스 환경에서  $w_n$ 이 시각적으로 표시되지 않는 문제를  $wr_wn$ 로 수정하여 시각적으로 결과값으로 표시되게 하였다. 이는 결과값으로 나오는 result~.txt에 적용되며, 시각적으로 이것이 한 줄임을 표시할 수 있게 하였다. 그리고 한 파일에 모든 페이지가 저장된다는 점에서 페이지에 대한 정보를 담았다.

### 4 Discussion of Whole Stage

#### 4.1 어려웠던 점과 느꼈던 점

이번 학기에 가장 힘들었던 수업을 꼽으라면 난 망설이지 않고 이 수업을 선택할 수 있다. c언어를 제대로 구사하지 못하는 상태에서, 다른 팀들이 개발이 모두 끝난 상황에서, 혼자서 Term 프로젝트를 시작한 점은 이번 학기를 정말 악몽 같은 학기로 만든 주 원인이다. 그러나 나는 이 점이 힘들었던 점인 동시에 엄청난 이점이라는 생각이 들었다. 실제로 4명에서 프로젝트를 진행함에 있어서, 모든 인원의 생각을 조율해야 하고, 각자가 알고 있는 개념에 대한 synchronization 과정을 수없이 진행해야 했다. 그 과정에서 많은 에너지 소모가 있었고, 개발의 속도 혹은 전체 진행 과정의 속도 역시 현저히 느려질 수 밖에 없었다. 그러나 이 과정을 혼자 진행하다 보니, 내가 알고 있는 개념만 혹은 방법만 맞다는 가정이 있다면 이 synchronization 과정이나 팀원들과의 의견 조율로 인한 에너지 소모가 적고, 개발의 속도 면에서 엄청난 효율을 가질 수 있었다. 물론 혼자서 진행하는 프로젝트인 만큼, 다른 팀에 비해서 생각하지 못한 부분이나 완벽하지 못한 부분이 존재할 여지가 훨씬 많고, 프로그램의 완성도 면에서 봤을 때도 많은 차이를 보인다는 것은 부정할 수 없다. 그러나 소프트웨어 설계와 개발 과정에서 개발이나 설계인원이 설계 속도와 개발 속도에 비례해서 증가하지 않는다는 점, 아니 오히려 어느 선을 넘어서면 반비례 관계로 갈 수도 있다는 점을 느꼈다.

때문에 개발과정에서 힘든 점보단 오히려 여러 가지 문서화를 해야 하는 상황에서 한

정된 시간 안에 요소들에 대한 이해를 하고 이에 대해 문서화를 하는 과정에서 많은 어려움을 느꼈다. 그러나 이 모든 과정을 혼자 진행한 만큼 모든 과정에 대한 이해와, 연습을 하게 된 것 같아서 힘들었지만 배운 것이 정말 많은 수업이었고 실습이었다. 이 기법에 대해서 조금 더 정리하고 다듬어서, 다음 프로젝트를 진행함에 있어서 설계 단계에서 적용하여, 더 효율적으로 개발을 할 수 있도록 할 것이다.